

# An Improved Independent Component Analysis Algorithm Based on Artificial Immune System

Li-Yuan Chen and Chi-Jie Lu, *Member, IACSIT*

**Abstract**—Traditional independent component analysis (ICA) method based on FastICA algorithm faced two main disadvantages. One is that the order of the independent components (ICs) is difficult to be determined and the other is that the FastICA algorithm often leads to local minimum solution, and the suitable source signals are not isolated. To alleviate these problems, an improved ICA algorithm based on artificial immune system (AIS) (called AIS-ICA) is presented. AIS is an attractive heuristic technique and has many advantages over other heuristic techniques such as it can be easily implemented and has great capability of escaping local optimal solutions. The basic idea of the proposed AIS-ICA algorithm is to use AIS to determine the separating matrix of ICA. Simulation results from the artificial signal data illustrate the efficiency of the proposed AIS-ICA approach.

**Index Terms**—Independent component analysis, artificial immune system, signal separation, heuristic algorithm.

## I. INTRODUCTION

Blind source separation (BSS) is to separate the source from the received signals without any prior knowledge of the source signal. Problems related to BSS have become an active research area in the fields of statistical signal processing and unsupervised neural learning [1]-[3]. Independent component analysis (ICA) is one of the most used methods for BSS. The goal of ICA is to recover independent sources when given only sensor observations that are unknown linear mixtures of the unobserved independent source signals. It has been investigated extensively in image processing, financial time series data and statistical process control [1], [4]-[6].

For ICA, many effective algorithms have been proposed [1]-[3], the most used traditional algorithm is FastICA algorithm that uses the approximate Newtonian iteration algorithm. But in practical application, it often leads to local minimum solution and the suitable source signals are not isolated. Moreover, the order of the independent components (ICs) is difficult to be determined. These two problems are the main drawbacks of FastICA algorithm [7]-[9]. To overcome these disadvantages, an improved ICA algorithm based on artificial immune system (AIS) (called AIS-ICA) is presented.

AIS is an adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving [10], [11].

Manuscript received October 4, 2012; revised December 27, 2012. This work was supported in part by the National Science Council of the Republic of China under Grant No. NSC 101-2221-E-231-006.

Li-Yun Chen and Chi-Jie Lu are with Department of Industrial Management, Chien Hsin University of Science and Technology, Zhong-Li 320, Taoyuan, Taiwan (e-mail: jerrylyu@uch.edu.tw; chijie.lu@gmail.com).

They are one among many types of algorithm inspired by biological systems, including evolutionary algorithms, swarm intelligence, neural networks and membrane computing. AIS are bio-inspired algorithms that take their inspiration from the human immune system. Within AIS, there are many different types of algorithm, and research to date has focused primarily on the theories of immune networks, clonal selection and negative selection. It is based on natural immune system principles, and it can offer strong and robust information processing capabilities for solving complex problems. There are various applications of AIS, and they include data analysis, scheduling, classification, fault detection and security of information systems [10]-[13]. Since AIS has many advantages over other heuristic techniques such as it can be easily implemented and has great capability of escaping local optimal solutions [11], it is used in this study to develop the AIS-ICA algorithm.

The basic idea of the proposed AIS-ICA method is to use AIS to replace the Newtonian iteration in FastICA algorithm to find the separating matrix of ICA. Then, the independent components can be extracted and the order of ICs can be determined according to the AIS results. The artificial signal data are used as illustrative examples. Experimental results showed that the proposed AIS-ICA approach is an effective ICA algorithm.

The remainder of this paper is organized as follows: Section 2 gives an overview of ICA and AIS. Section 3 describes the proposed AIS-ICA algorithm. The simulation results are shown in Section 4. The conclusions are drawn in Section 5.

