International Journal of Innovative Computing, Information and Control
Volume **11**, Number **2**, April **2015**

ICIC International ©2015 ISSN 1349-4198
pp. 447–462

UNDERDETERMINED BLIND SOURCE SEPARATION BASED ON DISTRIBUTED COMPRESSED SENSING

Yong Xu[1,2], Yujie Zhang[3] and Hongwei Li[3,*]

[1]Institute of Geophysics and Geomatics
[3]School of Mathematics and Physics
China University of Geosciences
No. 388, Lumo Road, Wuhan 430074, P. R. China
xuyong654321@126.com; *Corresponding author: hwli@cug.edu.cn

[2]Institute of Statistics
Hubei University of Economics
No. 8, Yangqiaohu Ave., Jiangxia Dist., Wuhan 430205, P. R. China

Received March 2014; revised July 2014

Abstract. *Underdetermined Blind Source Separation (UBSS) is an important issue which is frequently solved by a two-stage approach: estimating mixing matrix and reconstructing source signals. In this paper, a UBSS model with introduced feature is presented. The distributed compressed sensing (DCS)-based algorithm is proposed to reconstruct the sources. In the reconstruction stage, the observations are firstly segmented into a number of blocks. Then a reconstructing algorithm based on DCS is employed at the stacked blocks. Finally, the source matrix is obtained from these reconstructed blocks. Compared with other methods, simulations demonstrate lower complexity of the proposed method even if the number of sources is larger than 10.*

Keywords: Underdetermined blind source separation, Sparsity, Distributed compressed sensing, Block matrix, Complexity

1. **Introduction.** The issue of underdetermined blind source separation (UBSS), where the number of the sources is larger than that of the mixtures, has been extensively researched in recent years. The goal of UBSS is to estimate the unknown sources from the mixture without prior information of the mixing system. Taking into account a noise-free instantaneous model, the issue could be described as [1]:

$$\mathbf{Z} = \mathbf{AS}, \tag{1}$$

where $\mathbf{A} \in \mathbf{R}^{c\times l}$ is the unknown mixing matrix assumed to be full row rank with $c < l$, $\mathbf{Z} \in \mathbf{R}^{c\times n}$ is the observed data matrix whose row $\mathbf{z}_i$ is the i-th observed signals with all times $t = 1, 2, \dots, n$. $\mathbf{S} \in \mathbf{R}^{l\times n}$ is the unknown source matrix with l source $\mathbf{s}_i$ $(i = 1, 2, \dots, l)$, where $\mathbf{s}_i$ is a row-vector of size n.

In case of UBSS, estimating the mixing matrix is not sufficient for recovering the sources since the mixing matrix is not invertible. Sparsity of the sources is required as the prior information to solve the underdetermined problem. The sparsity of sources means that, at a given data point t, few sources are prominently larger, while the remaining ones are very close to zero. Particularly, it is called K-sparse if no more than K sources are nonzero at each data point. Many signals are not sparse in time domain, but they are sparse in certain transformed domains such as the frequency domain by Fourier transform [1]. In recent years, a lot of work have been studied about UBSS. The method most broadly used in UBSS includes two stages which are estimating mixing matrix and recovering source signals. For the mixing matrix estimation stage, some clustering methods such as time-frequency transform [1], k-means clustering method [2], have been proposed. For the sources recovering stage, there are l_1-norm solution method [2], shortest path decomposition methods [3,4], statistically sparse decomposition principle [5,6], etc. Another method for UBSS is based on sparse representation, supposing that the sources are sparse or can be decomposed into the combination of sparse components [4-7]. Employing such a representation, a novel algorithm was proposed based on compressed sensing (CS) [8], in which the UBSS model is reformulated into a signal recovery model. An approach based on multichannel CS theory was proposed for blind speech separation [9], in which the mixed signals were sampled by CS theory approach. Then the speech sources were separated and recovered without reconstructing the mixed signals.

Unfortunately, in case of UBSS, the number of sources l is required to be no larger than 10 [2-5,9], otherwise, these methods are very computationally demanding. On the other hand, as all the source signals of full length are stacked into a single vector for CS, which leads to a significant dimension of the measurement, the process of source estimation in [8] is also computationally demanding if $l > 10$ (see Section 4).

The paper considers a realistic case that was not mentioned in traditional UBSS papers, i.e., not all of the sources are sparse in some environment. For instance, in a party some people speak intermittently, while background music is played continuously, i.e., the source from the music is not sparse. The feature is added to UBSS model, that is, some $\mathbf{s}_i$ are assumed with all data points being nonzero or significant values. We call these sources "major sources".

Besides that, due to the increasing needs in applications, e.g., dozens of radar signals are needed to be separated for extracting some vital information in a region, it is necessary to develop efficient and reliable algorithms to solve large size UBSS (e.g., the number of sources is larger than 10). In order to lower the computational complexity in source estimation of large size UBSS, multi-signal-based CS algorithms are considered in this paper. Enlightened by [8], the relationship between DCS and UBSS is explored in this paper, including their models and algorithms. As a result, a DCS-based algorithm is proposed in the sources estimation stage of UBSS.

The remainder of the paper is organized as follows. Section 2 presents the relationship between DCS and UBSS, which results in the DCS-based method for UBSS. Section 3 demonstrates the comparison between our method and other two UBSS methods, in terms of computational complexity. In Section 4, several numerical experiments are conducted to illustrate the performance of separation, complexity and application potential of this method.

2. Relationship between DCS and UBSS.

2.1. Compressed sensing.

CS has received great interests in recent years [11]. Assume that $\mathbf{x}$ is an N-dimension signal, it is called K-sparse (or compressible) if $\mathbf{x}$ can be closely approximated by K ($\ll N$) coefficients under linear transformations. According to the CS theory, signal x can be obtained through the linear measurement system as follows:

$$\mathbf{y} = \phi\mathbf{x}, \tag{2}$$

where $\mathbf{y} \in \mathbf{R}^{M\times 1}$ is the measurement vector with M ($\ll N$) data points, ϕ represents an $M \times N$ measurement matrix. The goal of CS is to reconstruct signal via measurement. The core of reconstructing $\mathbf{x}$ is to find its support-set, i.e., the indices of non-zero entries of $\mathbf{x}$. The CS framework is interesting as it indicates that $\mathbf{x}$ can be precisely recovered from only $M = O\left(K\text{log}N\right)$ measurements, implying the potential of prominent cost reduction in digital data acquisition [12].

2.2. **Distributed compressed sensing (DCS).** Here we consider the following DCS model [10], that is:

$$\mathbf{y}_j = \mathbf{\Phi}\mathbf{x}_j \quad (j = 1, 2, \ldots, T), \tag{3}$$

where column vectors $\mathbf{x}_j \in \mathbf{R}^{N\times 1}$ and $\mathbf{y}_j \in \mathbf{R}^{M\times 1}$ are signal and corresponding measurement respectively.

