

# A Class of Sequential Blind Source Separation Method in Order Using Swarm Optimization Algorithm

Wang Rongjie<sup>1,2</sup> · Zhan Yiju<sup>3</sup> · Zhou Haifeng<sup>1</sup>

Received: 23 February 2015 / Revised: 24 October 2015 / Accepted: 26 October 2015 /  
Published online: 12 November 2015  
© Springer Science+Business Media New York 2015

**Abstract** We consider the problem of sequential, blind source separation in some specific order from a mixture of sub- and sup-Gaussian sources. Three methods of separation are developed, specifically, kurtosis maximization using (a) particle swarm optimization, (b) differential evolution, and (c) artificial bee colony algorithm, all of which produce the separation in decreasing order of the absolute kurtosis based on the maximization of the kurtosis cost function. The validity of the methods was confirmed through simulation. Moreover, compared with other conventional methods, the proposed method separated the various sources with greater accuracy. Finally, we performed a real-world experiment to separate electroencephalogram (EEG) signals from a super-determined mixture with Gaussian noise. Whereas the conventional methods separate simultaneously EEG signals of interest along with noise, the result of this example shows the proposed methods recover from the outset solely those EEG signals of interest. This feature will be of benefit in many practical applications.

---

✉ Wang Rongjie  
Roger811207@163.com  
Zhan Yiju  
Yjzhan@zd100com.cn  
Zhou Haifeng  
Zh216@163.com

<sup>1</sup> Marine Engineering Institute, Jimei University, Xiamen 361021, China

<sup>2</sup> Fujian Provincial Key Laboratory of Naval Architecture and Ocean Engineering, Xiamen 361021, China

<sup>3</sup> School of Engineering, Sun Yat-sen University, Guangzhou 510006, China

**Keywords** Blind source separation · Kurtosis · Sub-Gaussian distribution · Sup-Gaussian distribution · Swarm optimization algorithm

## 1 Introduction

Recently, blind source separation (BSS), or blind signal separation, has received considerable attention in the fields of digital communications, biomedical engineering, speech processing, image processing, and robot navigation [2, 7, 9, 15, 19, 20]. The objective of BSS is to extract the original source signals from their mixtures using only the information of the observed signal with no or very limited knowledge about the source signals. Over the past few years, many methods have been developed to solve the BSS problem. They can be largely classified into two approaches. In the context of BSS, all the sources are simultaneously recovered or estimated from the observations, such as those based on the joint diagonalization cost function and the natural gradient [26], those based on temporal predictability and the differential search algorithm [3], and the nonlinear contract function-based BSS [31]. However, the different neurons of these BSS methods [3, 26] may converge to the same maximization, i.e., extracting the same sources from mixtures. The other class of algorithm is called blind source extraction (BSE). BSE is an extension of the BSS technique. Compared with these methods [3, 26, 31], the objective of BSE is to recover a single source or a part of a source from the observations one by one [6, 24], and this leads to low computation complexity and greater flexibility. This class of algorithms has a disadvantage in that the choice of nonlinearity has taken several directions because of the conflict between boundedness and analyticity. In this respect, Liouville's theorem states that if a function is analytic and bounded in the plane, then it is a constant. Conventional BSS approaches do not ensure separation of source signals with a specified order in accordance with their stochastic properties. There are actually three types of source signals in [10, 16]; they are known as standard normal, sup-Gaussian, and sub-Gaussian. The kurtosis in [27, 32] is used to measure signals of the different distributions. This paper introduces a class of BSS methods based on swarm optimization, which is able to separate sup and sub-Gaussian sources from mixtures in decreasing order according to the absolute value of kurtosis. However, it requires the maximization of the cost function, which is nonlinear and multimodal. Indeed, the swarm optimization algorithm has yielded excellent performances in similar problems [17, 18, 21]. For these reasons, we believe it is necessary and certainly beneficial to study the application of swarm optimization algorithms in sequential BSS.

The paper is organized as follows. Sequential BSS is formulated in Sect. 2.1. Section 2.2 provides the description of the employed swarm intelligent algorithms: the particle swarm optimization, the differential evolution approach, and the artificial bee colony algorithm. Section 2.3 proposes a swarm intelligent optimization-based BSS. The performance of the proposed method is numerically evaluated in Sect. 3. Section 4 is devoted to concluding remarks.

## 2 Sequential Blind Source Separation Method in Order Using Swarm Optimization Algorithm

### 2.1 Problem Formulation

Consider the  $n$  unobservable components of a zero-mean vector  $\mathbf{s}(t) = [s_1(t), s_2(t), \dots, s_n(t)]^T$ , each component being mutually and statistically independent; here superscript  $T$  denotes transposition and  $\mathbf{s}(t) \in \mathbf{R}^n$ . The available sensor vector  $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_m(t)]^T$ , where  $m$  is the number of sensors and  $\mathbf{x}(t) \in \mathbf{R}^m$ , is given by

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

where  $\mathbf{A} \in \mathbf{R}^{m \times n}$  is a non-singular and unobservable matrix with a nonzero determinant, and its rank is  $n$ . Here,  $t = 0, 1, \dots, N-1$  is the instant time of sampling. The objective of sequential BSS is to recover or estimate only a single source from the observations. The separated signal can be given by

$$y(t) = \mathbf{w}^T \mathbf{x}(t) = \mathbf{w}^T \mathbf{A}\mathbf{s}(t), \quad (2)$$

where the extracted weighted vector  $\mathbf{w}$  is a  $m$  column vector. Let  $\mathbf{g} = \mathbf{w}^T \mathbf{A}$ ; if only one element of  $\mathbf{g}$  is nonzero while the others are zero, then  $y(t)$  is a recovery or estimate of  $\mathbf{s}(t)$  with the same site being nonzero in  $\mathbf{g}$ .  $\mathbf{s}(t)$  is recovered using (2)  $n$  times.

Denote  $\mathbf{B}$  as a whitening matrix and selected so as to sphere the observations, then  $\mathbf{z}(t) = \mathbf{B}\mathbf{x}(t)$ . We estimate each source,  $y(t)$ , separately finding a vector  $\mathbf{w}$  such that

$$y(t) = \mathbf{w}^T \mathbf{z}(t). \quad (3)$$

Without loss of generality, we make the typical assumption that the source signals have unit variance, i.e.,  $E[\mathbf{s}(t)\mathbf{s}(t)^T] = \mathbf{I}$ . It also means  $E[y(t)^2] = 1$ , and hence we can rewrite  $E[y(t)^2]$  as

$$\begin{aligned} E \left[ \mathbf{w}^T \mathbf{z}(t) \mathbf{z}(t)^T \mathbf{w} \right] &= \mathbf{w}^T E \left[ \mathbf{z}(t) \mathbf{z}(t)^T \right] \mathbf{w} \\ &= 1 \end{aligned} \quad (4)$$

From (4), we can obtain  $\|\mathbf{w}\|^2 = 1$ , where  $\|\cdot\|$  denotes the Frobenius norm.

In [27, 32], the kurtosis of a zero-mean random variable  $y(t)$  is defined as

$$K(y) \triangleq \text{cum}[y(t), y(t), y(t), y(t)]. \quad (5)$$

For Gaussian signals, the kurtosis is zero; for non-Gaussian signals, the kurtosis is nonzero. The kurtosis can be both positive and negative. Random variables that have positive kurtosis are called sup-Gaussian, and those with negative kurtosis are called sub-Gaussian. Non-Gaussianity is measured using the absolute value of kurtosis or the square of kurtosis in [17, 18, 21]. We use (5) to find an optimal  $\mathbf{w}$ ,

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{arg\,max}} |K(y)| \quad \text{s.t. } \|\mathbf{w}\|^2 = 1. \tag{6}$$

To justify the use of the optimization criterion in (6), we state the following theorem and give its proof.

**Theorem** Assume that  $\mathbf{z}(t) = \mathbf{BAs}(t)$  where  $\mathbf{B}$  is the whitening matrix. Then, the local maxima (or minima) of  $K(y)$  under the constraint  $\|\mathbf{w}\|^2 = 1$  such that the corresponding independent components  $s_i(t)$  satisfy  $K(s_i) > 0$  (or  $K(s_i) < 0$ ).

*Proof* We define a linear transformation  $\mathbf{q} = (\mathbf{BA})^T \mathbf{w}$ . Without loss of generality, it is sufficient to analyze the stability of point  $\mathbf{q}_i = [0, \dots, 0, \delta_i, 0, \dots, 0]^T$  with  $|\delta_i| = 1$ . At  $\mathbf{q}_i$ , we have  $y(t) = \mathbf{q}_i^T \mathbf{s}(t) = \delta_i s_i(t)$ . Substituting it into (5), we then obtain

$$K(\mathbf{q}) = E \left\{ [\mathbf{q}^T \mathbf{s}(t) \mathbf{s}(t)^T \mathbf{q}]^2 \right\} - 3 \left\{ \mathbf{q}^T E[\mathbf{s}(t) \mathbf{s}(t)^T] \mathbf{q} \right\}^2. \tag{7}$$

Evaluating the first-order gradients and second order of  $s_i(t)$  at optimal point  $\mathbf{q}_i$  and using the independence of source signals, we obtain

$$\left. \frac{\partial K(\mathbf{q})}{\partial \mathbf{q}} \right|_{\mathbf{q}=\mathbf{q}_i} = [0, \dots, 0, 4K(s_i)\delta_i^3, 0, \dots, 0]^T, \tag{8}$$

$$\left. \frac{\partial K(\mathbf{q})}{\partial \mathbf{q} \partial \mathbf{q}^T} \right|_{\mathbf{q}=\mathbf{q}_i} = \begin{bmatrix} 0, \dots, 0, & \dots, & 0, \dots, 0 \\ \vdots & & \vdots \\ 0, \dots, 0, & 12K(s_i)\delta_i^2, & 0, \dots, 0 \\ 0, \dots, 0, & \dots, & 0, \dots, 0 \\ \vdots & & \vdots \\ 0, \dots, 0, & \dots, & 0, \dots, 0 \end{bmatrix}. \tag{9}$$