## II. METHODOLOGY

### A. ICA

Let  $\mathbf{X} = [\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m]^T$  be a multivariate data matrix of size  $m \times n$ ,  $m \leq n$ , consisting of observed random variables  $\tilde{x}_i$  of size  $1 \times n$ ,  $i = 1, 2, \dots, m$ . In the basic ICA model, the matrix  $\mathbf{X}$  can be modeled as

$$\mathbf{X} = \mathbf{A}\mathbf{S} = \sum_{i=1}^m \tilde{a}_i \tilde{s}_i \quad (1)$$

where  $\tilde{a}_i$  is the  $i^{\text{th}}$  column of unknown mixing matrix  $\mathbf{A}$  of size  $m \times m$ ;  $\tilde{s}_i$  is the  $i^{\text{th}}$  row of source matrix  $\mathbf{S}$  of size  $m \times n$ . The vectors  $\tilde{s}_i$  are unknown latent sources (variables) that cannot be directly observed from the observed variables  $\tilde{x}_i$ . The ICA model aims at finding an  $m \times m$  de-mixing matrix  $\mathbf{W}$  such that

$$\mathbf{B} = [\tilde{b}_i] = \mathbf{W}\mathbf{X} = [\tilde{w}_i \mathbf{X}], \quad (2)$$

where  $\tilde{b}_i$  is the  $i^{th}$  row of the matrix  $\mathbf{B}$ ,  $i=1, 2, \dots, m$ . The vectors  $\tilde{b}_i$  must be as statistically independent as possible, and are called independent components (ICs). The ICs are used to estimate the latent variables  $\tilde{s}_i$ . The vector  $\tilde{w}_i$  in eq. (2) is the  $i^{th}$  row of the de-mixing matrix  $\mathbf{W}$ ,  $i=1, 2, \dots, m$ . It is used to transform the observed multivariate matrix  $\mathbf{X}$  to generate the corresponding IC, i.e.  $\tilde{b}_i = \tilde{w}_i \mathbf{X}$ ,  $i=1, 2, \dots, m$ .

The ICA modeling is formulated as an optimization problem by setting up the measure of independence of ICs as an objective function and using some optimization techniques to solve for the de-mixing matrix  $\mathbf{W}$  [1]-[3]. In general, the ICs are obtained by using the de-mixing matrix  $\mathbf{W}$  to multiply the matrix  $\mathbf{X}$ . The de-mixing matrix  $\mathbf{W}$  can be determined using an unsupervised learning algorithm with the objective of maximizing the statistical independence of ICs. And the statistical independence of ICs can be measured in terms of their non-Gaussian properties [1].

Normally, non-Gaussianity can be verified by two common statistics: kurtosis and negentropy. The kurtosis of a random variable  $\tilde{b}$ , fourth-order cumulant, is classically defined by

$$kurt(\tilde{b}) = E(\tilde{b}^4) - 3(E(\tilde{b}^2))^2. \quad (3)$$

If variable  $\tilde{b}$  is assumed to be zero mean and unit variance, the right-hand side simplifies to  $E(\tilde{b}^4) - 3$ . This shows that kurtosis is simply a normalized version of the fourth moment  $E(\tilde{b}^4)$ . For a Gaussian  $\tilde{b}$ , the fourth moment equals  $3(E(\tilde{b}^2))^2$ . Thus, kurtosis is zero for a Gaussian random variable and non-zero for most non-Gaussian random variables.

Unlike kurtosis, negentropy is determined according to the information quantity of (differential) entropy. Entropy is a measure of the average uncertainty in a random variable. The differential entropy  $H$  of random variable  $\tilde{b}$  with density  $f(\tilde{b})$  is defined as  $H(\tilde{b}) = -\int p(\tilde{b}) \log p(\tilde{b}) d\tilde{b}$ . According to a fundamental result of information theory, a Gaussian variable will have the highest entropy value among a set of random variables with equal variance [1]. For obtaining a measure of non-Gaussianity, the negentropy  $J$  is defined as follows:

$$J(\tilde{b}) = H(\tilde{b}_{gauss}) - H(\tilde{b}) \quad (4)$$

where  $\tilde{b}_{gauss}$  is a Gaussian random vector of the same covariance matrix as  $\tilde{b}$ .

