THE α-ICA ALGORITHM

Yasuo Matsuyama, Naoto Katsumata, Yoshitaka Suzuki and Shuichiro Imahara

Department of Electrical, Electronics and Computer Engineering,
Waseda University, Tokyo 169-8555, Japan
{yasuo, katsu, ysuzuki, shoe16}@wizard.elec.waseda.ac.jp

ABSTRACT

The α-logarithm is an extension of the logarithm which contains the usual logarithm as the special case of $\alpha = -1$. Usage of information measures based upon this extended logarithm is expected to be effective to speedup of convergence, i.e., to the improvement of learning aptitude. In this paper, speedup of the mutual-info-min ICA is investigated. The method is based upon the minimization of the α-mutual information expressed by the α-divergence. Therein, two main strategies of exploiting the past and future information are presented. The first one uses causal computation. This leads to the momentum α-ICA. The other exploits prediction for non-causal part. This gives the turbo α-ICA. It is also possible to devise orthogonal versions which can possibly suppress null signals. Examples are shown on sample-based α-ICA's in the batch mode. The usage of the α-dependent strategies shows remarkable speedup with only a little extra complexity.

1. INTRODUCTION

Logarithm has been widely used for measuring amount of information such as entropy, mutual information and Kullback-Leibler divergence (KL-divergence). Among them, the KL-divergence contains mutual information and entropy as its special cases [1]. This relationship played the main role in deriving the entropy-max Independent Component Analysis (entropy-max ICA) [2]. This maximization is closely related to the minimization of mutual information [3]. Since the mutual information is a special case of the KL-divergence, the mutual-info-min ICA can be regarded as a KL-divergence-min ICA.

As was mentioned above, the KL-divergence is based on the logarithm. On the other hand, there is a class of extended logarithm which can define meaningful information measures. This is the α-logarithm [4], [5] which includes the usual logarithm as the special case of $\alpha = -1$. The α-logarithm has been successfully applied to the speedup of the EM algorithm (Expectation-Maximization algorithm) [4], [5]. This is because the convexity of the optimizing function can be controlled by the parameter α. Therefore, we can expect that the usage of the α-logarithm will be meritorious to speedup of the ICA. In the subsequent sections, this anticipation is positively answered by giving the following algorithms:

(a) Momentum α-ICA,
(b) Turbo α-ICA,
(c) Orthogonal momentum α-ICA,
(d) Orthogonal turbo α-ICA.
  (d0) Purely orthogonal turbo α-ICA
  (d1) Combined turbo α-ICA
  (d2) Orthogonal look-ahead turbo α-ICA

All of these algorithms with $\alpha = -1$ are reduced to the logarithmic case. The complexity of the presented methods is only a little higher than the logarithmic case. Yet, obtained speedup is remarkable. Experiments will show this evidence.

Before stepping into mathematical derivations of algorithms, we comment the usage of typographical fonts. Bold fonts for vectors will not be used in this paper. This is to avoid complicated boldface suffices so that expressions look simpler. Instead, careful comments will be given where misleading might occur.

2. α-DIVERGENCE AND INDEPENDENCE

2.1. α-Logarithm and α-Divergence

The α-logarithm $L^{(\alpha)}(x)$ is defined as follows [4], [5].

$$L^{(\alpha)}(x) = \frac{2}{1+\alpha}(x^{\frac{1+\alpha}{2}} - 1) \qquad (1)$$

This $L^{(\alpha)}(x)$ includes $\log_e x$ as the special case of $\alpha = -1$. The α-logarithm inherits convexity of the logarithm controlled by the parameter α [5].

The α-divergence between probability densities p and q is defined as follows [6], [7], [8].

$$D^{(\alpha)}(p\|q) = \frac{4}{1-\alpha^2}\left\{1 - \int_{\mathcal{Y}} p(q/p)^{\frac{1+\alpha}{2}} dy\right\} \geq 0 \qquad (2)$$

This work was partially supported by Grant-in-Aid for Scientific Research, and Waseda University's Research Projects on High Technologies and New Technologies.

This informational amount reflects the distance between $p(y)$ and $q(y)$. It is non-negative and is equal to zero if and only if $p(y) = q(y)$ almost everywhere with respect to $y \in \mathcal{Y} = R^K$.