Introducing a small perturbation  $\Delta \mathbf{q} = [\Delta q_1, \Delta q_2, \dots, \Delta q_m]^T$  to  $\mathbf{q}_i$ . The difference between  $K(\mathbf{q}_i + \Delta \mathbf{q})$  and  $K(\mathbf{q}_i)$  to second order in a Taylor series expansion is

$$\begin{aligned} & K(\mathbf{q}_i + \Delta \mathbf{q}) - K(\mathbf{q}_i) \\ &= \left[ \left. \frac{\partial K(\mathbf{q})}{\partial \mathbf{q}} \right|_{\mathbf{q}=\mathbf{q}_i} \right]^T \Delta \mathbf{q} + \left[ \frac{1}{2} \Delta \mathbf{q}^T \left. \frac{\partial K(\mathbf{q})}{\partial \mathbf{q} \partial \mathbf{q}^T} \right|_{\mathbf{q}=\mathbf{q}_i} \right] \Delta \mathbf{q} + o(\|\Delta \mathbf{q}\|^2) \\ &= 2K(s_i)[2\delta_i^3 \Delta q_i + 3\delta_i^2 \Delta q_i^2] + o(\|\Delta \mathbf{q}\|^2). \end{aligned} \tag{10}$$

From the constraint  $\|\mathbf{q} + \Delta \mathbf{q}\|^2 = 1$ , we obtain  $2\delta_i \Delta q_i = -\|\Delta \mathbf{q}\|^2$ , which implies that the higher-order terms in (10) are  $o(\|\Delta \mathbf{q}\|^2)$  and can be neglected. Through a substitution, (10) can be simplified as

$$K(\mathbf{q}_i + \Delta \mathbf{q}) - K(\mathbf{q}_i) = -2K(s_i) \|\Delta \mathbf{q}\|^2 + o(\|\Delta \mathbf{q}\|^2), \tag{11}$$

which clearly shows that  $\mathbf{q}_i$  is an extremum and of the type implied by the conditions of the theorem.

According to the above analysis, it is clear that the blind separation in (6) would extract the dominant source with the highest absolute kurtosis value. However, it may be difficult to implement, because it requires the maximization of the cost function, which is nonlinear and multimodal. Inspired by nature, the swarm optimization algorithm is a technique of dynamic systems research, which endeavors to develop new computational tools for problem solving by simulating and mimicking natural behaviors of organisms, and may result in the design of novel computational systems [4]. Indeed, some swarm optimization algorithms have obtained excellent performances in similar problems [1, 8, 12]. Because of these reasons, we believe it is necessary and certainly beneficial to study the application of swarm optimization algorithms in sequential BSS. In the subsequent sections, we describe the details of particle swarm optimization, differential evolution, and the artificial bee colony algorithm used in this work.

## 2.2 Swarm Optimization Algorithms

### 2.2.1 Particle Swarm Optimization

The particle swarm optimization (PSO) is a population-based, self-adaptive search optimization technique first introduced by Kennedy and Eberhart [14]. The PSO algorithm emulates swarm behavior and the individuals represent points in the multidimensional space. A particle represents a possible potential solution. The swarm initially has a population of random solutions. Each particle is given a random velocity and is flown through the problem space. Each particle has a memory and keeps track of its previous optimal position, denoted  $P_{\text{best}}$ , and corresponding fitness value. Each swarm remembers another value, denoted  $G_{\text{best}}$ , which is the optimal solution achieved so far by a swarm. The velocity  $V_l(d)$  and the position  $\theta_l(d)$  of the  $d$ th dimension of  $l$ th particle are updated by

$$V_l(d) = \omega_k V_l(d) + c_1 r_1(d)[P_{\text{best},l}(d) - \theta_l(d)] + c_2 r_2(d)[G_{\text{best}}(d) - \theta_l(d)], \quad (12)$$

$$\omega_k = \omega_{\text{max}} - \frac{(\omega_{\text{max}} - \omega_{\text{min}})}{k_{\text{max}}} k, \quad (13)$$

$$\theta_l(d) = \theta_l(d) + V_l(d), \quad (14)$$

where  $d = 1, 2, \dots, D$  with  $D$  the dimension of the candidate solution;  $P_{\text{best},l}(d)$  and  $G_{\text{best}}(d)$  are the  $d$ th dimensional positions of  $P_{\text{best}}$  of the  $l$ th particle and  $G_{\text{best}}$ , respectively;  $r_1(d)$  and  $r_2(d)$  are random numbers in the range  $[0, 1]$  and are different in different dimensions;  $k$  and  $k_{\text{max}}$  are the iteration number and total number of iterations, respectively,  $k = 1, 2, \dots, k_{\text{max}}$ ;  $c_1$  and  $c_2$  are acceleration coefficients, and  $\omega_k$  is the inertial weight, which plays the role of balancing the global search and local search. According to [29],  $\omega_{\text{max}} = 0.9$ ,  $\omega_{\text{min}} = 0.5$  and  $c_1 + c_2 = 4$ .  $l = 1, 2, \dots, N_p$ , with  $N_p$  the particle population size.

Table 1 presents the pseudocode for the PSO algorithm that we employed in this work.

**Table 1** Particle swarm optimization

---



---

```

Initialize: set  $k_{\max}, N_p, D$ . Generate initial  $\theta_l(d)$  randomly from  $[\theta_{\min}(d), \theta_{\max}(d)]$ ,
 $l=1,2, \dots, N_p, d=1, 2, \dots, D$ .  $\theta_{\max}(d)$  and  $\theta_{\min}(d)$  are maximum and minimum of
candidate solution  $d$ -th dimension, respectively.
Calculate fit function of each candidate solution, and select individual
particle's optimal position  $\mathbf{P}_{\text{best},l}$  and swarm particle's optimal position  $\mathbf{G}_{\text{best}}$ .
while the fit function criterion is not reached and  $k \leq k_{\max}$ 
  for  $l=1$  to  $N_p$ 
    for  $d=1$  to  $D$ 
      Update  $V_l(d)$  by (12) and (13);
      Update  $\theta_l(d)$  by (14);
    end
    Calculate fit function of each candidate solution;
    According to fit function value of individual particle, select  $\mathbf{P}_{\text{best},l}$ 
    from the set  $\{\text{previous } \mathbf{P}_{\text{best},l}, \theta_l(d)\}$ ;
  end
  According to fit function value of swarm, select  $\mathbf{G}_{\text{best}}$  from the set
   $\{\text{previous } \mathbf{G}_{\text{best}}, \text{current } \mathbf{P}_{\text{best},l}\}$ ;
   $k=k+1$ ;
end

```

---



---

### 2.2.2 Differential Evolution

The differential evolution (DE) was proposed by Storn and Price in 1995 [23,25]. It is a population-based and directed-search method. Let us assume that  $\theta_{l,k}$  ( $l = 1, 2, \dots, N_p$ ) are candidate solution vectors in iteration  $k$  ( $N_p =$  population size). Successive populations are generated by adding the weighted difference of two randomly selected vectors into a third randomly selected vector. For classical DE, the mutation, crossover, and selection operators are defined straightforwardly as follows.

*Mutation* for each vector  $\theta_{l,k}$  in iteration  $k$ , a mutant  $\mathbf{V}_{l,k}$  is defined by

$$\mathbf{V}_{l,k} = \theta_{r_1,k} + F(\theta_{r_2,k} - \theta_{r_3,k}), \tag{15}$$

where  $r_1, r_2$ , and  $r_3$  are mutually different random integer indices selected from  $[1, N_p]$ . Further,  $r_1 \neq r_2 \neq r_3$ .  $F \in [0, 2]$  is a real random constant that determines the amplification of the added differential variation of  $(\theta_{r_2,k} - \theta_{r_1,k})$ .

*Crossover* DE uses the crossover operation to generate new solutions by shuffling competing vectors and also to increase the diversity of the population. It defines the following trial vector:

$$\mathbf{U}_{l,k} = [U_{l1,k}, U_{l2,k}, \dots, U_{lD,k}], \tag{16}$$

where

$$U_{lj,k} = \begin{cases} V_{lj,k}, & \varphi \leq C_R \text{ or } j = k \\ \theta_{lj,k}, & \text{otherwise} \end{cases}, \tag{17}$$

$C_R \in [0, 1]$  is the predefined crossover rate constant,  $\varphi$  a uniform random number generator, and  $k \in [1, D]$  a random parameter index, chosen once for each  $l$  to ensure that at least one parameter is always selected from the mutated vector  $V_{lj,k}$ .

**Table 2** Differential evolution

---



---

```

Initialize: set  $k_{\max}$ ,  $N_p$ ,  $D$ ,  $C_R$ . Generate initial  $\theta_l(d)$  randomly from  $[\theta_{\min}(d)$ 
 $\theta_{\max}(d)]$ ,  $l=1,2, \dots, N_p$ ,  $d=1, 2, \dots, D$ . while the fit function criterion is not
reached and  $k \leq k_{\max}$ 
  for  $l=1$  to  $N_p$ 
    Update  $V_{l,k}$  by (15);
    Update  $U_{l,k}$  by (16)-(17);
    Calculate fit function of each  $U_{l,k}$ ;
    According to fit function values of  $\theta_{l,k}$  and  $U_{l,k}$ , select new  $\theta_{l,k+1}$  from
     $\theta_{l,k}$  and  $U_{l,k}$ ;
  end
   $k=k+1$ ;
end

```

---



---

*Selection* The approach must determine which vectors  $\theta_{l,k}$  and  $U_{l,k}$  should be a next iteration  $\theta_{l,k+1}$  depending on their fitness value. Table 2 outlines the pseudocode of DE.