The negentropy is always non-negative and is zero if and only if  $\tilde{b}$  has a Gaussian distribution. Since negentropy is very difficult to compute, an approximation of negentropy is proposed as follows [1]:

$$J(\tilde{b}) \approx [E\{G(\tilde{b})\} - E\{G(\tilde{\delta})\}]^2 \quad (5)$$

where  $\tilde{\delta}$  is a Gaussian variable of zero mean and unit variance, and  $\tilde{b}$  is a random variable with zero mean and unit variance.  $G$  is a nonquadratic function, and is given

by  $G(\tilde{b}) = \tilde{b}^4$  in this study. The steps of the FastICA algorithm are shown in Fig. 1 [1].

**Step 1: Data centering.** The mean of the observed mixed signal data  $\mathbf{X}$  is computed and the mean is subtracted from the observed data set to make it zero mean.

$$\mathbf{Xc} \leftarrow \mathbf{X} - E\{\mathbf{X}\}$$

**Step 2: Whitening.** The covariance matrix  $\mathbf{cov} \mathbf{X}$  of the centered data  $\mathbf{Xc}$  is computed. The eigenvalue decomposition of  $\mathbf{cov} \mathbf{X}$  is performed. If  $D$  is the eigenvalue matrix and  $E$  is the eigenvector matrix then

$$\mathbf{Z} = D^{-1/2} E * \mathbf{Xc}$$

**Step 3: Fixed-point iteration for one unit.** The fast ICA algorithm for one unit estimates one row of the demixing matrix  $\mathbf{w}$  as a vector  $\mathbf{w}^T$  that is an extremum of contrast functions. Estimation of  $\mathbf{w}$  proceeds iteratively with following steps until a convergence as stated below is achieved.

3.1 Choose an initial random vector  $\mathbf{w}$  of unit norm.

3.2  $\mathbf{w}_p \leftarrow E\{\mathbf{z}g(\mathbf{w}_p^T \mathbf{z})\} - E\{g'(\mathbf{w}_p^T \mathbf{z})\} \mathbf{w}$ , where

$$\begin{cases} g_1(\mathbf{w}_p^T \mathbf{z}) = \tanh(a\mathbf{w}_p^T \mathbf{z}); \\ g'_1(\mathbf{w}_p^T \mathbf{z}) = a(1 - \tanh^2(a\mathbf{w}_p^T \mathbf{z})) \quad 1 \leq a \leq 2 \end{cases}$$

$$\begin{cases} g_2(\mathbf{w}_p^T \mathbf{z}) = \exp(-(\mathbf{w}_p^T \mathbf{z})^2 / 2) \\ g'_2(\mathbf{w}_p^T \mathbf{z}) = (1 - (\mathbf{w}_p^T \mathbf{z})^2) \exp(-(\mathbf{w}_p^T \mathbf{z})^2 / 2) \end{cases}$$

$$\begin{cases} g_3(\mathbf{w}_p^T \mathbf{z}) = (\mathbf{w}_p^T \mathbf{z})^3 \\ g'_3(\mathbf{w}_p^T \mathbf{z}) = 3(\mathbf{w}_p^T \mathbf{z})^2 \end{cases}$$

3.3  $\mathbf{w}_p^* \leftarrow \mathbf{w}_p / \|\mathbf{w}_p\|$  where  $\|\mathbf{w}_p\|$  is the norm of  $\mathbf{w}$

3.4 If  $|\mathbf{w}(\text{old}) - \mathbf{w}(\text{new})| \leq \epsilon$  is not satisfied then go back to Step 3 where  $\epsilon$  is a convergence parameter ( $\sim 10^{-4}$ ) and  $\mathbf{w}(\text{old})$  is the value of  $\mathbf{w}$  before it's replacement by the newly calculated value  $\mathbf{w}(\text{new})$