On the other hand, Equation (3) can be represented as:

$$\mathbf{Y}_{M\times T} = \mathbf{\Phi}_{M\times N}\mathbf{X}_{N\times T}, \tag{4}$$

where $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T)$ and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$. Note that Equation (4) can be interpreted as compressed sensing based on multi-signal ensembles.

It is assumed that there exists a known sparse basis ψ so that each $\mathbf{x}_j$ can be sparsely represented by K ($K \ll N$) non-zero components. In CS, ψ and $\mathbf{\Phi}$ should be as orthogonal as possible so that each measurement contains more different information of original signals as far as possible [6]. Especially, $\mathbf{x}_j$ is K-sparse signal in time domain if $\psi = \mathbf{I}_N$. Then each $\mathbf{x}_j$ can be divided into common part and innovation part, i.e.,

$$\mathbf{x}_j = \mathbf{c}_j + \mathbf{z}_j = \psi(\theta_j + \sigma_j) \quad (j = 1, 2, \ldots, T), \tag{5}$$

where $\mathbf{c}_j$ and $\mathbf{z}_j$ represent the common part and innovation part of j-th signal respectively. The common part is common only in the support set and non-zero coefficients of the innovation parts are different among the sensors. If the signal is sparse in certain transform domain instead of in time domain, θ_j and σ_j represent the common part coefficients and innovation part coefficients respectively [10].

If $\mathbf{x}_j$ is K-sparse, sparsity K can be represented as:

$$K = K_c + K_j, \tag{6}$$

where K_c and K_j represent cardinality of the sparse common part and the sparse innovation part, respectively. For all $\mathbf{x}_j$ ($j = 1, 2, \ldots, T$), they share same K_c while having different K_j among them. Then total sparsity of $\mathbf{X}$ is written as:

$$K_{total} = K_c + \sum_{j=1}^{T} K_j. \tag{7}$$

For example, there is a multi-signal ensemble

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 4 & 3 \\ 0 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix}, \tag{8}$$

where

$$\mathbf{x}_1 = \mathbf{c}_1 + \mathbf{z}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{9}$$

$$\mathbf{x}_2 = \mathbf{c}_2 + \mathbf{z}_2 = \begin{pmatrix} 0 \\ 4 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 3 \\ 0 \end{pmatrix} \tag{10}$$

and

$$\mathbf{x}_3 = \mathbf{c}_3 + \mathbf{z}_3 = \begin{pmatrix} 0 \\ 3 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 5 \end{pmatrix}. \tag{11}$$

Comparing with these three sparse vectors, $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$ share the same nonzero index, while $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z_3}$ are sparse with different nonzero indexes. That is $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$ are the common parts, while $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z}_3$ are the innovation parts. Correspondingly, $K_c = 1$, $K_1 = K_2 = K_3 = 1$, $K_{total} = 1 + 3 = 4$.

2.3. **Relationship between UBSS and DCS.** In this paper, the mixing matrix $\mathbf{A}$ is assumed to have been estimated correctly in the mixing matrix estimation stage, i.e., $\mathbf{A}$ is known. For convenience and without loss of generality, source signals are assumed to be sparse in time domain. We are mainly concerned with the reconstruction of sources in UBSS.

Firstly, the mixture $\mathbf{Z}$ in Equation (1) is divided into k blocks $\mathbf{Z}_i$ $(i = 1, 2, \ldots, k)$, with each block containing $\frac{n}{k}$ columns as follows:

$$\mathbf{Z} = \left(\ \mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_k \ \right). \tag{12}$$

Then all blocks are stacked one by one, i.e., $\mathbf{Z}_{i+1}$ is stacked below $\mathbf{Z}_i$ $(i = 1, 2, \ldots, k-1)$. Thus, all of the blocks form a $ck \times \frac{n}{k}$ matrix, which is denoted by $\mathbf{Y}$ as below:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_k \end{pmatrix}. \tag{13}$$

Similarly, source matrix $\mathbf{S}$ is separated into k blocks correspondingly, with each block containing $\frac{n}{k}$ columns as follows:

$$\mathbf{S} = \left(\ \mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_k \ \right). \tag{14}$$

All $\mathbf{S}_i$ are stacked to be a $lk \times \frac{n}{k}$ matrix, which is denoted by $\mathbf{X}$:

$$\mathbf{X} = \begin{pmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \vdots \\ \mathbf{S}_k \end{pmatrix}. \tag{15}$$

Let $\mathbf{\Phi}$ be a $k \times k$ block diagonal matrix whose diagonal components are all equal to the mixing matrix $\mathbf{A}$, while other components are $c \times l$ null matrixes, i.e.,

$$\mathbf{\Phi} = \begin{pmatrix} \mathbf{A}_{c\times l} & \mathbf{O}_{c\times l} & \cdots & \mathbf{O}_{c\times l} \\ \mathbf{O}_{c\times l} & \mathbf{A}_{c\times l} & \cdots & \mathbf{O}_{c\times l} \\ \vdots & \vdots & & \vdots \\ \mathbf{O}_{c\times l} & \mathbf{O}_{c\times l} & \cdots & \mathbf{A}_{c\times l} \end{pmatrix}. \tag{16}$$

Then the mixed system Equation (1) can be reformulated as:

$$\underbrace{\begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_k \end{pmatrix}}_{\mathbf{Y}} = \underbrace{\begin{pmatrix} \mathbf{A}_{c\times l} & \mathbf{O}_{c\times l} & \cdots & \mathbf{O}_{c\times l} \\ \mathbf{O}_{c\times l} & \mathbf{A}_{c\times l} & \cdots & \mathbf{O}_{c\times l} \\ \vdots & \vdots & & \vdots \\ \mathbf{O}_{c\times l} & \mathbf{O}_{c\times l} & \cdots & \mathbf{A}_{c\times l} \end{pmatrix}}_{\mathbf{\Phi}} \underbrace{\begin{pmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \vdots \\ \mathbf{S}_k \end{pmatrix}}_{\mathbf{X}} \quad . \tag{17}$$

The matrix form of Equation (17) is:

$$\mathbf{Y}_{ck\times\frac{n}{k}} = \mathbf{\Phi}_{ck\times lk}\mathbf{X}_{lk\times\frac{n}{k}}. \tag{18}$$

Let $\mathbf{y}_j$ $(j = 1, 2, \ldots, \frac{n}{k})$ be the j-th column of $\mathbf{Y}$, then $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{\frac{n}{k}})$. Similarly, $\mathbf{x}_j$ $(j = 1, 2, \ldots, \frac{n}{k})$ represents j-th column of $\mathbf{X}$, then $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{\frac{n}{k}})$.