### 2.2.3 Artificial Bee Colony Algorithm

The artificial bee colony (ABC) is an algorithm that simulates the foraging behavior of a bee colony, and can be used to solve the multidimensional optimization problems. A honey bee swarm consists of three kinds of bees: employed bees, onlooker bees, and scout bees [13]. In the ABC algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the object function of the associated solution. Let  $\theta(l)$  represent the  $l$ -th food source in the population,  $N_s$  is the population size of food source. The units of the original ABC can be explained as follows:

*Employed Bee Phase* At this stage, each employed bee generates a new food source  $\theta_{EB}(l)$  in the neighborhood of its present position by

$$\theta_{EB}(l, d) = \theta(l, d) + \varphi_{ld}[\theta(l, d) - \theta(r_1, d)], \quad (18)$$

where  $d = 1, 2, \dots, D$  and  $D$  is the dimension of candidate solution,  $r_1$  is randomly chosen from  $[1, N_F]$  and has to be different from  $l$ , i.e.,  $r_1 \neq l$ .  $\varphi_{ld}$  is a random number in the range  $[-1, 1]$ .

Once  $\theta_{EB}(l)$  is obtained, it is evaluated and compared with  $\theta(l)$ . If the fitness of  $\theta_{EB}(l)$  is better than that of  $\theta(l)$ ,  $\theta_{EB}(l)$  replaces  $\theta(l)$ ; otherwise if  $\theta(l)$  is retained, then  $k_{\text{count}}(l) = k_{\text{count}}(l) + 1$ .  $k_{\text{count}}(l)$  is a non-improvement number of the solution  $\theta(l)$ .

*Onlooker Bee Phase* After all worker bees complete their searches, they share in a dance their information related to the nectar amounts and the location of their sources with onlooker bees. An onlooker bee evaluates the nectar information given by each worker bee and chooses a food source site with a probability related to its nectar amount. This probability depends on the fitness values of the solutions in the population; it can be described as

$$p_l = \frac{F(l)}{\sum_{l=1}^{N_s} F(l)} \quad l = 1, 2, \dots, N_s, \tag{19}$$

where  $F(l)$  is the fitness function of  $l$ th candidate solution. Once the onlooker has selected her food source  $\theta(l)$ , she produces a modification on  $\theta_{OB}(l)$  using

$$\theta_{OB}(l, d) = \begin{cases} \theta(l, d) + \varphi_{ld}[\theta(l, d) - \theta(r_1, d)], & p_l < MR \\ \theta(l, d), & \text{otherwise} \end{cases}, \tag{20}$$

where  $MR$  is a control parameter of the ABC algorithm in the range  $[0, 1]$ , which controls the number of parameters to be modified. As in the phase of the worker bee, the onlooker bee produces a new position of the food source based on the fitness value. If the fitness of  $\theta_{OB}(l)$  is worse than that of  $\theta(l)$ ,  $\theta(l)$  is retained and  $k_{count}(l) = k_{count}(l)+1$ ; otherwise  $\theta_{OB}(l)$  replaces  $\theta(l)$ .

*Scout Bee Phase* If a food source  $\theta(l)$  cannot be further improved through a predetermined  $k_{limit}(l)$ , it is abandoned. The food source abandoned by the bees is replaced with a new food source discovered by the scouts. The scout produces a food source randomly from the range of  $[\theta_{min}(d)\theta_{max}(d)]$ .

Pseudocode of ABC algorithm is given in Table 3.

### 2.3 Sequential Blind Source Separation Method in Order Using Swarm Optimization Algorithm

The idea of this paper is the application of the swarm optimization algorithm in sequential BSS. In other words, the swarm optimization algorithms were used to find the global optima of (6). As explained above, the upper and lower bounds of a candidate solution are assumed known in advance. Because of the constraint  $\|\mathbf{w}\|^2 = 1$ , we can rewrite  $\mathbf{w}$  as

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{m-1} \\ w_m \end{bmatrix} = \begin{bmatrix} \cos \alpha_1 \\ \sin \alpha_1 \cos \alpha_1 \\ \vdots \\ \sin \alpha_1 \sin \alpha_2 \dots \cos \alpha_{m-1} \\ \sin \alpha_1 \sin \alpha_2 \dots \sin \alpha_{m-1} \end{bmatrix}, \tag{21}$$

where  $[\alpha_1, \alpha_2, \dots, \alpha_{m-1}] \in [0, 2\pi]^{(m-1)}$ , i.e., the upper and lower bounds of  $\alpha_i (i = 1, 2, \dots, m - 1)$  are 0 and  $2\pi$ , respectively.

The idea described in the above method aims at the extraction of one source having the highest absolute kurtosis value. However, it can also be used for recovering the remaining sources. After one source is extracted, we can remove it from the mixture. This can be achieved through a deflation procedure, in which the goal is to eliminate the contribution of the extracted  $y(t)$ . Mathematically, this procedure is given by

$$\mathbf{x}(t) \leftarrow \mathbf{x}(t) - \mathbf{h}y(t), \tag{22}$$

where  $\mathbf{h} = E[\mathbf{x}(t)y(t)]/E[y^2(t)]$  is a vector that minimizes  $E\{\|\mathbf{x}(t) - \mathbf{h}y(t)\|^2\}$ .



**Table 3** Artificial bee colony algorithm

---

**Initialize:** set maximum number of iterations  $k_{\max}$ ,  $k_{\text{limits}}$ ,  $N_F$ ,  $D$ ; Generate initial  $\theta(l, d)$  randomly from  $[\theta_{\min}(d) \theta_{\max}(d)]$ ,  $l=1, 2, \dots, N_F$ ,  $d=1, 2, \dots, D$ .

**while** the fit function criterion is not satisfied **and**  $k \leq k_{\max}$

**Employed bee phase**

for  $l=1$  to  $N_S$

$\theta_{\text{EB}}(l) = \theta(l)$ ;

Randomly select the dimension of the solution,  $d$ ;

Update  $\theta_{\text{EB}}(l, d)$  by (18);

Calculate fitness value  $F_{\text{EB}}(l)$  of  $\theta_{\text{EB}}(l)$ ;

if  $F_{\text{EB}}(l)$  is better than  $F(l)$ ,  $\theta(l) = \theta_{\text{EB}}(l)$ ,

$k_{\text{count}}(l) = 0$ ;

else  $k_{\text{count}}(l) = k_{\text{count}}(l) + 1$ ;

end

**Onlooker bee phase**

for  $l=1$  to  $N_S$

Update  $\theta_{\text{OB}}(l)$  by (19);

Calculate fitness value  $F_{\text{OB}}(l)$  of  $\theta_{\text{OB}}(l)$ ;

if  $F_{\text{OB}}(l)$  is better than  $F(l)$ ,  $\theta(l) = \theta_{\text{OB}}(l)$ ,

$k_{\text{count}}(l) = 0$ ;

else  $k_{\text{count}}(l) = k_{\text{count}}(l) + 1$ ;

end

**Scout bee phase**

for  $l=1$  to  $N_S$

if  $k_{\text{count}}(l) \geq k_{\text{limit}}$

Generate new  $\theta_{\text{SB}}(l)$  randomly;

Calculate fitness value  $F_{\text{SB}}(l)$  of  $\theta_{\text{SB}}(l)$ ;

if  $F_{\text{SB}}(l)$  is better than  $F(l)$ ,  $\theta(l) = \theta_{\text{SB}}(l)$ ,

$k_{\text{count}}(l) = 0$ ;

end

$k = k + 1$ ;

**end**

---

As the outcome of the first deflation is a BSS problem with  $n$  mixtures and  $n - 1$  sources, we reduce the dimension of  $n$  via principle component analysis (PCA) [28] to obtain a  $n - 1$  model. Finally, the remaining sources can be extracted by repeating the same steps.

For clarity, the pseudocode of the proposed method is presented in Table 4.

### 3 Simulation Analysis

In this section, a detailed simulation is provided to investigate the performance of the proposed method. In all experiments, the elements of the mixing matrix  $A$  in (1) are

**Table 4** Procedural steps for sequential blind source separation used in the swarm optimization algorithm

<ol style="list-style-type: none"> <li>1. <b>Initialize</b> <math>n'</math>, <math>n'=n</math>; <b>if</b> <math>m &gt; n'</math>, reduce observed signal to <math>n'</math> through PCA in [29];</li> <li>2. Calculate <math>\mathbf{B}</math> by computing the Schur decomposition of <math>\mathbf{C}=\mathbf{x}(t)\mathbf{x}(t)^T</math>, then sphere the observations, <math>\mathbf{z}(t)=\mathbf{B}\mathbf{x}(t)</math>;</li> <li>3. Set <math>\alpha</math> in (21) as <math>\theta</math>, then obtain the optimal extracted vector <math>w</math> through maximizing (6) using the PSO, DE or ABC;</li> <li>4. Estimate <math>y(t)</math> using (3);</li> <li>5. Deflate using (22);</li> <li>6. <math>n' \leftarrow n'-1</math>;</li> <li>7. <b>If</b> <math>m &gt; n'</math>, reduce the observed signal to <math>n'</math> through PCA in [28];</li> <li>8. <b>If</b> <math>n' &gt; 1</math> <b>go to step 3 end</b></li> </ol>
---

randomly generated in accordance with a uniform distribution in  $[-1, 1]$ . The ABC parameters are used for the simulation as specified in (23), whereas the DE and PSO parameters are used for the simulation as given in (24) and (25), respectively. The simulation experiments were performed using the program MALAB ver.R2010b.