2.2. α-Mutual Information

Mutual information is an amount of information which measures independence among random variables. Therefore, it was used in deriving the mutual-info-min ICA [3]. It is a well-known fact of importance that the mutual information is a special case of the KL-divergence [1]. Thus, the mutual-info-min approach can be interpreted as the minimization of the KL-divergence. This is the clue to obtain the class of α-ICA's itemized as (a) - (d) in Section I.

Inheriting the property of the mutual information, the α-mutual information can be defined as follows.

$$I^{(\alpha)}(\wedge_{i=1}^K Y_i) = D^{(\alpha)}\left(p(y_1,\ldots,y_K)\middle\|\prod_{i=1}^K q_i(y_i)\right) \quad (3)$$

The symbol "$\wedge$" is adopted instead of the symbol ";" which appears in standard references [1]. Note that the case of $\alpha = -1$ is reduced to the usual mutual information. The class of α-ICA algorithms is derived using this $I^{(\alpha)}(\wedge_{i=1}^K Y_i)$. This looks a little bit complicated. But, resulting algorithms are simple to implement.

3. α-INDEPENDENT COMPONENT ANALYSIS

3.1. Derivation of the α-ICA

We denote the α-mutual information as follows for simplicity:

$$\begin{aligned} D^{(\alpha)}\left(p(y)\|q(y)\right) &\stackrel{\text{def}}{=} D^{(\alpha)}\left(p(y_1,\ldots,y_K)\|\prod_{i=1}^K q_i(y_i)\right) \\ &= I^{(\alpha)}(\wedge_{i=1}^K Y_i). \end{aligned} \quad (4)$$

In the problem of ICA, we can observe a set of random vectors

$$X(t) = \mathrm{col}\,[X_1(t),\ldots,X_K(t)], \quad (t = 1,\ldots,N). \quad (5)$$

Usually, the components $X_i(t)$ and $X_j(t)$ are not independent each other. Thus, we want to find a matrix $W = \Lambda\Pi A^{-1}$ so that components of

$$WX(t) \stackrel{\text{def}}{=} Y(t) = \mathrm{col}\,[Y_1(t),\ldots,Y_K(t)] \quad (6)$$

are independent each other for $t = 1,\ldots,N$. Here, the matrix A^{-1} gives

$$A^{-1}X(t) = S(t) \quad (7)$$

where

$$S(t) = \mathrm{col}\,[S_1(t),\ldots,S_K(t)] \quad (8)$$

is unknown except that its columns are independent. Λ is a nonsingular diagonal matrix and Π is a permutation matrix. Both are also unknown.

The matrix W can be found by minimizing the above α-mutual information in the form of the α-divergence. A plain gradient descent method can be obtained by computing the gradient of

$$\begin{aligned} I^{(\alpha)}(\wedge_{i=1}^K Y_i) &= \frac{4}{1-\alpha^2}\left[1 - \int_{\mathcal{Y}} p(y)\left\{\frac{\prod_{i=1}^N q_i(y_i)}{p(y)}\right\}^{\frac{1+\alpha}{2}} dy\right] \\ &= \frac{4}{1-\alpha^2}\left[1 - \int_{\mathcal{X}} p(x)\left\{\frac{|W|\prod_{i=1}^N q_i(y_i)}{p(x)}\right\}^{\frac{1+\alpha}{2}} dx\right]. \end{aligned} \quad (9)$$

Then, the negative gradient is as follows.

$$\begin{aligned} -\nabla I^{(\alpha)}(\wedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I^{(\alpha)}(\wedge_{i=1}^K Y_i)}{\partial W} \\ &= \frac{2}{1-\alpha}\int_{\mathcal{X}} p(x) \\ &\quad\times\left\{\frac{|W|\prod_{i=1}^K q_i(y_i)}{p(x)}\right\}^{\frac{1+\alpha}{2}}\left\{W^{-T} - \varphi(y)x^T\right\} dx \end{aligned} \quad (10)$$

Here,

$$-\varphi(y) = \mathrm{col}\left[\left\{\frac{q_1'(y_1)}{q_1(y_1)},\ldots,\frac{q_K'(y_K)}{q_K(y_K)}\right\}\right]. \quad (11)$$

This form has been investigated after [9] by many researchers [2], [3]. Since we obtained the negative gradient (10), a simple form of the update equation is

$$W[t+1] = W[t] + \Delta^{(\alpha)}W[t] \quad (12)$$

with

$$\Delta^{(\alpha)}W[t] = \rho_t\left\{-\nabla I^{(\alpha)}(\wedge_{i=1}^K Y_i)\right\}_{W=W(t)}. \quad (13)$$

Here, t is the index for iteration and ρ_t is a learning rate.