Since each column of $\mathbf{S}$ in Equation (1) is sparse, each $\mathbf{x}_j$ is also sparse in Equation (14). Obviously Equation (18) can be interpreted as a DCS model similar to Equation (3), where $\mathbf{\Phi}$ is the measurement matrix, $\mathbf{x}_j$ and $\mathbf{y}_j$ are compressed signal and measurement respectively. If each column of $\mathbf{S}$ is K-sparse with $K = K_c + K_j$, the total sparsity of $\mathbf{X}$, which is denoted by K_{total}, is:

$$K_{total} = kK_c + \sum_{j=1}^{n} K_j. \tag{19}$$

In CS theory, it requires $\mathbf{\Phi}$ with low coherence to recover $\mathbf{x}_j$ $(j = 1, 2, \ldots, \frac{n}{k})$ exactly from the measurements $\mathbf{y}_j$ $(j = 1, 2, \ldots, \frac{n}{k})$ [12]. The coherent of $\mathbf{\Phi}$ is defined as the largest correlation between any two columns of $\mathbf{\Phi}$:

$$\mu(\mathbf{\Phi}) = \max_{1\le i,j\le kl, i\ne j} |< \phi_i, \phi_j >|, \tag{20}$$

where $< \phi_i, \phi_j >$ is the inner product between ϕ_i and ϕ_j. CS requires the sensing matrix $\mathbf{\Phi}$ with low coherence, in other word, the coherence of measurement $\mathbf{\Phi}$ should be as small as possible. Essentially, the magnitude of $\mu(\mathbf{\Phi})$ is only related to $\mathbf{A}$, that is,

$$\mu(\mathbf{\Phi}) = \max_{1\le i,j\le l, i\ne j} |< \mathbf{a}_i, \mathbf{a}_j >|, \tag{21}$$

where $\mathbf{a}_i$ is the i-th column of $\mathbf{A}$. Gaussian matrices have been proved to satisfy the condition of coherence [12]. In this paper, Gaussian matrixes are taken as $\mathbf{A}$.

Following the discussion above, the recovering of sources for UBSS model can be interpreted as the reconstruction of multi-signal ensemble for DCS model. However, one may propose that it is not needed to split the data into blocks as the original form (Equation (1)) is already in the same format as the DCS problem. Thus the data can be processed block by block or stack them together and solve the problem simultaneously. The question is, in CS theory, for a K-sparse signal, exact recovery needs its dimension far exceeds its sparsity [13]. In this paper, if we regard Equation (1) as the DCS problem directly, each column of $\mathbf{S}$ is the target to be recovered. Note that these columns are l-dimensional, so it requires $l \gg K$ for recovering all columns. That means the number of signals l should far exceed the sparsity. These limits significantly the applicability of this approach as the number of signals of UBSS is not a very large value in real-world scenarios. From this perspective, we transform Equation (1) to Equation (17) so that lk (the dimension of each $\mathbf{x}_j$) far exceeds kK (the sparsity of $\mathbf{x}_j$). Thus, the UBSS problem can be solved by DCS algorithms.

2.4. **Sparsity adaptive matching pursuit algorithm for DCS.** Sparsity adaptive matching pursuit (SAMP) algorithm for DCS (referred as DCS-SAMP) was proposed in [10] as a new iterative greedy algorithm for multi-signal ensembles scenario. DCS-SAMP can reconstruct input signals from corresponding measurements jointly without any prior information of their sparseness. The algorithm is enlightened by SAMP algorithms [14] for single-signal. It has been proved that DCS-SAMP has nice reconstruction accuracy of the same order as that of linear programming (LP) optimization methods. On the other hand, DCS-SAMP demonstrates lower computational complexity, compared with that of orthogonal matching pursuit techniques, especially for exact sparse signals. The theoretical analysis illustrates that DCS-SAMP can reconstruct arbitrary sparse signals correctly in noiseless environment if the sensing matrix satisfies the condition of coherence. The whole reconstruction based on DCS-SAMP is composed of several stages, with several iterations at each stage. For each DCS-SAMP stage, only a small number of indices in the correct support set of the multi-signal ensembles are determined, then a series of iterations are performed to improve the estimate for higher accuracy. After convergence at this stage, the size of the estimated list is increased and other coordinates in the support set are estimated in next stage. The algorithm iterates continuously until the real support set is estimated completely. Then the whole estimations of the multi-signal ensembles are acquired via solving a least-square problem.

2.5. **The DCS-based algorithm for UBSS.** For clarity, some notations we propose are listed as follows.

r: iteration index.

$\mathbf{A}^{+}$: the pseudo-inverse of $\mathbf{A}$.

$\mathbf{A}_I$: a submatrix of $\mathbf{A}$ which is composed of the columns of $\mathbf{A}$ with indices $i \in I$.

$\mathbf{x}(I)$: a vector composed of the entries of vector $\mathbf{x}$ with indices $i \in I$.

F: the finalist of $\mathbf{X}$.

F_r: the finalist in the r-th iteration.

s: the step size of DCS-SAMP.

q: stage index.

$\mathbf{res}_r$: the residue in the r-th iteration.

β: size of the finalist.

$\|\mathbf{A}\|_{row-\infty}$: taking $\infty - norm$ of matrix $\mathbf{A}$ row by row.

$Max(\mathbf{x}, b)$: the set of indices corresponding to the b largest amplitude components of $\mathbf{x}$.

The main procedure of the DCS-based UBSS can be listed as follows.

Input: mixing matrix $\mathbf{A}$*, observation* $\mathbf{Z}$*, the number of blocks* k*.*

Step 1: separate $\mathbf{Z}$ into k blocks and stack all blocks to obtain $\mathbf{Y}$ according to Equation (13). Correspondingly, the measurement matrix is generated according to Equation (16). Thus the whole UBSS model is reformulated to a DCS model. $\mathbf{X}$ and $\mathbf{Y}$ are equivalent to the compressed signals and corresponding measurements, respectively.

Step 2: initialize the parameters used in the algorithm as follows: $\mathbf{res_0} = \mathbf{Y}$, $F_0 = \phi$, $\beta = p$, $r = 1$, $q = 1$, $\hat{\mathbf{x}}_j = \mathbf{0}$ $(j = 1, 2, \ldots, \frac{n}{k})$.

Step 3: obtain a final test F in the r-th iteration, i.e., $C_r = F_{r-1} \cup Max(\left\|\mathbf{\Phi}^{\mathrm{T}}\mathbf{res_{r-1}}\right\|_{row-\infty}, \beta)$ (make candidate list), $F = Max(\left\|\mathbf{\Phi}^{+}_{C_r}\mathbf{Y}\right\|_{row-\infty}, \beta)$ (final test).

Step 4: generate a new residue based on the final test in Step 3, i.e., $\mathbf{res} = \mathbf{Y} - \mathbf{\Phi}_F\mathbf{\Phi}^{+}_F\mathbf{Y}$.