**Step 4: Evaluation of second independent component.** To estimate the other ICs, Step 3 of the algorithm is repeated for getting weight vectors  $w_i$ ,  $i = 2, 3, \dots, n$ . To prevent different vectors from converging to the same optimum and hence the same IC, the weight vectors are decorrelated using Gram-Schmidt like orthogonalization. When  $p$  vectors  $w_i$ ,  $i = 2, 3, \dots, p$  have been estimated, Step 3 is run for  $w_{p+1}$  and after every iteration step the following iteration steps are performed.

$$w_{p+1} = w_{p+1} - \sum_{j=1}^p (w_{p+1}^T w_j) w_j \quad (6)$$

$$w_{p+1} = \frac{w_{p+1}}{\sqrt{w_{p+1}^T w_{p+1}}} \quad (7)$$

Fig. 1. Steps of FastICA algorithm.

## B. AIS

Artificial Immune algorithm based on clonal selection principle (CLONALG) is used to optimize functions. CLONALG is a population based algorithm and its only variation operator is mutation. The main search power of CLONALG relies on this mutation operator and therefore, such hyper-mutation operator is the efficiency deciding factor of this technique [14], [15].

The first step of CLONALG is initialization. In this step, antigens are represented by the value of the objective function  $f(x)$  that we want to optimize (minimize or maximize) and antibodies are represented by the variables of the problem ( $x_i, i = 1, 2, \dots, s$ ) which are potential solutions. An initial population of  $N$  antibodies is created randomly.

The second step is cloning. In this stage, antibody's affinity corresponds to the evaluation of the objective function given by the antigen. According to the affinity or fitness, the antibodies are cloned; the best antibody being cloned the most and worst being cloned the least number of times. The number of clones generated from the  $n$  selected antibodies is given by

$$T_c = \sum \text{round}(\beta * p/i), i = 1, 2, \dots, n$$

where  $T_c$  is the total number of clones,  $\beta$  is a multiplier factor and  $p$  is the population size of the antibodies.

In the final step Hyper-mutation and selection, the clones are mutated in inverse proportion to their affinity; the best antibody's clones are mutated lesser and worst antibody's clones are mutated most. The clones are then evaluated along with their original antibodies out of which the best  $G$  antibodies are selected for the next iteration. The mutation can be uniform, Gaussian or exponential. The pseudo code of CLONALG is shown in Fig. 2 [14], [15].

1. Generate  $N$  antibodies randomly.
2. Determine the affinity or fitness of each antibody.
3. Select the  $n$  highest affinity antibodies.
4. The  $n$  selected antibodies will be cloned proportionally to their affinities, generating a repertory  $C$  of clones. Higher the affinity is, higher becomes the number of clones generated for that selected antibody.
5. The clone from  $C$  are subjected to a hyper-mutation process inversely proportional to their antigenic affinity; higher the affinity, smaller is the mutation rate.
6. Determine the affinity of the mutated clones.
7. From this set mutated clones and antibodies, select the highest affinity to compose the new antibodies population.
8. Replace the  $d$  lowest affinity antibodies by new individuals generated at random.
9. Continue until terminating condition.

Fig. 2. Pseudo code of CLONALG.

### III. PROPOSED AIS-ICA ALGORITHM

In the FastICA model, because of using the Newtonian iteration, it often leads to local minimum solution. For these reasons, the AIS method is introduced to replace the Newtonian iteration for solving the problem of finding the maximum non-gaussianity of  $\mathbf{W}\mathbf{x}$ . According to the abovementioned pseudo code of CLONALG and steps of the FastICA algorithm, the steps of the proposed AIS-ICA algorithm are summarized and shown in Fig. 3.

1. Data centering and whitening.
2. Let  $p=1$ ,  $h$ =number of ICs,  $Ab$  are Antibodies,  $i=1, 2, \dots, m$ , and used to represent vector  $\mathbf{w}_i$ . Make the Eq.