3.2. α-Natural Gradient

Here, we pay attention to the natural gradient [11] for its role as the reduction of the computational complexity on the update term (13). The natural gradient is related to the Fisher information matrix, say $G(W)$. Note that the α-version of the information matrix $M^{(\alpha)}(W)$ has a relationship [4], [5] such that

$$M^{(\alpha)}(W) = \tfrac{1-\alpha}{2}G(W). \quad (14)$$

Therefore, we have the α-version's natural gradient by multiplying $\frac{1-\alpha}{2}W^TW$ to the above gradient of (10).

$$\begin{aligned} -\tilde{\nabla}^{(\alpha)}I^{(\alpha)}(\wedge_{i=1}^K Y_i) &\stackrel{\text{def}}{=} -\frac{\partial I^{(\alpha)}(\wedge_{i=1}^K Y_i)}{\partial W}\left(\tfrac{1-\alpha}{2}W^TW\right) \\ &= \int_{\mathcal{Y}} p(y)\left\{\frac{q(y)}{p(y)}\right\}^{\frac{1+\alpha}{2}}(I - \varphi(y)y^T)dy\,W \end{aligned} \quad (15)$$

Therefore, we use

$$\tilde{\Delta}^{(\alpha)}W = \rho_t\left\{-\tilde{\nabla}^{(\alpha)}I^{(\alpha)}(\wedge_{i=1}^K Y_i)\right\} \quad (16)$$

as the update amount instead of Equation (13). This is the basic update equation from which all of the α-versions of the ICA algorithms are derived. But, it still contains an unknown probability density $q(y)$. Therefore, further modifications are necessary in order to obtain concrete algorithms.

4. IMPLEMENTATION OF THE α-ICA

4.1. Expansion of the α-Dependent Term

First, we expand the integrand of (15).

$$0 \leq \left\{ \frac{q(y)}{p(y)} \right\}^{\frac{1+\alpha}{2}} \approx \frac{1-\alpha}{2} + \frac{1+\alpha}{2} \frac{q(y)}{p(y)} \tag{17}$$

Therefore, we have

$$\begin{aligned} -\tilde{\nabla}^{(\alpha)} I^{(\alpha)}(\wedge_{i=1}^{K} Y_i) \approx \rho_t \tfrac{1-\alpha}{2} \left\{ I - E_{p(y)}[\varphi(y) y^T] \right\} W \\ + \rho_t \tfrac{1+\alpha}{2} \left\{ I - E_{q(y)}[\varphi(y) y^T] \right\} W. \end{aligned} \tag{18}$$

This equation is close to the final implementation based on observed samples since it is a summation of two types of expectations. But, it still contains an unknown probability density $q(y)$. Therefore, further modifications are necessary towards truly implementable computer algorithms. In the following subsections, we give effective approximations to this $q(y)$.

4.2. Causal Realization as the Momentum α-ICA

First, we observe that $q(y)$ is the target function of $p(t)$ such that

$$q(y) = \lim_{t \to \infty} p^{(t)}(y) \tag{19}$$

under an appropriate convergence criterion (e.g., the vague convergence). Then, there is a causal approximation at the t-th iteration such that

$$q(y) \approx p^{(t)}(y) \quad \text{and} \quad p(y) = p^{(t-\tau)}(y). \tag{20}$$

By this approximation, we have the following sample-based learning algorithm.