$$\begin{cases} k_{\max} = 200 \\ k_{\text{limit}} = 5 \\ N_F = 20 \end{cases} \quad (23)$$

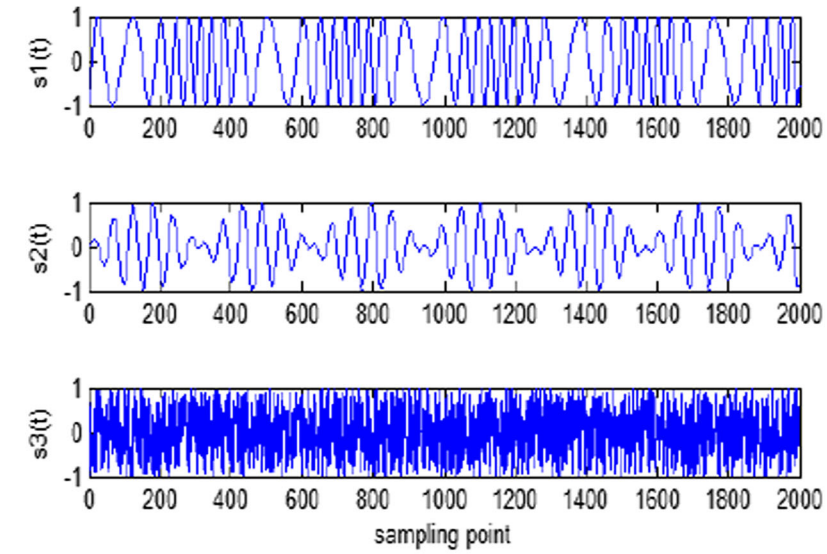
$$\begin{cases} k_{\max} = 200 \\ N_P = 20 \\ \omega_{\max} = 0.9 \\ \omega_{\min} = 0.4 \\ c_1 = c_2 = 0.2 \end{cases} \quad (24)$$

$$\begin{cases} k_{\max} = 200 \\ C_R = 0.5 \\ N_P = 20 \end{cases} \quad (25)$$

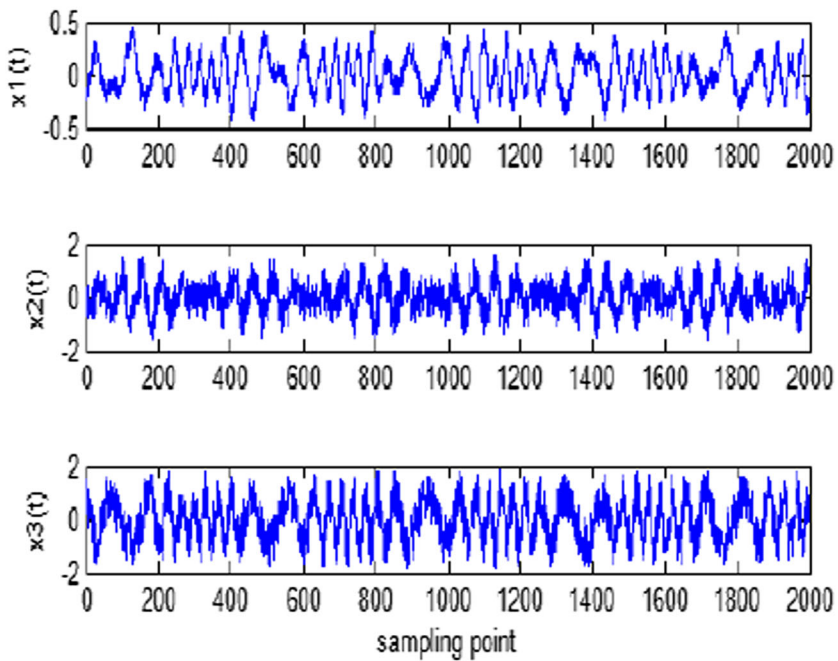
*Experiment 1* In the first experiment, we consider the separation of sub-Gaussian signals. The source signals are shown in Fig. 1a; Fig. 1b shows the observation signal given by (1). Figure 2 shows the convergence curve of the different swarm optimization algorithms for first extraction of the signal. Table 5 shows the kurtosis of the estimated signals for a sub-Gaussian. The performance of the BSS algorithm can be evaluated using (26) and (27). A comparison of results obtained from different BSS methods for the sub-Gaussian signals is given in Table 6. Note that the simulation results were presented as average values of 30 independent runs. The method<sup>1</sup> used in [11] is that the cost function  $J(y) = \frac{1}{4}y(t)^4$  was maximized by the ABC algorithm of Sect. 2.

$$\text{SIR}_i = 10 \log \frac{\|\mathbf{s}_i\|^2}{\|\mathbf{s}_i - \mathbf{y}_i\|^2} \quad i = 1, 2, \dots, n \quad (26)$$

$$\text{SNR} = \frac{1}{n} \sum_i 10 \log \frac{\|\mathbf{s}_i\|^2}{\|\mathbf{s}_i - \mathbf{y}_i\|^2} \quad (27)$$

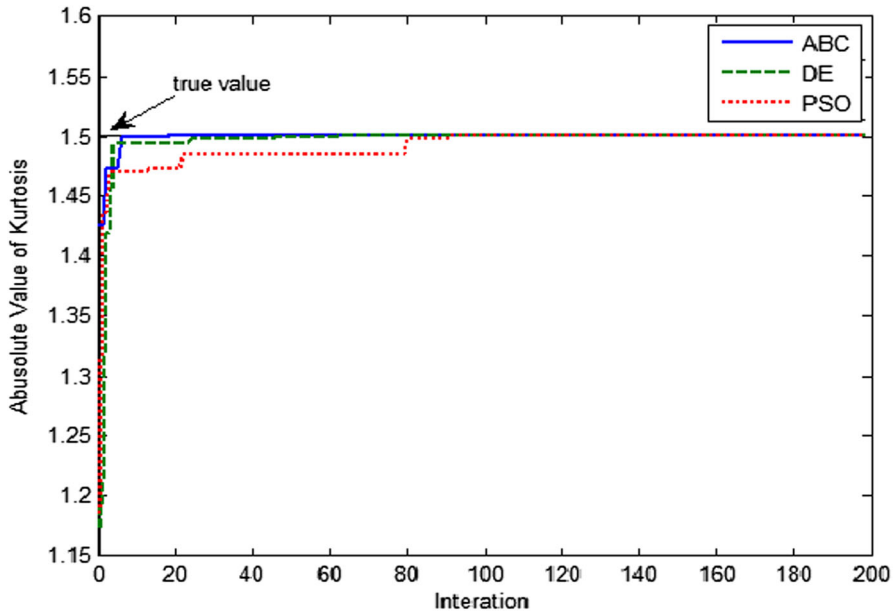


(a)



(b)

**Fig. 1** Signals of Experiment 1. **a** Source signals. **b** Observation signals



**Fig. 2** Convergence curves for different swarm optimization algorithms for first extracted signals with sub-Gaussian nature

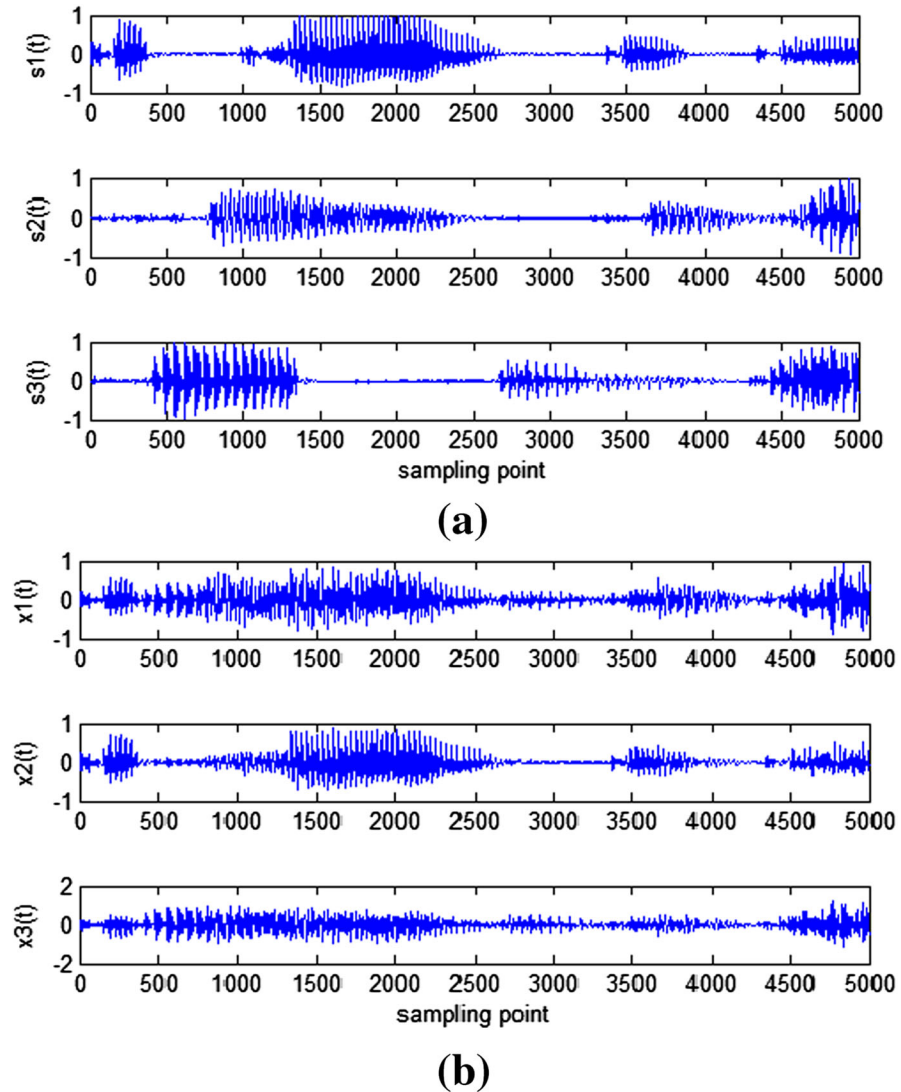
**Table 5** Kurtosis of separated signal of sub-Gaussian nature