- (5), i.e. Maximize  $J(\tilde{\mathbf{b}})$ , the objective function of the AIS. Initialize the Antibodies randomly.
3. Let Eq. (5) be the fitness function. Calculate the fitness value/affinity of each Antibody.
4. Find the best antibody  $Ab^*$  with the maximum fitness value.
5. Select the  $n$  highest affinity antibodies  $b_{A_i}, i=1, \dots, n$ .
6. The selected antibodies  $b_{A_i}$  are cloned according to their affinities and use to generate mutated clones  $C_{A_i}, i=1, 2, \dots, C$ . Calculate the affinity of the mutated clones  $C_{A_i}$ .
7. Compose the  $d$  new antibodies population from  $b_{A_i}$  and  $C_{A_i}$ . Replace the  $d$  lowest affinity antibodies by new antibodies randomly.
8. Determine whether stopping criteria (i.e. the maximum iteration number  $s_t$ ) is satisfied. If not, go back to step 3; else, the one-unit algorithm is ended. Thus  $\mathbf{w}_i$  is obtained and go to step 9.
9. Calculate the new antibodies according to Eqs. (6) and (7).
10. Let  $p=p+1$ . If  $p < h$ , go back to step 3 until get demixing matrix  $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_h)^T$ .
11. Get the source matrix  $\mathbf{B}$  according to Eq. (2).

Fig. 3. Steps of the proposed AIS-ICA algorithm.

### IV. SIMULATION RESULTS

In this section, the proposed AIS-ICA algorithm and FastICA will be used to separate the mixed signal. The artificially generated mixtures are used to evaluate the effectiveness of the proposed AIS-ICA algorithm compared to FastICA algorithm. Six artificial signals that have the following distribution [16] are used as source signals and shown in Fig. 4.

- (a) Modulated sinusoid:  
 $P(t) = 2 * \sin(t/149) * \cos(t/8) + 0.2 * \text{rand}()$
- (b) Square wave :  
 $Q(t) = \text{sign}(\sin(12*t + 9*\cos(2/29))) + 0.1 * \text{rand}()$
- (c) Sawtooth:  
 $R(t) = (\text{rem}(t, 79) - 17) / 23 + 0.1 * \text{rand}()$
- (d) Impulsive curve:  
 $S(t) = ((\text{rem}(t, 23) - 11) / 9)^5 + 0.1 * \text{rand}()$
- (e) Exponential:  
 $U(t) = 5 * \exp(-t/121) * \cos(37*t) + 0.1 * \text{rand}()$
- (f) Spiky noise:  
 $V(t) = ((\text{rand}() * 2 - 1) * \log(\text{rand}()))$

where the  $\text{rem}$  function returns a result that is between 0 and  $\text{sign}(e) * \text{abs}(z)$ . If  $z$  is zero,  $\text{rem}$  returns NaN (not a number). The  $\text{rand}$  function generates a set of uniformly distributed pseudo- random numbers.

According to a review of the literature [10], [11], the parameters of the AIS-ICA are set as:  $h=6$ ;  $Ab=10$ ; the highest affinity antibodies  $n=5$ ; mutated clones  $C=5$ ; the maximum iteration number  $s_t=50$ .

The source signals are mixed by a known 6 by 6 mixing matrix (as shown in Table 1) to generate the mixed signals which are depicted in Fig. 5. The mixed singles are passed through FastICA and the proposed AIS-ICA algorithms. The separating outputs are given in Fig. 6. From Fig. 6 (b), it can

be seen that FastICA couldn't separate the independent components clearly, whereas the proposed AIS-ICA separates them clearly (as seen in Fig. 6(a)). That is, it can be seen that the source signals separated by the AIS-ICA algorithm are more accurate than those separated by the FastICA algorithm since the AIS-ICA algorithm avoids the local minimum resolution.

The order of the ICs can be determined by the proposed AIS-ICA algorithm. Fig. 7 shows the separating results of the AIS-ICA and FastICA algorithms by three rounds. From Fig. 7 (a1)-(a3), it can be seen that the orders of the separating results among the three rounds are identical. The first to sixth ICs, respectively, represent the Spiky noise, Exponential, Impulsive curve, Square wave, Sawtooth and Modulated sinusoid signals. However, it can be observed from Fig. 7(b1)-(b3) that the order of ICs cannot be determined by FastICA algorithm. It is worthy to note that, among the three rounds, the AIS-ICA algorithms consistently provides better separating results than that of FastICA algorithm.