[Momentum α-ICA: Algorithm of type (a)]

If we use $q(y)$ as $p^{(t)}(y)$ and $p(y)$ as $p^{(t-\tau)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$\begin{aligned} &\tilde{\Delta}^{(\alpha)} W(t) = \tilde{\Delta} W(t) + \tfrac{1-\alpha}{1+\alpha} \tilde{\Delta} W(t-\tau) \\ &= \rho_t \left\{ I - \varphi(Y(t)) Y(t)^T \right\} W(t) \\ &\quad + \tfrac{1-\alpha}{1+\alpha} \rho_t \left\{ I - \varphi(Y(t-\tau)) Y(t-\tau)^T \right\} W(t-\tau) \end{aligned} \tag{21}$$

Thus, we added a momentum term $\tilde{\Delta} W(t-\tau)$ weighted by α. Note that

$$\tilde{\Delta} W(t) = \tilde{\Delta}^{(-1)} W(t) \tag{22}$$

since $\alpha = -1$ is the case of the logarithm.

4.3. Non-causal Realization as the Turbo α-ICA

There is a non-causal approximation at the t-th iteration such that

$$q(y) \approx p^{(t+\tau)}(y) \quad \text{and} \quad p(y) = p^{(t)}(y). \tag{23}$$

Then, we have the following sample-based learning algorithm.

[α-Turbo method (α-Look-ahead method): Algorithm of type (b)]

$$\begin{aligned} &\tilde{\Delta}^{(\alpha)} W(t) = \tilde{\Delta} W(t) + \tfrac{1+\alpha}{1-\alpha} \tilde{\Delta} \hat{W}(t+\tau) \\ &= \rho_t \left\{ I - \varphi(Y(t)) Y(t)^T \right\} W(t) \\ &\quad + \tfrac{1+\alpha}{1-\alpha} \rho_t \left\{ I - \varphi(\hat{Y}(t+\tau)) \hat{Y}(t+\tau)^T \right\} \hat{W}(t+\tau) \end{aligned} \tag{24}$$

Here, the look-ahead terms $\hat{W}(t+\tau)$ and $\hat{Y}(t+\tau)$ are estimations of $W(t+\tau)$ and $Y(t+\tau)$ using the usual log-version. Thus, we added a predicted term $\tilde{\Delta} \hat{W}(t+\tau)$ weighted by α.

We comment here that there is a duality in α between Equations (21) and (24). This is an inherited property from the α-divergence. We also note in advance that $\tau = 1$ works effectively enough for both causal and non-causal methods in spite of the asymptotic relationship (19).

5. ORTHOGONAL α-ICA

Amari et al. [10] introduced an orthogonal ICA which is expected to suppress zero-power fake signals. The idea is to find an update term, say $\tilde{\Delta}^{+} W$, which is orthogonal to $\tilde{\Delta} W$ so that

$$< \tilde{\Delta} W, \tilde{\Delta}^{+} W >_W = 0. \tag{25}$$

Such an update term $\tilde{\Delta}^{+} W$ is obtained as follows. Let

$$\Lambda = \mathrm{diag}\,[\lambda_i]_{i=1}^{K} \tag{26}$$

is a non-singular diagonal matrix. Let

$$W + \tilde{\Delta} W = (I + d\Lambda) W. \tag{27}$$

Then,

$$\tilde{\Delta}^{+} W = \rho \{ \Lambda - \varphi(y) y^T \} W \tag{28}$$

such that

$$\Lambda = \mathrm{diag}\,[\varphi_i(y_i) y_i]_{i=1}^{K}. \tag{29}$$

Therefore, we have the following four types of orthogonal α-ICA algorithms.

[Orthogonal momentum α-ICA: Algorithm of type (c)]

If we use $q(y)$ as $p^{(t)}(y)$ and $p(y)$ as $p^{(t-\tau)}(y)$ at the t-th iteration, then the sample-based learning is as follows.

$$\tilde{\Delta}^{(\alpha)+}W(t) = \tilde{\Delta}^{+}W(t) + \frac{1-\alpha}{1+\alpha}\tilde{\Delta}^{+}W(t-\tau)$$
$$= \rho_t \left\{\Lambda(t) - \varphi(Y(t))Y(t)^T\right\} W(t) + \frac{1-\alpha}{1+\alpha}\rho_t$$
$$\times \left\{\Lambda(t-\tau) - \varphi(Y(t-\tau))Y(t-\tau)^T\right\} W(t-\tau) \quad (30)$$