Separated signal	Source signal	True value	Estimated value		
			PSO-BSS	DE-BSS	ABC-BSS
$y_1$	$s_1$	-1.5003	-1.5004	-1.5004	-1.5004
$y_2$	$s_3$	-1.1875	-1.1869	-1.1875	-1.1868
$y_3$	$s_2$	-0.7338	-0.7297	-0.7297	-0.7297

In Table 5, the method using the ABC algorithm is denoted ABC-BSS and that using DE and PSO is denoted DE-BSS and PSO-BSS, respectively

**Table 6** Comparison of results from different BSS methods for a sub-Gaussian signal

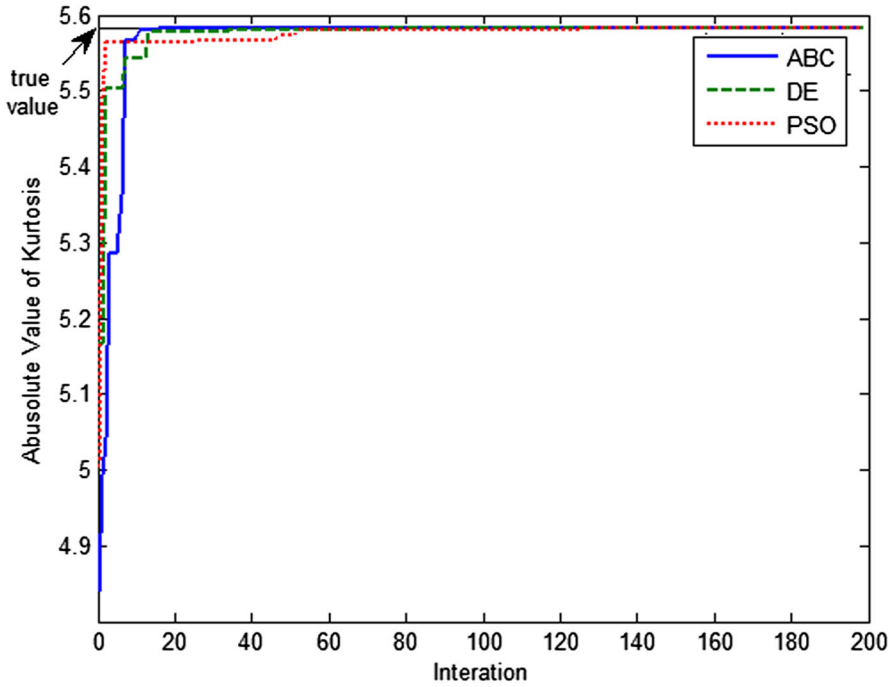
Source signals	BSS methods			
	PSO-BSS	DE-BSS	ABC-BSS	The method <sup>1</sup> in [11]
$SIR_i$ /dB				
$s_1$	43.3334	43.3045	43.3335	43.2375
$s_2$	29.9809	29.9836	29.9851	29.5672
$s_3$	30.3215	30.3310	30.3259	30.1221
SNR/dB	34.5453	34.5397	34.5481	34.3089



**Fig. 3** Signals of Experiment 2. **a** Source signals. **b** Observation signals

where the vector  $\mathbf{y}_i$  is an estimation of  $\mathbf{s}_i$ . Clearly,  $\text{SIR}_i \rightarrow \infty$  and  $\text{SNR} \rightarrow \infty$  if perfect separation is achieved.

*Experiment 2* In this experiment, the three speech signals in Fig. 3a are used. They are sup-Gaussian distributions and are available in [5]. Figure 3b shows the observation signal given by (1). Figure 4 shows the convergence curve of different swarm optimization algorithms for the first extracted signal. Table 7 gives the kurtosis of the estimated signals of the sub-Gaussian. A comparison of results from the different BSS methods for sup-Gaussian signals is shown in Table 8. The method<sup>2</sup> used in [11] is that the cost function  $J(y) = \frac{1}{a_1} \exp[-a_1 y(t)^2/2] y(t)$  was maximized using the ABC algorithm given in Sect. 2,  $a_1 \approx 1$ .



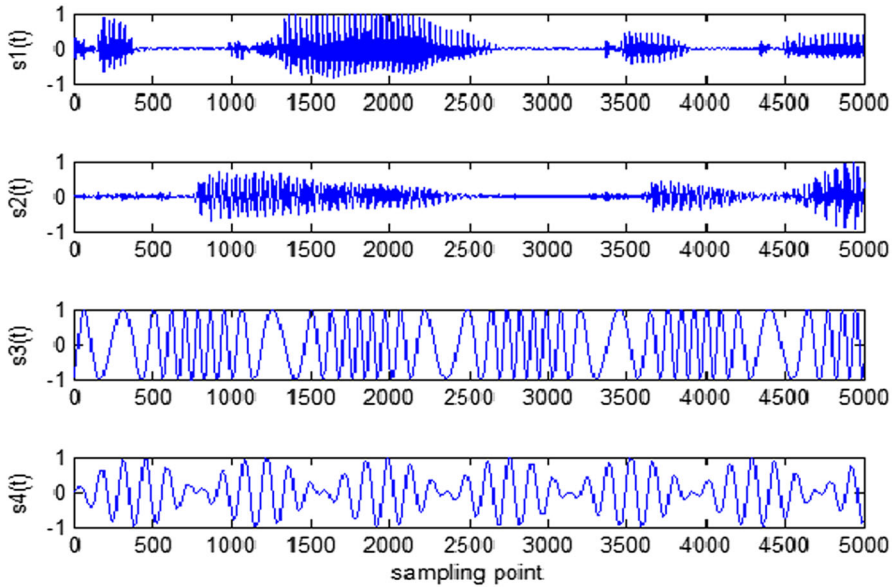
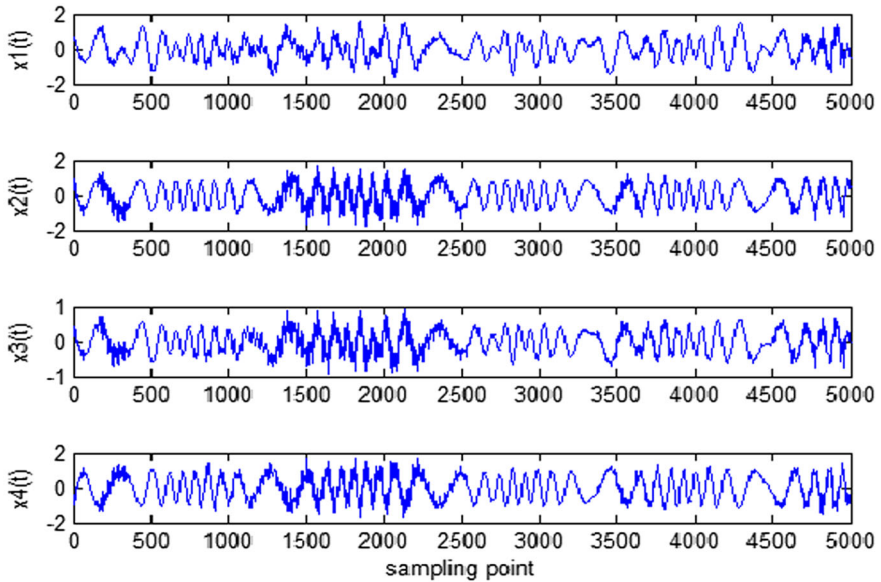
**Fig. 4** Convergent curve of different swarm optimization algorithms for the first extracted sup-Gaussian signals

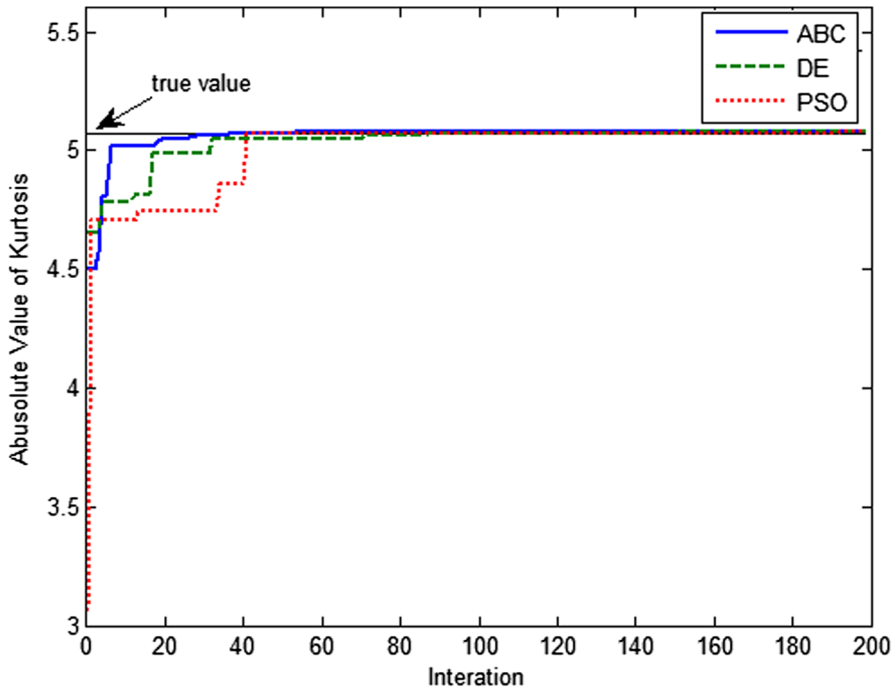
**Table 7** Kurtosis of the separated sup-Gaussian signals