TABLE I. THE MIXING MATRIX.

0.2	0.5	-0.6	0.8	0.5	-0.2
0.7	-0.2	-0.3	0.8	0.1	0.7
-0.2	0.8	0.1	-0.4	0.9	0.1
-0.1	0.2	-0.6	-0.3	-0.8	-0.1
0.7	0.6	0.1	0.4	0.2	0.3
-0.1	0.1	-0.1	0.1	-0.1	0.1

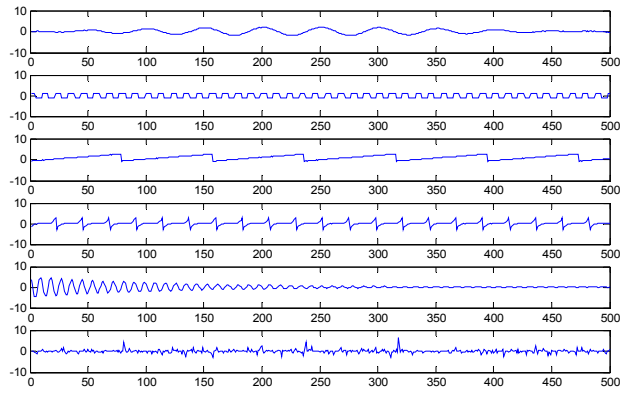


Fig. 4. Six source signals.

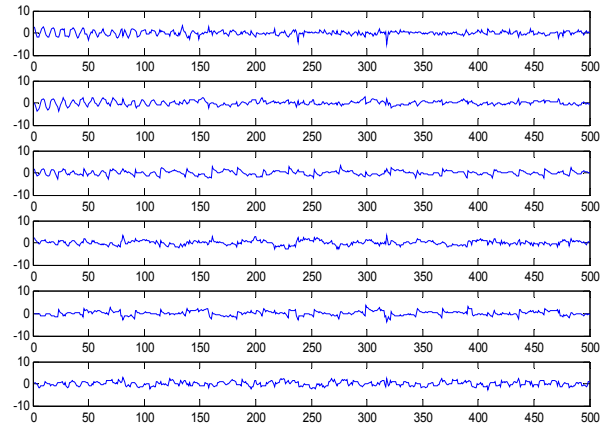
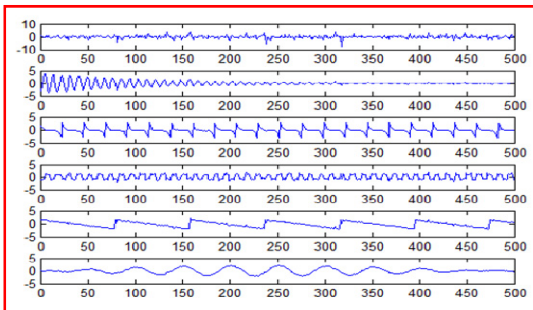
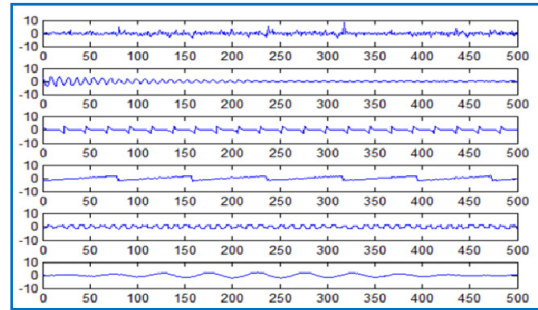


Fig. 5. The mixed signals.