[α-Turbo methods (α-Look-ahead methods): Algorithms of types (d0) $\sim$ (d2)]

The update term is as follows.

$$\tilde{\Delta}^{(\alpha)+}W(t) = \tilde{\Delta}^{+}W(t) + \frac{1+\alpha}{1-\alpha}\tilde{\Delta}^{+}\hat{W}(t+\tau)$$
$$= \rho_t \left\{\Gamma(t) - \varphi(Y(t))Y(t)^T\right\} W(t) + \frac{1+\alpha}{1-\alpha}\rho_t$$
$$\times \left\{\hat{\Omega}(t+\tau) - \varphi(\hat{Y}(t+\tau))\hat{Y}(t+\tau)^T\right\} \hat{W}(t+\tau) \quad (31)$$

Here, the matrices $\Gamma(t)$ and $\hat{\Omega}(t+\tau)$ are as follows.

(d0) $\Gamma(t) = \Lambda(t)$ and $\hat{\Omega}(t+\tau) = \hat{\Lambda}(t+\tau)$ give a purely orthogonal turbo α-ICA.

(d1) $\Gamma(t) = I$ and $\hat{\Omega}(t+\tau) = \hat{\Lambda}(t+\tau)$ give a hybrid turbo α-ICA of type I.

(d2) $\Gamma(t) = \Lambda(t)$ and $\hat{\Omega}(t+\tau) = I$ give a hybrid turbo α-ICA of type II.

6. SPEEDUP EVALUATION BY EXPERIMENTS

6.1. Benchmark Problems

All of the expectations in the update equations are replaced by $\frac{1}{T}\sum_{i=1}^{T}[\cdot]$ where T is the number of samples in a selected window. Thus, the algorithms are evaluated in their batch modes. We chose mixtures of time series [3] as benchmarking problems. The nonlinearity of $\varphi(y) = y^3$ [9] was used. The convergence speed was measured by the cross-talking error U [3] which checks the closeness of

$$Z = [z_{ij}] = WA \quad (32)$$

to the matrix $\Lambda\Pi$.

$$U = \sum_{i=1}^{K}\left\{(\max_k |z_{ik}|)^{-1}\sum_{j=1}^{K}|z_{ij}| - 1\right\}$$
$$+ \sum_{j=1}^{K}\left\{(\max_k |z_{kj}|)^{-1}\sum_{i=1}^{K}|z_{ij}| - 1\right\} \quad (33)$$

Experiments were started with generating five mixture signals as in Fig. 1 from four true sources. Thus, there is one null source. First, we summarize the convergence speed. The learning rate was $\rho_t = 0.1$. Fig. 2 is the speed of the momentum method for various α. Fig. 3 is the case of the turbo method for various α. The horizontal axes are numbers of iterations. The vertical axes are the cross-talking errors of (33). Both the α-momentum and the α-turbo methods effectively sped up the learning. We can observe that the α-turbo method is more meritorious than the α-momentum method. Around six times speedup over the plain log-version can be observed. The extra complexity for the look-ahead of the α-turbo ICA can be reduced to a fraction of the total complexity of $\tilde{\Delta}W(t)$. This can be done by choosing the window length of the look-ahead smaller than that of $\tilde{\Delta}W(t)$. Fig. 4 illustrates separated signals. Since there were four true sources, there is one ghost signal which is located as the top waveform.

Next, we check the cases of the orthogonal updates. Fig. 5 is the case of the orthogonal momentum α-ICA. Fig. 6 shows the convergence of the purely orthogonal turbo α-ICA. Speedup of the turbo α-method is again remarkable. But, these orthogonal methods require much more iterations than the cases of Fig. 2 and 3. This is because diagonal terms of the update matrices are all zero in the case of the orthogonal methods. Fig. 7 shows separated signals by this purely orthogonal turbo α-ICA. In this case, the third signal can be identified as a null signal. But, such a separation is not always successful.

Next, we check the hybrid methods. Fig. 8 is the convergence curve of the hybrid turbo α-ICA of type I. Fig. 9 shows the convergence of the orthogonal turbo α-ICA of type II. Speedup tendencies are the same as Fig. 2 and 3. In these cases, recovered signals are almost the same as Fig. 4.