Separated signal	Source signal	True value	Estimated value		
			PSO-BSS	DE-BSS	ABC-BSS
y1	s3	5.5813	5.5816	5.5816	5.5816
y2	s1	5.0720	5.0737	5.0737	5.0737
y3	s2	3.9000	3.8907	3.8907	3.8907

**Table 8** Comparison of results from different BSS methods for sup-Gaussian

Source signals	BSS methods			
	PSO-BSS	DE-BSS	ABC-BSS	The method <sup>2</sup> in [11]
SIRi/dB				
s1	38.3914	38.3921	38.3913	38.3332
s2	25.3188	25.3185	25.3188	24.7163
s3	47.3185	47.3157	47.3185	38.6097
SNR/dB	37.0095	37.0088	37.0095	33.8864

**(a)****(b)****Fig. 5** Signals of Experiment 3. **a** Source signals. **b** Observation signals



**Fig. 6** Convergent curve of different swarm optimization algorithms for first extracted signals with sub- and sup-Gaussian nature

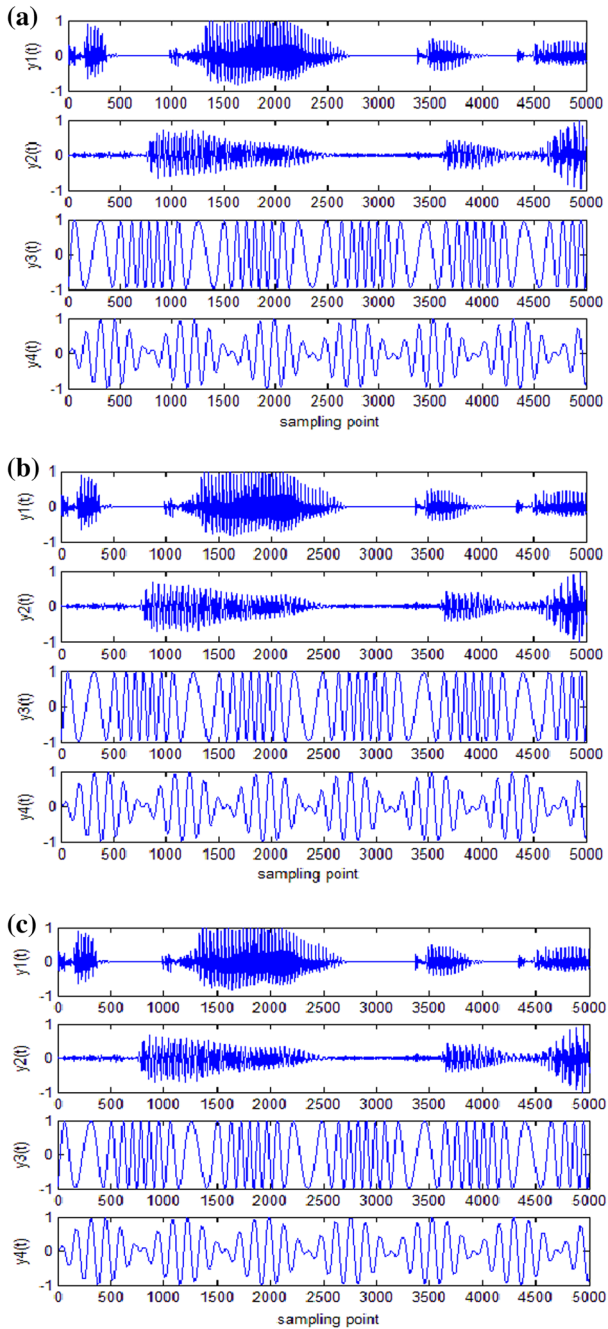
**Table 9** Kurtosis of separated signals of sub- and sup-Gaussian nature

Separated signal	Source signal	True value	Estimated value		
			PSO-BSS	DE-BSS	ABC-BSS
$y_1$	$s_1$	5.0720	5.0761	5.0762	5.0762
$y_2$	$s_2$	3.9000	3.8478	3.9013	3.9013
$y_3$	$s_3$	-1.5004	-1.4995	-1.4995	-1.4996
$y_4$	$s_4$	-0.7331	-0.7336	-0.7336	-0.7335

**Table 10** Comparison of results from different BSS methods for sub- and sup-Gaussian signals

Source signals	BSS methods					
	PSO-BSS	DE-BSS	ABC-BSS	Method <sup>3</sup> in [11]	Method <sup>1</sup> in [26]	Method <sup>1</sup> in [24]
$SIR_i$ /dB						
$s_1$	33.2932	32.7590	33.9471	35.4620	30.0469	25.8577
$s_2$	39.3677	39.4511	41.1020	15.7000	16.3074	24.4787
$s_3$	27.9899	27.1727	28.2107	26.8559	26.8471	2.0858
$s_4$	34.2056	33.7043	34.0221	13.0322	13.7630	10.0781
SNR/dB	33.7141	33.2718	34.3208	22.7625	21.7411	15.6251





**Fig. 7** Recovered signals in Experiment 3. **a** Recovered signals of ABC-BSS method. **b** Recovered signals of DE-BSS method. **c** Recovered signals of PSO-BSS method. **d** Recovered signals of the method<sup>3</sup> in [11]. **e** Recovered signals using the method<sup>1</sup> in [26]. **f** Recovered signals of the method<sup>1</sup> in [24]

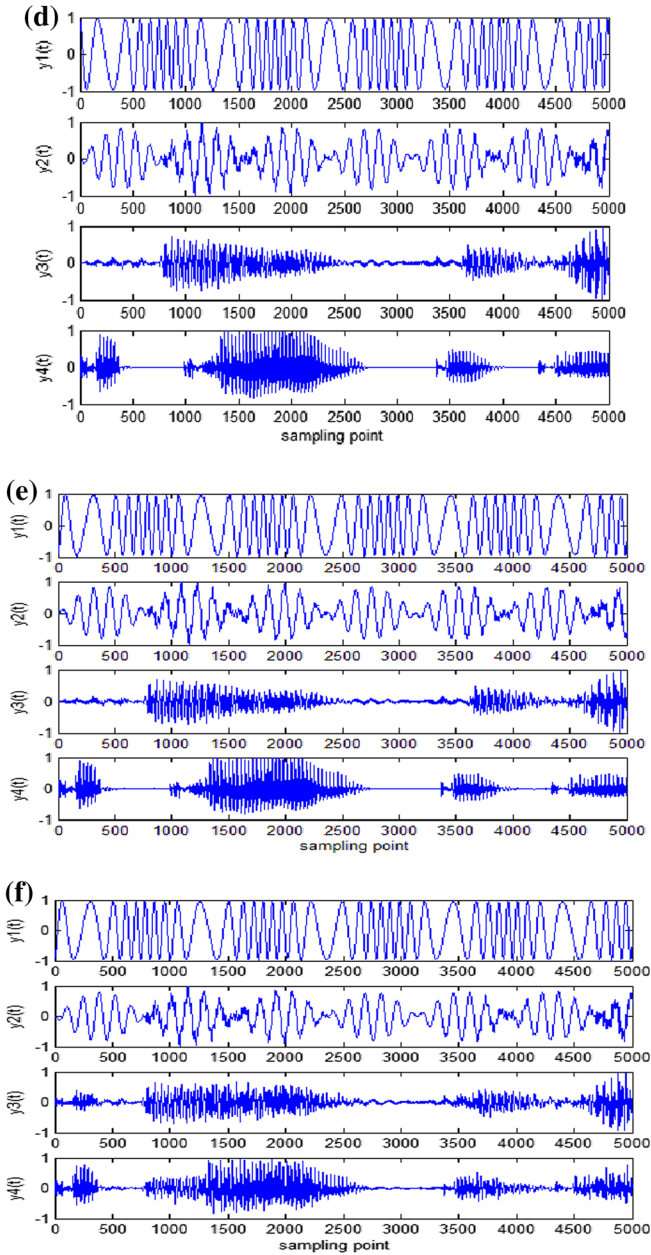
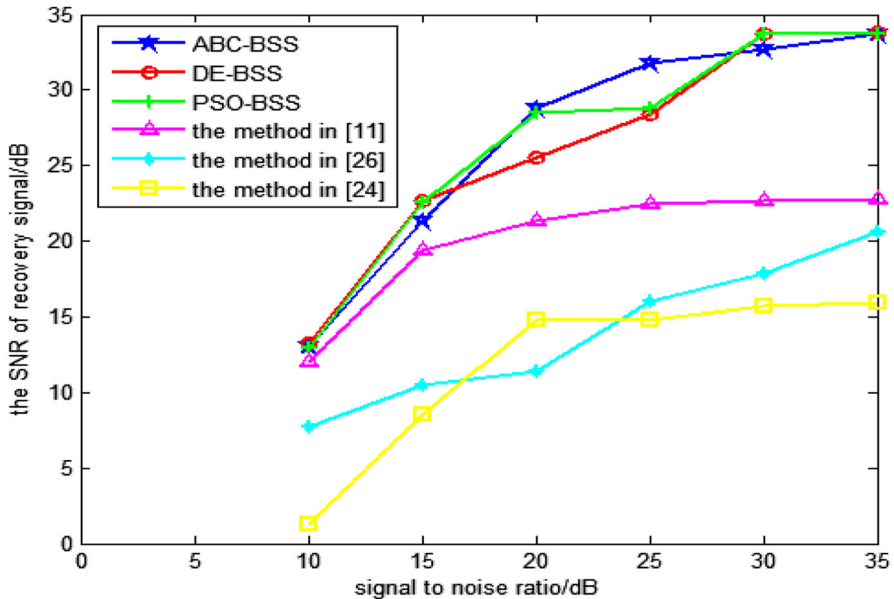
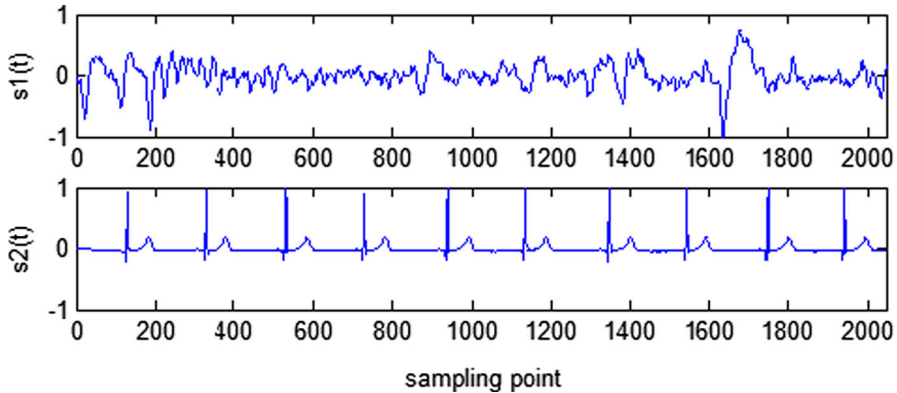