Step 5: (i) if the new residue satisfies the halting condition, e.g., $\|\mathbf{res}\|_2$ is smaller than a given threshold, quit the iteration; (ii) else if $\|\mathbf{res}\|_2 \geq \|\mathbf{res_{r-1}}\|_2$, increase the stage index and update the size of finalist, i.e., $q = q + 1$, $\beta = q \times s$; (iii) else update the final list, residue and iteration index, i.e., $F_r = F$, $\mathbf{res_r} = \mathbf{res}$, $r = r + 1$.

Step 6: repeat Step 3 to Step 5 until halting condition is true.

Step 7: the entries of $\hat{\mathbf{x}}_j$ $(j = 1, 2, \ldots, \frac{n}{k})$ with coordinates in the finalist are obtained by solving a least-square problem, i.e., $\hat{\mathbf{x}}_j(F) = \mathbf{\Phi}^{+}_F\mathbf{y}_j$, while other entries are zeros.

Step 8: reformulate $\hat{\mathbf{X}}$ according to Equation (14), thus the estimation of $\mathbf{S}$ is obtained.

Note that s is the step size of DCS-SAMP. Except the only requirement $s \leq K$, for signals with decayed magnitudes, small s is preferable while large s is advantageous for binary signals [10].

3. **The Proposed Method vs. Existing Methods in Terms of Complexity.** In this section, the proposed method is compared with other two methods at the source reconstruction stage of UBSS, in terms of computational complexity.

If each $\mathbf{x}_j$ is N-dimensional and the measurement matrix is $m\times N$, let $K_s = s\cdot\lceil K_{total}/s\rceil$, where $\lceil\cdot\rceil$ denotes the round up operation, the complexity of DCS-SAMP is upper-bounded by $O(K_s mN)$ [10]. Note that the size of the measurement matrix is $cl \times kl$ in Equation (18), each $\mathbf{x}_j$ is kl-dimensional, correspondingly, the complexity of DCS-SAMP is upper-bounded by $O(K_s clk^2)$.

For UBSS, some methods have been proposed to separate signals at the stage of recovering source signals, such as the shortest path decomposition method [3,4] and basis pursuit method [15]. However, these methods are computationally demanding. The principle of shortest path decomposition method can be explained as follows: firstly, select c columns from $\mathbf{A}$ randomly to form a sub-matrix which is denoted by $\mathbf{A}_c = (\mathbf{A}_{i_1}, \mathbf{A}_{i_2}, \dots, \mathbf{A}_{i_c})$, where $i_1, i_2, \dots, i_c$ are the column indices. Obviously, there are $\begin{pmatrix} l \\ c \end{pmatrix}$ sub-matrixes like $\mathbf{A}_c$. Secondly, $\mathbf{S}$ is reconstructed according to Equation (22):

$$\begin{cases} (\mathbf{s}_{i_1}(t), \mathbf{s}_{i_2}(t), \dots, \mathbf{s}_{i_c}(t))^T = \left(\mathbf{A}_c^{\min}\right)^{-1}\mathbf{Z}(t) \\ \mathbf{s}_j(t) = \mathbf{0} \qquad (j \neq i_1, i_2, \dots, i_c) \end{cases} . \tag{22}$$

$\mathbf{A}_c^{\min}$ is denoted as the sub-matrix that makes $\sum\limits_{j=1}^{l} |\mathbf{s}_j(t)|$ minimum. Finally, the operator $(\mathbf{A}_c^{\min})^{-1}\mathbf{Z}(t)$ is the estimation of $\mathbf{S}$ at the time t. As can be seen, the computational complexity of shortest path method mainly lies in the calculation of inverse matrix [4]. Practically, with the increase of the number of sources l, calculating $(A_c^{\min})^{-1}Z(t)$ is computationally more demanding. However, the complexity of DCS-SAMP is upper-bounded by $O(K_s clk^2)$, there is no significant increase in the computational complexity of the proposed method for increased l (see more details in Table 1).

Another reconstructing method, i.e., basis pursuit (BP) essentially belongs to the linear program method. From [14], if signal $\mathbf{x}$ is N-dimensional, BP has a reconstruction complexity of $O(N^3)$. Note that each $\mathbf{x}_j$ is kl-dimensional in Equation (18), and basis pursuit is implemented on $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{\frac{n}{k}}$ respectively. So the total complexity of BP is $O(nl^3k^2)$. Since $c < l$, $K_s \ll nl$ (nl is the number of entries of source matrix), $K_s clk^2$ is significantly less than nl^3k^2, thus the computational complexity of DCS-SAMP is far less than that of BP, which shows that the speed of DCS-SAMP is significantly faster than that of BP at the reconstruction stage.

4. **Numerical Simulations.** In this section, we conduct several computer simulations to evaluate the performance of the approach proposed in Section 2. The true source matrix $\mathbf{S}$ is supposed to be known and stored as a reference for comparison with the results obtained by the proposed method. The $c\times l$ mixing matrix $\mathbf{A}$ is created whose entries are chosen form Gaussian distribution with mean 0 and variance 1. This matrix $\mathbf{A}$ generically satisfies the condition discussed in Section 2. Several trials are conducted to explore the proposed method, including its intrinsic characteristics and advantages compared with other approaches for UBSS. In each trial, a different realization of the mixing matrix $\mathbf{A}$ is used.

4.1. **Performance of DCS-based UBSS.** For convenience and without loss of generality, we assume the source matrix $\mathbf{S}$ is K-sparse in time domain, i.e., at each column of $\mathbf{S}$, there are only K sources with nonzero entries, while others are zero. The indices of the nonzero sources are generated randomly from a discrete uniform distribution, and the amplitudes of the sources are generated randomly from a standard Gaussian distribution.

We take $c = 10$, $l = 20$, $n = 200$, $k = 10$, i.e., there are 20 source signals with 200 data points. The known source matrix $\mathbf{S}$ is created with 1 "major source". Moreover, each column of $\mathbf{S}$ is assumed to be 3-sparse. Since $\mathbf{S}$ is divided into 10 blocks, the total sparsity is $K_{total} = 10 + 200 \times 2 = 410$. Fifty trials are run to recover $\mathbf{S}$ by the proposed method. To evaluate the performance of the proposed algorithm, the average errors between $\mathbf{s}_j$ $(j = 1, 2, \ldots, l)$ and the corresponding estimations $\hat{\mathbf{s}}_j$ $(j = 1, 2, \ldots, l)$ are calculated as follows:

$$error_j = E(\|\mathbf{s}_j - \hat{\mathbf{s}}_j\|_2 / \|\mathbf{s}_j\|_2) \quad (j = 1, 2, \ldots, l). \tag{23}$$

The smaller the average errors are, the better the performance is. The calculation of mathematical expectation is replaced by an average over the number of trials. The average errors of recovery are represented in Figure 1. From Figure 1, all $error_j$s are very close to zero, which shows the capability of the proposed method for separation with 20 sources.