7. CONCLUDING REMARKS

We showed efficient usage of the α-logarithm to the independent component analysis. There are several effective variants of the α-ICA method. They are summarized in Table I.

Table I Classification of the α-ICA algorithms.

realization	momentum	turbo	hybrid
basic	(a)	(b)	(d1)
orthogonal	(c)	(d0)	(d2)

Among them, we found that the class of turbo α-ICA algorithms are very effective on the speedup. This observation is valid for both basic and orthogonal cases of (b) and (d0).

We had another observation. Purely orthogonal updates of (c) and (d0) are much slower than the basic methods of (a) and (b). This is due to the lack of the diagonal terms. But, the purely orthogonal methods have the possibility to suppress the appearance of fake signals. Yet, this ability is not always promised.

The hybrid methods showed similar speed to the cases (a) and (b). But, a ghost signal appeared. Thus,

our recommendation is (i) the α-turbo ICA of type (b), and then (ii) the α-momentum ICA of type (a).

The remarkable speedup of the ICA with only a little additional complexity is quite meritorious to the extraction of independent components from gigantic source data. Results on this issue will be reported at the presentation and/or forthcoming papers.

In closing this paper, we note that the speedup by the α-logarithm is a general property beyond individual problems [12]. For instance, the speedup can also be observed in the α-EM algorithms [4], [5].

REFERENCES

[1] Cover, T.M., and Thomas, J.A., Elements of Information Theory, Wiley-Interscience, New York, 1991.

[2] Bell, A.J., and Sejnowski, T.J, An information-maximization approach to blind separation and blind deconvolution, Neural Computation, vol. 7, pp. 1129-1159, 1995.

[3] Yang, H.H., and Amari, S., Adaptive online learning algorithm for blind separation: Maximum entropy and minimum mutual information, Neural Computation, vol. 9, pp. 1457-1482, 1997.

[4] Matsuyama, Y., The α-EM algorithm: A block connectable generalized learning tool for neural networks, Lecture Notes in Computer Science, No. 1240, pp. 483-492, 1997.

[5] Matsuyama, Y., The α-EM algorithm and its basic properties, Trans Inst. Electro. Info. and Comm. Eng., vol. J82-D-I, No. 12, 1999.

[6] Rényi, A., On the measures of entropy and information, Proc. 4th Berkeley Symp. Math. Stat. and Pr., vol. 1, pp.547-561, 1960.

[7] Csiszár, I., A class of informativity of observation channels, Period. Math. Hunga., vol. 2, pp. 191-213, 1972.

[8] Amari, S., Differential geometry theory of statistics, Inst. Math Stat. Lecture notes, vol. 10, pp. 21-94, 1985.

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

[10] Amari, S., Chen, T-P., and Cichocki, A.J., Non-holonomic constraints in learning blind source separation, Proc. ICONIP'97, vol. 1, pp. 633-636, 1997.

[11] Amari, S., Natural gradient works efficiently in learning, Neural Computation, vol. 10, pp. 252-276, 1998.

[12] Matuyama, Y., Niimoto, T., Katsumata, N., Suzuki, Y., and Furukawa, S., α-EM Algorithm and α-ICA Learning Based Upon Extended Logarithmic Information Measures, Proc. IJCNN, 2000.

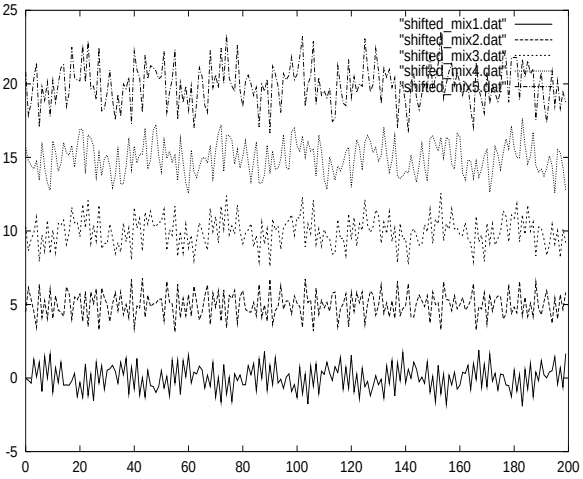

Fig. 1 Mixed source signals.