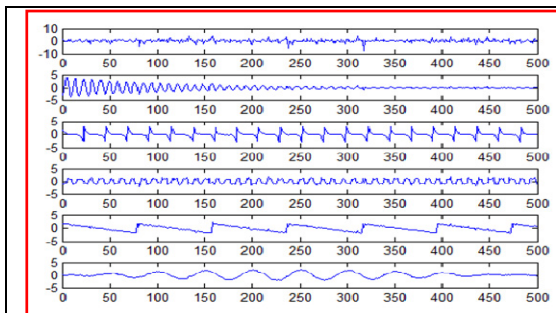


(a) AIS-ICA

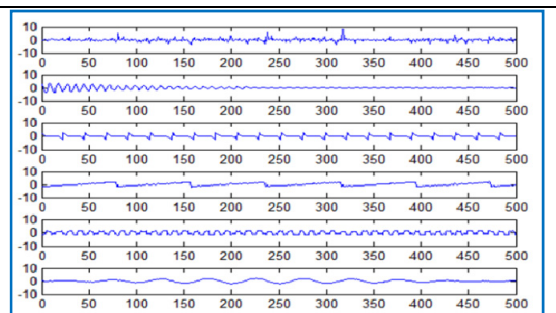


(b) FastICA

Fig. 6. ICs by AIS-ICA and FastICA algorithms.



(a1) 1<sup>st</sup> round of AIS-ICA



(b1) 1<sup>st</sup> round of FastICA

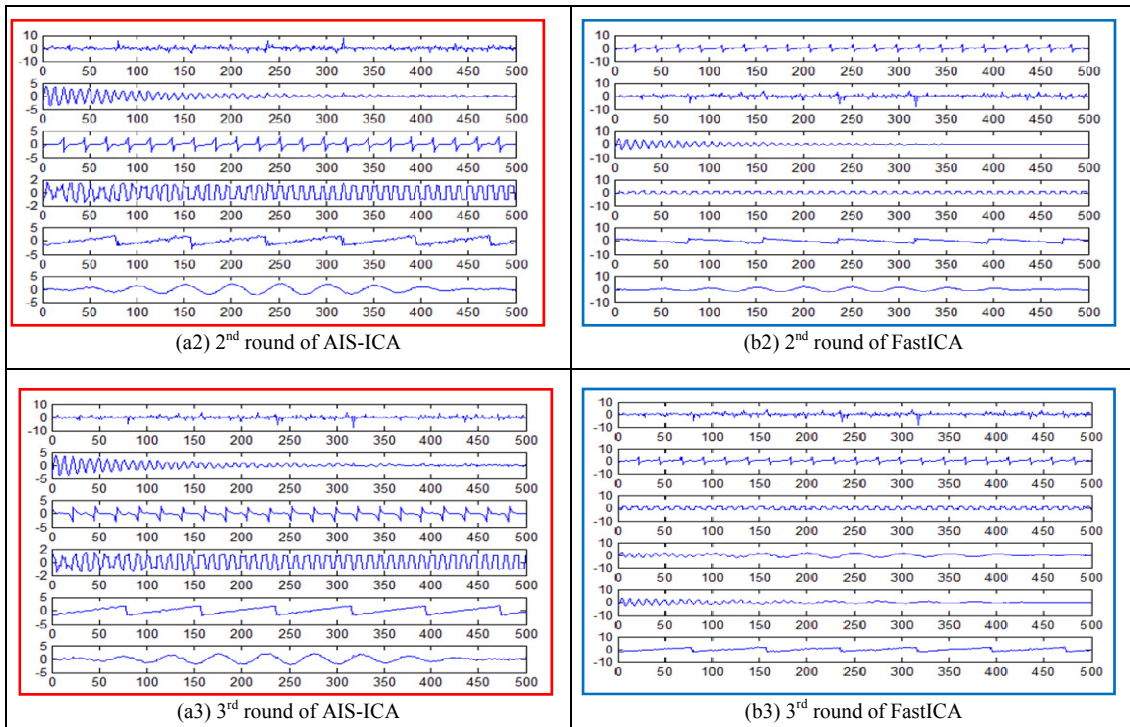


Fig. 7. The separating results of the AIS-ICA and FastICA algorithms by three rounds.