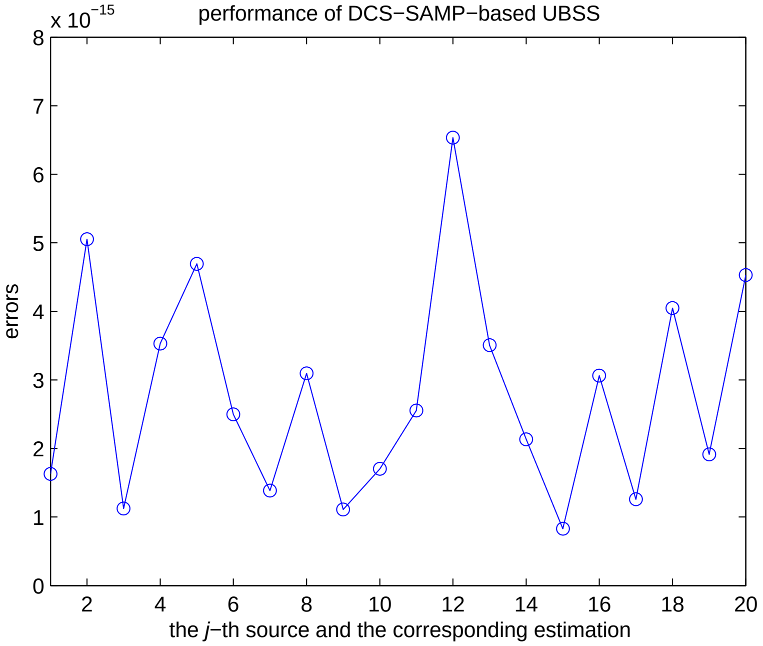

FIGURE 1. The average errors between j-th source and the corresponding estimation $\hat{\mathbf{s}}_j$ $(j = 1, 2, \ldots, l)$. The numerical values on x-axis denote j-th source and the corresponding estimation, and those on y-axis represent average errors between them.

4.2. **Comparison with other methods.** Assume each column of $\mathbf{S}$ is 2-sparse, the indices of nonzero entries at each column are chosen randomly, while $c = 8$, $k = 10$, $n = 200$. Table 1 shows the comparison of complexity among shortest path decomposition, BP and DCS-SAMP for reconstructing source $\mathbf{S}$. In order to obtain a measure of the computational complexity, the cpu times (second) for each of the methods (as l is varied) averaged over 100 trials is tabulated. From Table 1, the proposed method is computationally less demanding than other two methods and this trend is more obvious as l is increased, which indicates the advantages of DCS-based UBSS. It is also observed from Table 1 that there is no significant increase in the computational complexity of DCS-SAMP and BP with the increase of l, and the former performs faster than the latter. Using a value of $l > 11$,

Table 1. Comparison of averaged cpu times (sec) of three methods

l	9	10	11	12	13	14	15
DCS-SAMP	0.0193	0.0234	0.0245	0.0287	0.0291	0.0363	0.0368
Shortest Path Decomposition	0.0891	0.1822	0.6376	1.7321	3.6243	11.2212	42.2278
BP	0.8281	0.8719	1.0612	1.0695	1.5736	1.8659	1.9544

shortest path decomposition performs significantly slower than BP, while BP is slightly worse than shortest path decomposition in terms of cpu time with $l \leq 11$. When there are 15 source signals, the cpu time of the proposed method is far less than that of other two methods. When l varies from 9 to 15, the cpu time of the proposed method is increased by 1 times approximately, while this value is about 473 times for shortest path decomposition.

4.3. **The relationship between successful reconstruction and sparsity.**

Since the sparsity of DCS-SAMP is based on cardinality of the sparse common part K_c and the sparse innovation part K_j, we try to explore the relationship between probability of successful reconstruction and sparsity. "Successful" is defined for the sum of $error_j$ ($j = 1, 2, \ldots, l$) is no more than 10^{-6}. The frequency of success in 100 trials is used to calculate the probability of success.

4.3.1. *The relationship between successful reconstruction and K_c.* We take $c = 8$, $n = 200$, $k = 10$ to investigate how the cardinality of the sparse common part K_c affects the probability of successful reconstruction. A known $c \times l$ source matrix $\mathbf{S}$ with r ($r < l$) "major sources" while each of other rows having p ($p < n$) nonzero entries is created. The indices of r nonzero rows are chosen randomly from discrete uniform distributions. From Equation (7), K_c is rk, and

$$K_{total} = K_c + \sum_{j=1}^{\frac{n}{k}} K_j = rk + (l - r)p = pl + (k - p)r. \tag{24}$$

Here p is fixed to be 6, and $K_{total} = 6l + (k - 6)r$. For instance, if $r = 3$, $l = 20$, $K_{total} = 6 \times 20 + (10 - 6) \times 3 = 132$. Then 100 trials are run to reveal the relationship between probability of successful reconstruction and l when r varies. The specific result is represented in Figure 2.

In Figure 2, r is fixed in each set of trials, and the probability of successful reconstruction is calculated for the proposed method and different l. From Figure 2, no matter $r = 1$, 2 and 3, reconstructions are impossible when $l \geq 30$. Particularly, the probability of successful reconstruction decreases slowly unless l is more than 20 for $r = 1$. In case of $r = 3$, the curve of probability of successful reconstruction is uniformly lower than that obtained for $r = 1$ and 2. Especially, the probability of successful reconstruction is very small and the curve decreases rapidly for $r = 3$ as l increases.

It is also observed from Figure 2 that no matter $r = 1$, 2 or 3, the probability of successful reconstruction decreases as l increases. From CS theory, the core of reconstructing $\mathbf{x}_j$ $\left(j = 1, 2, \ldots, \frac{n}{k}\right)$ is to estimate their support sets. Since each $\mathbf{x}_j$ is a lk-dimensional vector, the increase of l results in rapid increase of dimension of $\mathbf{x}_j$. For instance, if l is increased by a factor of 1, the dimension of $\mathbf{x}_j$ will be increased by a factor of k. For K-sparse $\mathbf{x}_j$, there are $\begin{pmatrix} lk \\ K \end{pmatrix}$ possible support sets for $\mathbf{x}_j$. Obviously, it is more computationally demanding to find the correct support sets with the increase of l.