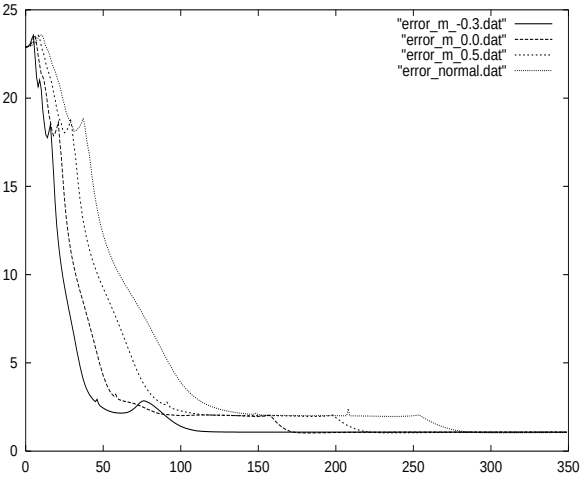

Fig. 2 Momentum α-ICA.

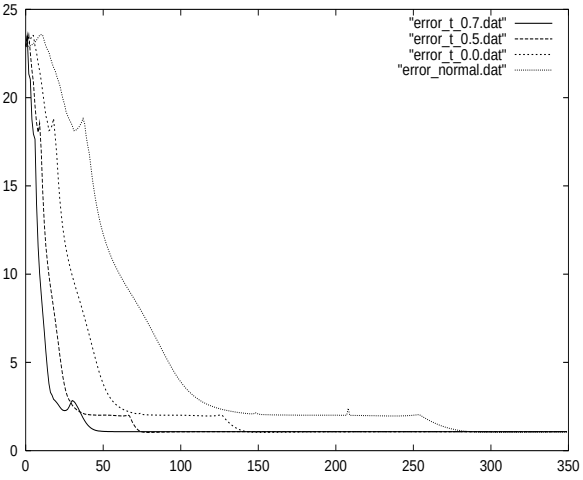

Fig. 3 Turbo α-ICA.

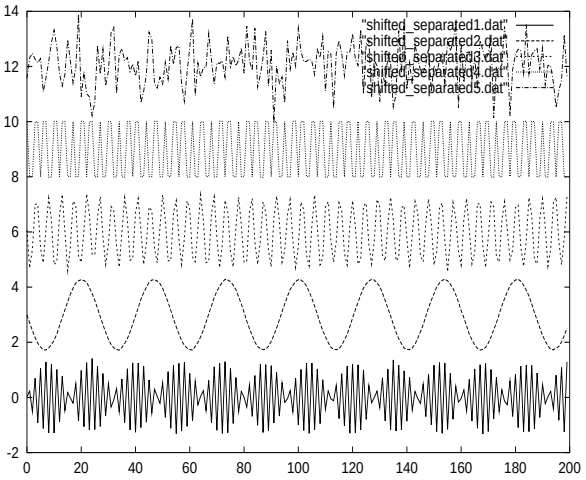

Fig. 4 Separated waveforms by turbo α-ICA.

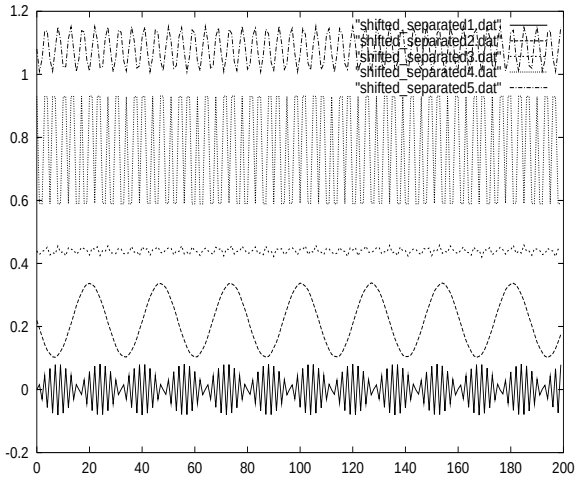

Fig. 7 Waveforms by orthogonal turbo α-ICA.