## V. CONCLUSION

ICA technique separates mixed signals blindly without any information of the mixing system. FastICA is the most popular gradient based ICA algorithm. However, the order of the ICs is difficult to be determined and easy to obtain local minimum solution are two main drawbacks of the FastICA algorithm. In order to overcome these two disadvantages, this paper presented an improved ICA algorithm based on AIS. In the proposed AIS-ICA algorithm, the AIS method is used to determine the demixing matrix of ICA. Experimental results from the artificial signal data showed that the AIS-ICA algorithms provides better separating results than that of FastICA algorithm. The order of ICs can be determined by the proposed AIS-ICA algorithm.

## REFERENCES

- [1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, New York: John Wiley & Sons, 2001.
- [2] V. David and A. Sanchez, "Frontiers of research in BSS/ICA," *Neurocomputing*, vol. 49, pp. 7-23, 2002.
- [3] A. Cichocki, S. I. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, New York: John Wiley & Sons, 2002.
- [4] C. J. Lu, T. S. Lee, and C. C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Systems*, vol. 47, pp. 115-125, 2009.
- [5] C. J. Lu, Y. E. Shao, and B. S. Li, "Mixture control chart patterns recognition using independent component analysis and support vector machine," *Neurocomputing*, vol. 74, pp. 1908-1914, 2011.
- [6] C. J. Lu, "An independent component analysis-based disturbance separation scheme for statistical process monitoring," *Journal of Intelligent Manufacturing*, vol. 23, pp. 561-573, 2012.
- [7] F. Nian, W. Li, X. Sun, and M. Li, "An Improved Particle Swarm Optimization Application to Independent Component Analysis," in *Proc. 2009 International Conference on Information Engineering and Computer Science*, Wuhan, China, 2009, pp. 1-4.
- [8] L. Xie and J. Wu, "Global optimal ICA and its application in MEG data analysis," *Neurocomputing*, vol. 69, pp. 2438-2442, 2006.
- [9] H. Zhang, C. Guo, Z. Shi, and E. Feng, "A new constrained fixed-point algorithm for ordering independent components," *Journal of Computational and Applied Mathematics*, vol. 220, pp. 548-558, 2008.
- [10] L. N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, New York: Springer-Verlag, 2002.
- [11] E. Harta and J. Timmis, "Application areas of AIS: The past, the present and the future," *Applied Soft Computing*, vol. 8, pp. 191-201, 2008.
- [12] Y. T. Chong and C. H. Chen, "Management and Forecast of Dynamic Customer Needs: An Artificial Immune and Neural System Approach," *Advanced Engineering Informatics*, vol. 24, pp. 96-106, 2010.
- [13] M. Mobini, Z. Mobini, and M. Rabbani, "An Artificial Immune Algorithm for the project scheduling problem under resource constraints," *Applied Soft Computing*, vol. 11, pp. 1975-1982, 2011.
- [14] L. N. De Castro and F. J. Von Zuben, "The clonal selection algorithm with engineering applications," in *Proceedings of GECCO 2000*, Las Vegas, USA, 2000, pp. 36-39.
- [15] L. N. De Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *Evolutionary Computation*, vol. 6, pp. 239-251, 2002.
- [16] M. K. Nath, "Independent component analysis of real data," in *Proceedings of 2009, International Conference on Advances in Pattern Recognition*, Kolkata, India, 2009, pp. 149-152.



**Li-Yun Chen** is an assistant professor in the Department of Industrial Management at Chien Hsin University of Science and Technology, Taiwan. He got his Ph.D. in Industrial Engineering and Management from Yuan Ze University, Taiwan.



**Chi-Jie Lu** is an associate professor in the Department of Industrial Management at Chien Hsin University of Science and Technology, Taiwan. He has published articles in various journals, including Pattern Recognition, Decision Support Systems, Neurocomputing, International Journal of Production Economics and European Journal of Operational Research.