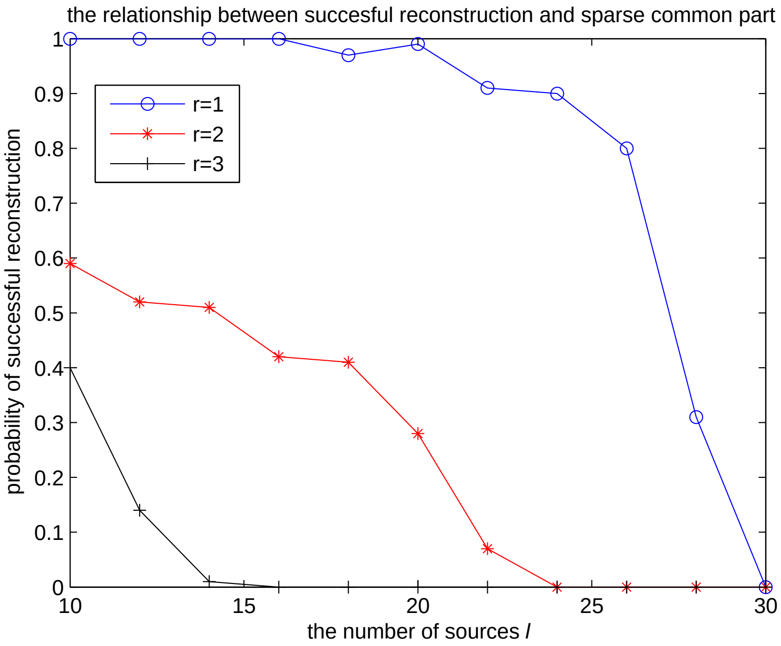

Figure 2. Probability of successful reconstruction obtained using the proposed method with $p = 6$ as l varies from 1 to 3. The numerical values on x-axis denote the number of sources and those on y-axis represent the probability of successful reconstruction.

Moreover, for each value of l, the probability of successful reconstruction reduces significantly with the increase of r. Actually, as $K_{total} = 6l + (k - 6)r$, if r is increased by a value of 1, correspondingly, for a large k, K_{total} will be increased by a value of $k - 6$. Larger K_{total} prevents x_j from being reconstructed correctly. As shown in Figure 2, in the case of $r = 3$, reconstruction results are very disappointing.

4.3.2. *The relationship between successful reconstruction and* K_j. The following discussions are focused on how the cardinality of the sparse innovation part K_j affects the probability of success. Compared with 4.3.1, two quantities vary in these simulations: p and the number of sources signals l. In the Monte Carlo simulations, the parameters are taken as $c = 8$, $n = 200$, $k = 10$, $r = 1$, correspondingly, $K_{total} = k + (l - 1)p$. 100 trials are performed to observe the probability of successful reconstruction as l is varied. These results of probability of successful reconstruction are plotted in Figure 3.

In Figure 3, p is fixed in each set of trials, and the probability of successful reconstruction is calculated for the proposed method as l is varied. Figure 3 shows that reconstructions are impossible for any value of p when $l \geq 35$. For $p = 4$, the probability stays at 1 until $l > 25$, while the largest decrease is obtained by increasing the number of sources from 25 to 30. In case of $p = 16$, the probability is very small. The curve decreases rapidly as l is increased and it is uniformly lower than curves for $p = 4$, 8 and 12.

It can also be seen in Figure 3 that, for any value of p, the probability of successful reconstruction lowers as l is increased. On the other hand, for each value of l, the probability of successful reconstruction reduces significantly as p is increased. Similar to the discussions in 4.3.1, since $K_{total} = k + (l - 1)p$, either the increase of l or the increase of p leads to fast increase of K_{total}. Particularly, if p is increased by 1, K_{total} will be increased by $l - 1$. Correspondingly, if l is increased by 1, K_{total} will be increased by p, which results in the rapid increase of K_{total}. As mentioned in 4.3.1, larger K_{total} keeps x_j from

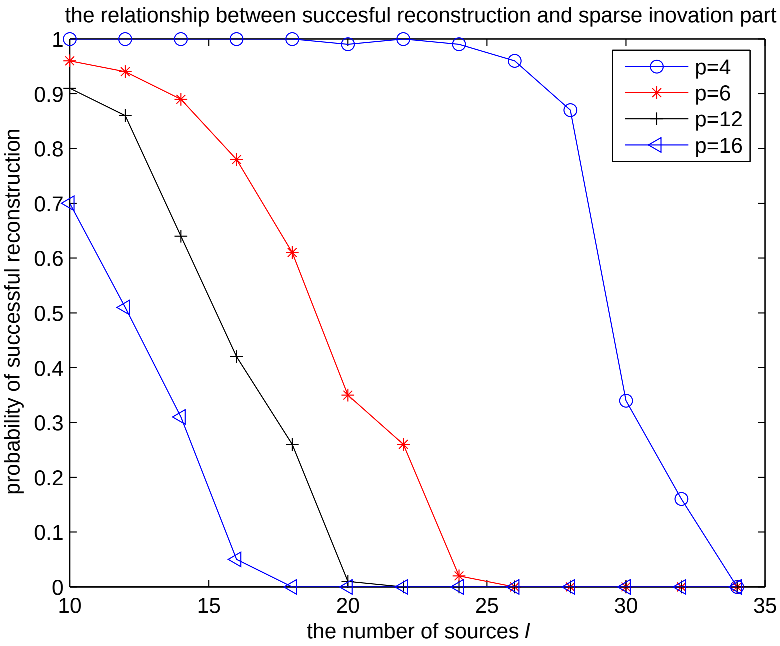

FIGURE 3. Probability of successful reconstruction obtained using the proposed method as l is varied for $p = 4$, 8, 12 and 16. The numerical values on x-axis denote the number of sources, and those on y-axis represent the probability of successful reconstruction.

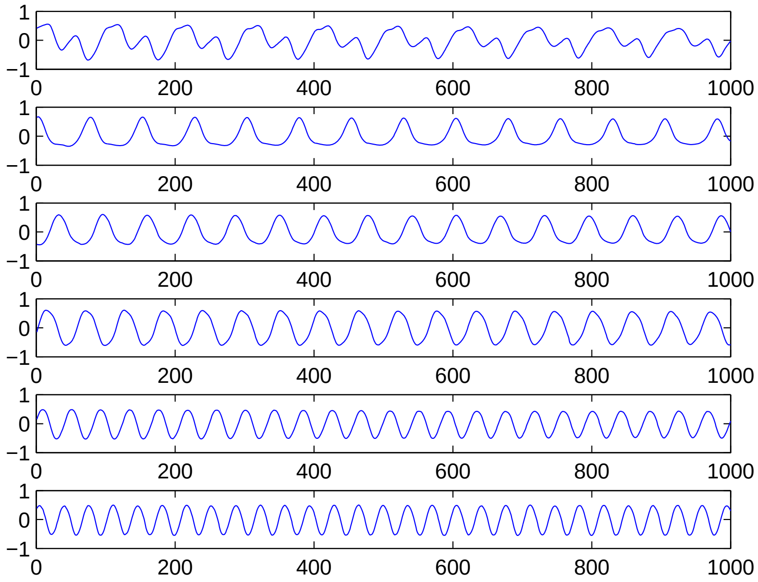

FIGURE 4. Plot of six speech source signals

being reconstructed correctly. As shown in Figure 3, when p is equal to 16, the result of reconstruction is very frustrating.