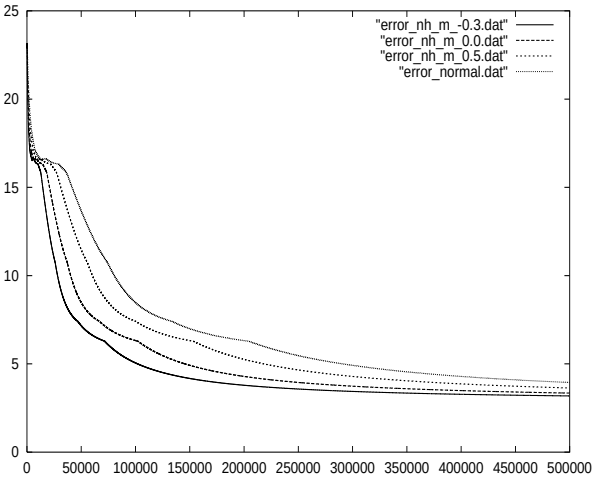

Fig. 5 Orthogonal momentum α-ICA.

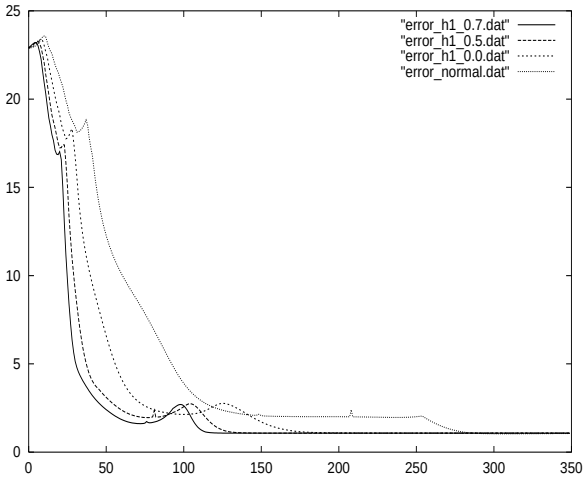

Fig. 8 Hybrid turbo α-ICA of type I.

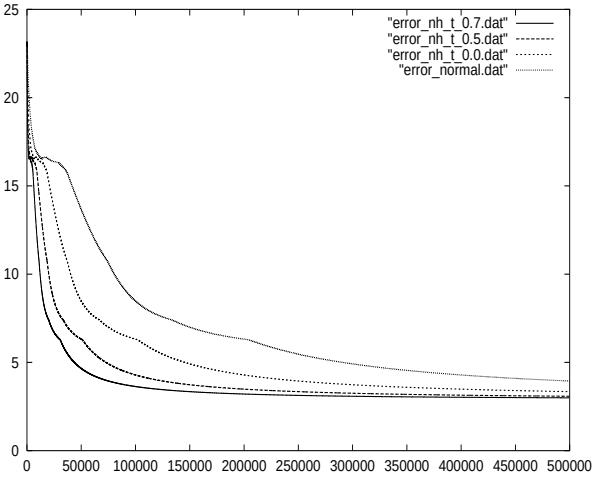

Fig. 6 Orthogonal turbo α-ICA.

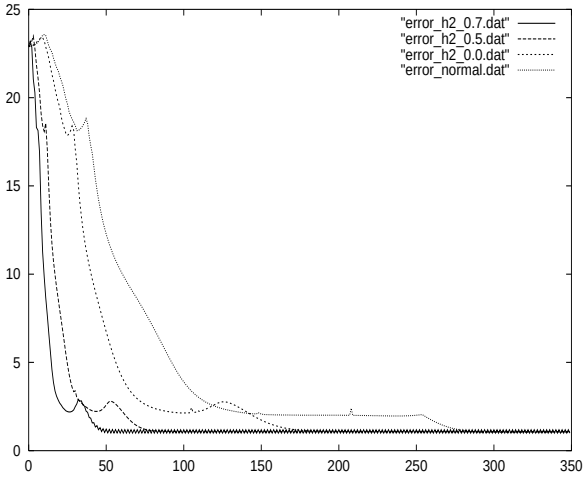

Fig. 9 Hybrid turbo α-ICA of type II.