Fig. 7 continued

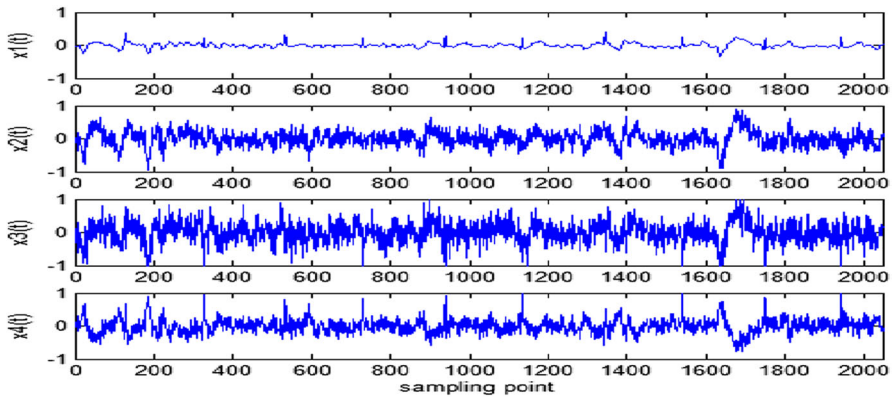


**Fig. 8** Comparison of results of different BSS methods



**Fig. 9** EEG signals

*Experiment 3* This experiment exemplifies the separation of four source signals given in Fig. 5a; the observation signals are shown in Fig. 5b. Both  $s_1(t)$  and  $s_2(t)$  are speech signals of sup-Gaussian nature, whereas  $s_3(t)$  and  $s_4(t)$  are the communication signals of sub-Gaussian nature. The convergent curve for different swarm optimization algorithms for the first extraction of the signal is shown in Fig. 6. The kurtosis of the estimated signals (Table 9) and a comparison of results from different BSS methods (Table 10) are given for sup- and sub-Gaussian signals. The method<sup>3</sup> used in [11] maximizes the cost function  $J(y) = \frac{1}{a_2} \log \cosh[a_2 y(t)]$  using the ABC algorithm in Sect. 2,  $1 \leq a_2 \leq 2$ . The method<sup>1</sup> in [26] maximizes the cost function in [26] using



**Fig. 10** Mixed EEG signals of the four channels

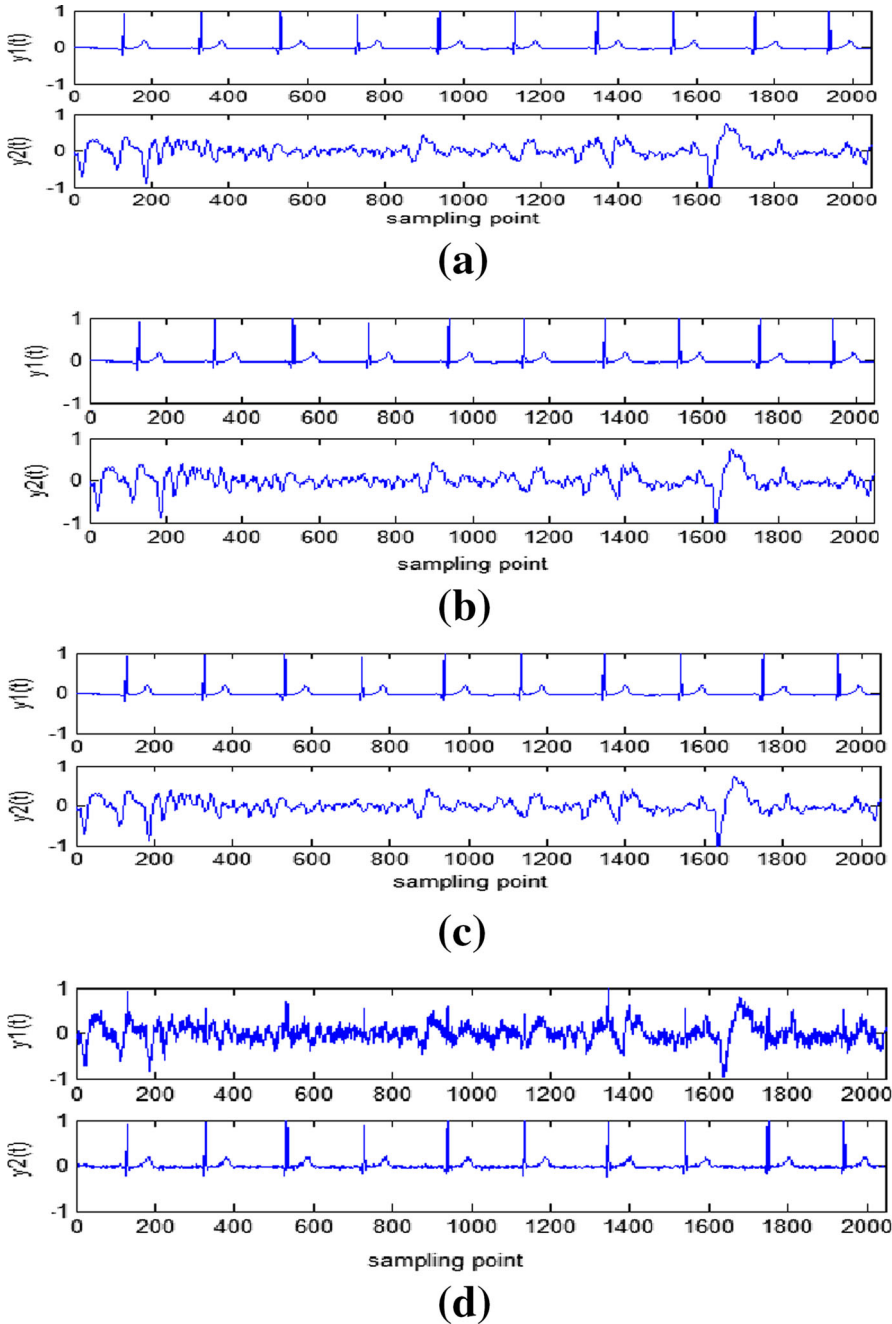
the DE algorithm in Sect. 2. The method<sup>1</sup> in [24] maximizes the cost function in [24] using the PSO algorithm in Sect. 2.

From Figs. 2, 4, and 6, we observe that the ABC algorithm requires less iterations to achieve the fitness value compared with DE and PSO. A comparison of results from Tables 6, 8, and 10 shows that the ABC-BSS can separate sources of any distribution and has superior performance compared with DE-BSS and PSO-BSS. From Fig. 7 and Tables 5, 7, and 9, the proposed method is able to extract source signals (of sub-Gaussian and/or sup-Gaussian nature) one by one with decreasing order of absolute kurtosis. However, the results in Fig. 7d–f clearly demonstrate that method<sup>3</sup> in [11], method<sup>1</sup> in [26], and method<sup>1</sup> in [24] cannot extract source signals in the order specified by their stochastic properties. Compared with the proposed methods, the method in [11] separated sources with lower accuracy, especially for the mixed case of sub- and sup-Gaussians.

*Experiment 4* To examine the noise impact of the separated performance of the proposed methods, in the last experiment, the four observation signals in Fig. 5b were added with different white Gaussian noise of varying signal-to-noise ratio (SNR) from 10 to 35 dB. Moreover, we verify the effectiveness of the proposed method by comparing it with the algorithms in [11, 24], and [26]. More details of the algorithm in [11] are available in [22]. The performance of the recovered signals is evaluated by (27). Figure 8 shows a comparison of results for five different BSS methods.

As observed in Fig. 8, the separated performance of ABC-BSS, DE-BSS, and PSO-BSS is quite similar. The comparison of results also shows that our method can separate sources of sub- and sup-Gaussian nature with higher accuracy compared with other conventional methods.