4.4. **Application for separation of speech signals.** In order to evaluate the performance of the proposed method in a more realistic case, we demonstrate the separation process of six speech signals for the convenience of intuitionistic description. These signals consist of 1000 data points, which are presented in Figure 4. Note that these speech signals are not sparse in time domain but compressible in frequency domain, discrete cosine transform (DCT) is applied on each signal. The corresponding coefficients are presented in Figure 5. It is observed that, for each signal, only few of its coefficients are larger than zero while others are very close to zero. Taking into account that sparsity is required for UBSS, instead of mixing these coefficients directly, we keep only a few large coefficients for each signal. A threshold is selected for this purpose, that is, all coefficients that are larger than the threshold are kept unchanged and all the others are changed to zero. That means the smaller the threshold is, the more coefficients that contain information of signals are kept. However, retaining too many coefficients may lead to insufficiency of sparsity. Through lots of tests, 0.008 is a feasible threshold for these signals to satisfying the sufficiency. The thresholded coefficients are the targets to be mixed and separated in the following steps.

After these pretreatments, the thresholded coefficients are mixed by a random Gaussian matrix. Then the proposed method is applied to separate these coefficients from observations and the recovery result is demonstrated in Figure 6. Finally, the separation result is obtained by using inverse discrete cosine transform (IDCT) and is presented in Figure 7. Comparing Figure 4 with Figure 7, it is observed that the error between each source signal and corresponding estimation is very small, where the error is defined as Equation (25), i.e.,

$$error(j) = \|\mathbf{s}_j - \hat{\mathbf{s}}_j\|_2 / \|\mathbf{s}_j\|_2 \quad (j = 1, 2, \ldots, 6), \tag{25}$$

where $\mathbf{s}_j$ and $\hat{\mathbf{s}}_j$ represent jth source and corresponding estimation respectively. The errors for each source signal in Figure 4 are 0.0053, 0.0051, 0.0055, 0.0054, 0.0063, 0.0061, which are acceptable in practice. So the effectiveness of the proposed method is verified for realistic signal condition.

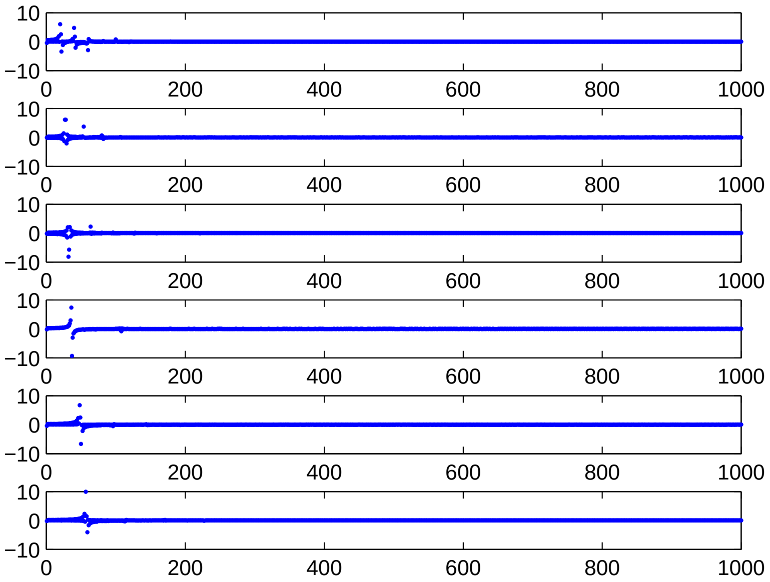

Figure 5. Plot of DCT coefficients of six source signals

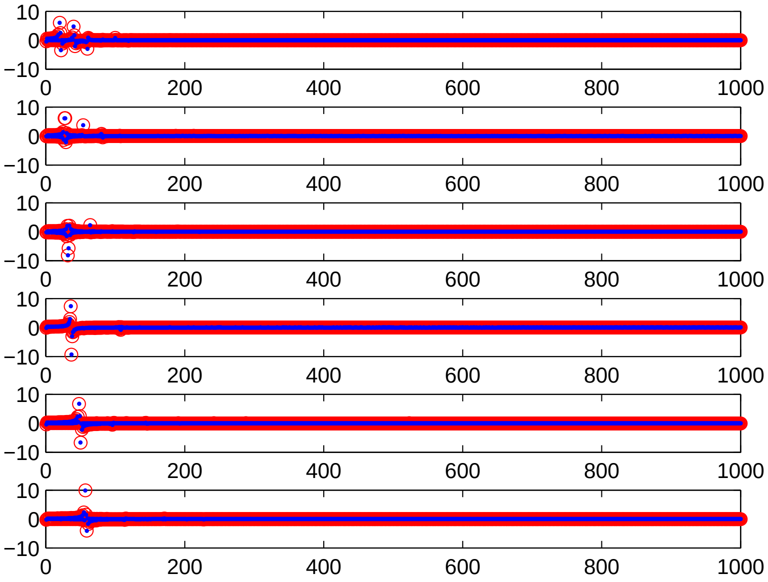

FIGURE 6. Thresholded coefficients and corresponding estimations. Blue dots: thresholded coefficients. Red circle: corresponding estimations.

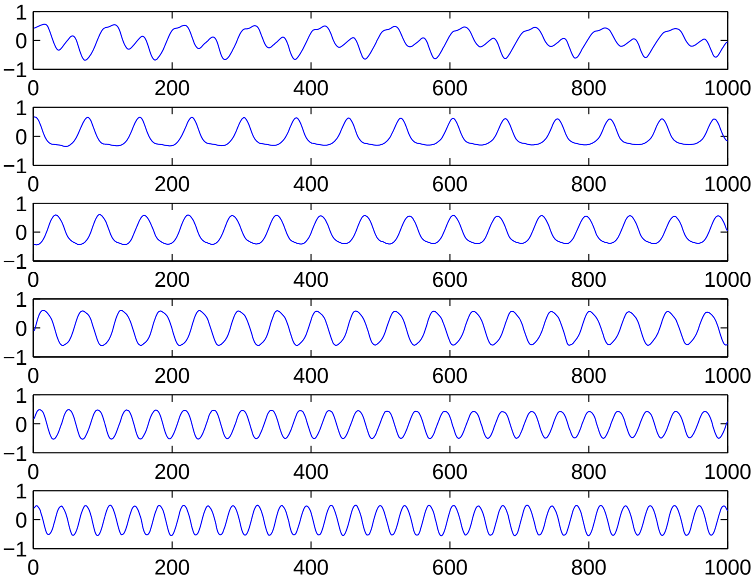

FIGURE 7. The plot of the separated signals by the proposed method

As comparison, BP and subspace pursuit (SP) are also used to settle the UBSS problem. As a CS algorithm, SP is used for UBSS in the same way as mentioned in [8], i.e., all sources are connected to be a long vector while the UBSS model is reformulated into a CS model [8]. Figure 8 and Figure 9 represent the separated signals performed by BP and SP respectively. The errors based on the three methods are listed in Table 2. Obviously, the errors resulted by the proposed method are significantly smaller than that of SP and BP, which demonstrates the application potential of the proposed method.