*Experiment 5* This class of BSS described in Sect. 2 could be essential in many applications, especially when the number of sources is large, and only some of them are interesting or useful signals buried in Gaussian noises. In this experiment, to further confirm the estimation performance of the proposed algorithms, we used a real-world example to compare the performance with those of the algorithm used in [24]. Figure 9 shows two electroencephalogram (EEG) signals that are available in



**Fig. 11** Recovered EEG signals of the different BSS methods. **a** Recovered EEG signals of the ABC-BSS method. **b** Recovered EEG signals of the DE-BSS method. **c** Recovered EEG signals of the DE-BSS method. **d** Recovered EEG signals of the method in [24]

**Table 11** Kurtosis of separated EEG signals

Separated signal	Source signal	True value	BSS methods		
			PSO-BSS	DE-BSS	ABC-BSS
$y_1$	$s_2$	39.1266	39.1372	39.1372	39.1272
$y_2$	$s_1$	3.5458	3.5405	3.5405	3.5405
$y_3$	$s_3$	-0.0356	-0.0474	-0.0474	-0.0377

**Table 12** Comparison of results for the different BSS methods applied to the EEG signal

Source signals	BSS methods			
	PSO-BSS	DE-BSS	ABC-BSS	Method in [24]
$SIR_i$ /dB				
$s_1$	28.0683	28.0683	28.0699	4.4413
$s_2$	38.7114	38.7119	38.7918	16.6768

[5], the kurtosis is positive and relatively large for the EEG signals (as also for clear heart, eye movement, and eye blinking artifacts); the kurtosis is zero or negative for noise. We mixed two EEG signals and one white Gaussian noise into four channels of an EEG. The mixed signals of the four channels are shown in Fig. 10. Figure 11 shows that we have employed different BSS methods to recover the EEG signals from the mixed signals; a comparison of the recovery performance is shown in Table 12. Table 11 shows the kurtosis of the estimated EEG signals. Note that  $m$  and  $n$  in this experiment are 4 and 3, respectively. Because the method in [26] uses an assumption that implies  $m = n$ , the method is not appropriate if  $m > n$ . Also, [11] makes the further assumption that the dimensions of  $\mathbf{x}(t)$  and  $\mathbf{s}(t)$  are equal.

As expected, Table 11 shows the proposed methods first recover those EEG signals of interest, whereas the method in [24] separated “useful” signals and “useless” noise simultaneously. Again, the superior performance of our method is confirmed by the results in Fig. 11 and Table 12. These features enable the proposed method to be of benefit in practice.

## 4 Conclusion

This paper introduced a class of sequential BSS methods that maximized the kurtosis cost function. They are able to extract the source signals of sub- and sup-Gaussian form one by one in a specific order determined according to their stochastic properties, specifically, in decreasing order of the absolute kurtosis. Simulation results demonstrated the excellent performance of the proposed method. Compared with previous methods, the methods introduced have the following merits: recovery of the original signals from a linear mixture of signals with higher accuracy and extraction of various sources in decreasing order according to the absolute value of the kurtosis. In addition, the proposed methods are suitable for  $m \geq n$ . These features make the methods suit-

able in many applications, especially when the number of sources is large and when only some signals are of interest or when useful signals are buried in white Gaussian noise.

Recently, complex-valued signals have arisen in many applications as diverse as wireless communications, radar, and functional magnetic resonance imaging. When processing has to be done in a transform domain such as Fourier space or complex wavelet domains, the data are again complex-valued. To perform BSS of such complex-valued signals, there are a number of options. For complex-valued signals, the Gaussianity implies zero kurtosis, sup-Gaussianity implies positive kurtosis, and sub-Gaussianity implies negative kurtosis [30]. The resolution of sequential BSS of various complex sources is one of the challenging problems of BSS. Future work may include developing sequential complex BSS algorithms that achieve the objective of extracting complex-valued signals one by one with specified order.

**Acknowledgments** This work was supported by the National Natural Science Foundation of China under Grant No. 51309116, the Foundation of Fujian Education Committee for Distinguished Young Scholars under Grant No. JA14169, the Scientific Research Foundation of Jimei University under Grant Nos. ZQ2013001 and ZC2013012, and the Scientific Research Foundation of Artificial Intelligence Key Laboratory of Sichuan Province under Grant No. 2014RYJ03.

## References

1. M. Babita, G. Panda, Development of efficient identification scheme for nonlinear dynamic systems using swarm intelligence techniques. *Expert Syst. Appl.* **37**(1), 556–566 (2010)
2. B. Bloemendal, L.J. Vande, P. Sommen, Blind source extraction for a combined fixed and wireless sensor network. In: *Proceedings of the 20th European Signal Processing Conference* (2012), pp. 1264–1268
3. L. Chen, L.Y. Zhang, Y.J. Guo, Blind signal separation algorithm based on temporal predictability and differential search algorithm. *J. Commun.* **35**(6), 117–125 (2013)
4. L.N. Castrode, *Fundamentals Of Natural Computing: Basic Concepts, Algorithms And Applications* (Chapman & Hall, CRC, Boca Raton, 2006)
5. Cichocki, A., Amari, S., Siwek, K., ICALAB toolboxes. <http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/benchmarks> (2007). Accessed 23 March 2007
6. S.A. Cruces-alvarez, A. Cichocki, S. Amari, From blind signal extraction to blind instantaneous signal separation: criteria, algorithm and stability. *IEEE Trans. Neural Netw.* **15**(4), 859–873 (2004)
7. P. Gita, R.N. Ganesh, T.N. Hung, Using blind source separation on accelerometry data to analyze and distinguish the toe walking gait from normal gait in ITW children. *Biomed. Signal Process. Control* **13**(1), 41–49 (2014)
8. W.Y. Gong, Z.H. Cai, Parameter extraction of solar cell models using repaired adaptive differential evolution. *Sol. Energy.* **94**(8), 209–220 (2013)
9. Y.N. Guo, S.H. Huang, Y.T. Li, Edge effect elimination in single-mixture blind source separation. *Circuits Syst. Signal Process.* **12**(5), 2317–2334 (2013)
10. E. Hoffmann, K. Korothea, K. Bert-uwe, Using information theoretic distance measures for solving the permutation problem of blind source separation of speech signals. *EURASIP J. Audio Speech Music Process.* **2012**(1), 1–14 (2012)
11. A. Hyvarinen, Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Netw.* **10**(3), 626–634 (1999)
12. D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm. *Appl. Math. Comput.* **214**(1), 108–132 (2009)
13. D. Karaboga, B. Akay, A modified artificial bee colony algorithm for constrained optimization problems. *Appl. Soft Comput.* **11**(3), 3021–3031 (2011)

14. J. Kennedy, R.C. Eberhart, Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks* (1995), pp. 1942–1948
15. I. Meganem, Y. Deville, S. Hosseini, Linear-quadratic blind source separation using NMF to unmix urban hyperspectral images. *IEEE Trans. Signal Process.* **62**(7), 1822–1833 (2014)
16. G.R. Naik, An overview of independent component analysis and its applications. *Informatica* **35**(1), 63–81 (2011)
17. G.R. Naik, Measure of quality of source separation for sub- and super-Gaussian audio mixtures. *Informatica.* **23**(4), 581–599 (2012)
18. G.R. Naik, W.W. Wang, Audio analysis of statistically instantaneous signals with mixed Gaussian probability distributions. *Int. J. Electron.* **99**(10), 1333–1350 (2012)
19. O. Naoya, I. Atsushi, Independent component analysis of optical flow for robot navigation. *Neuro-computing* **71**(10–12), 2140–2163 (2008)
20. J. Nikunen, T. Virtanen, Direction of arrival based spatial covariance model for blind sound source separation. *IEEE Trans. Audio Speech Lang. Process.* **22**(3), 727–739 (2014)
21. J.A. Palmer, M. Scott, Contrast functions for independent subspace analysis. *LVA/ICA* **2012**, 115–122 (2012)
22. C. Pierre, J. Christian, *Handbook of Blind Source Separation: Independent Component Analysis and Applications* (Academic Press, Burlington, 2010)
23. S. Rahmmayan, H.R. Tizhoosh, M.A. Magdy, Opposition-based differential evolution. *IEEE Trans. Evolut. Comput.* **12**(1), 64–79 (2008)
24. Z.W. Shi, H.J. Zhang, Z.G. Jiang, Hybrid linear and nonlinear complexity pursuit for blind source separation. *J. Comput. Appl. Math.* **236**(14), 3434–3444 (2012)
25. R. Storn, K. Price, *Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces* (University of California, Berkeley, 1996)
26. M. Uttachai, A. Patrick, A class of Frobenius norm-based algorithms using penalty term and natural gradient for blind signal separation. *IEEE Trans. Signal Process.* **16**(6), 1181–1193 (2008)
27. R.J. Wang, *Research on Algorithms of Underdetermined Blind Source Separation and Adaptive Complex Blind Source Separation* (Sun Yat-sen University, Guangzhou, 2012)
28. R.J. Wang, H.F. Zhou, Application of SVM in fault diagnosis of power electronics rectifiers. *WCICA* **2008**, 1256–1260 (2008)
29. R.J. Wang, Y. Zhu, Nonlinear dynamic system identification based on FLANN. *J. Jimei Univ. (Natural Science)* **16**(2), 128–134 (2011)
30. W.V.D. Wijesinghe, G.M.R.I. Godaliyadda, M.P.B. Ekanayake, A generalized ICA algorithm for extraction of super and sub Gaussian source signals from a complex valued mixture. *ICIIS* **2013**, 144–149 (2013)
31. C.Z. Zhang, J.P. Zhang, X.D. Sun, Blind source separation based on adaptive particle swarm optimization. *Syst. Eng. Electron.* **31**(6), 1275–1278 (2009)
32. X.D. Zhang, *Modern Signal Processing* (Tsinghua University Press, Beijing, 2002)