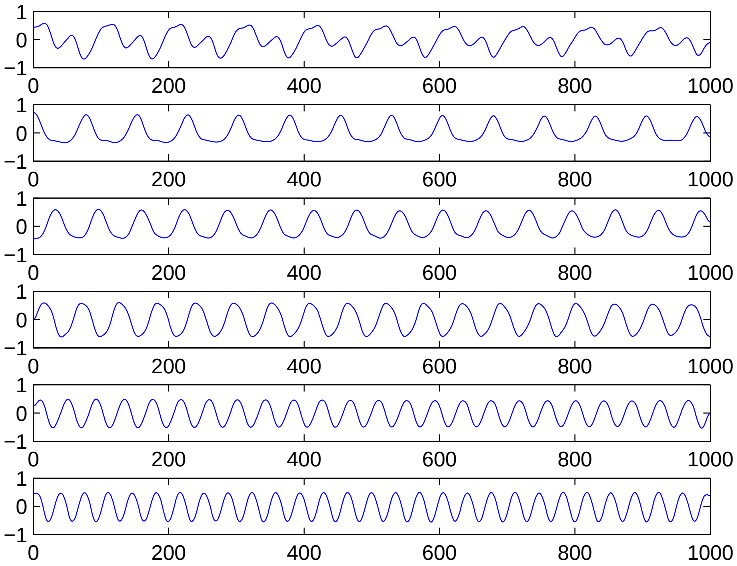

FIGURE 8. Plot of the separated signals by BP

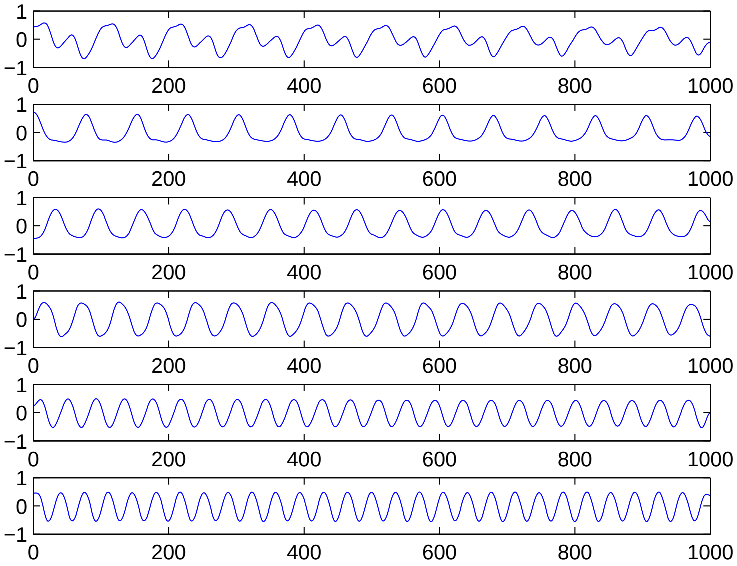

FIGURE 9. Plot of the separated signals by SP

TABLE 2. Separation errors based on three methods

Errors	Signal 1	Signal 2	Signal 3	Signal 4	Signal 5	Signal 6
Our method	0.0053	0.0051	0.0055	0.0054	0.0063	0.0061
BP	0.0278	0.0298	0.0290	0.0283	0.0328	0.0284
SP	0.0281	0.0342	0.0335	0.0312	0.0338	0.0301

5. **Conclusions.** In this paper, the relationship between UBSS and DCS is interpreted by equivalent transformation. A DCS-based algorithm is proposed to settle UBSS problem with introduced feature. In a series of simulations, we demonstrate how the performance of the proposed method varied with sparsity and the number of sources. The improvement in computational efficiency of the proposed method with larger source number (larger than 10) can be seen from the simulations, as compared with other UBSS approaches. The capability to separate compressible speech signals demonstrate the application potential of our method. Further research is required to develop a computationally efficient version of the proposed method in noisy environment as well as to better analytically characterize the performance of the method.

Acknowledgment. This work was supported by Natural Science Foundation of China (Grant No. 61203287, 61302138, 11126274), Natural Science Foundation of Hubei Province (No. 2013CFB414), the Special Fund for Basic Scientific Research of Central Colleges, China University of Geosciences (Grand No. CUGL130247) and the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) under Grants CUGL140422, QTZZ201405. The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] A. Ozerov and C. Fevotte, Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation, *IEEE Trans. Audio, Speech, and Language Processing*, vol.18, no.3, pp.550-563, 2010.

[2] Q. M. Yi, Blind source separation by weighted K-means clustering, *Journal of Systems Engineering and Electronics*, vol.19, no.5, pp.882-887, 2008.

[3] F. J. Theis, W. E. Lang and C. G. Puntonet, A geometric algorithm for overcomplete linear ICA, *Neurocomputing*, vol.56, pp.381-398, 2004.

[4] P. Bofill and M. Zibulevsky, Underdetermined blind source separation using sparse representations, *Signal Processing*, vol.81, no.11, pp.2353-2362, 2001.

[5] Y. Li, A. Cichocki and S. Amari, Analysis of sparse representation and blind source separation, *Neural Computation*, vol.16, no.6, pp.1193-1234, 2004.

[6] Y. Li, S. I. Amari, A. Cichocki and D. W. C. Ho, Underdetermined blind source separation based on sparse representation, *IEEE Trans. Signal Processing*, vol.54, no.2, pp.423-437, 2006.

[7] M. Zibulevsky and B. A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, *Neural Computation*, vol.13, no.4, pp.863-882, 2001.

[8] G. Z. Bao, Z. F. Ye, X. Xu and Y. Y. Zhou, A compressed sensing approach to blind separation of speech mixture based on a two-layer sparsity model, *IEEE Trans. Audio, Speech and Language Processing*, vol.21, no.5, pp.899-906, 2013.

[9] Y. Q. Li, C. R. Yin, H. W. Xu, H. P. Li, N. Fu and Y. Q. Zhang, Blind separation of speech sources in multichannel compressed sensing, *IEEE International Instrumentation and Measurement Technology Conference*, Minneapolis, MN, USA, pp.1771-1774, 2013.

[10] Q. Wang and Z. W. Liu, A robust and efficient algorithm for distributed compressed sensing, *Computers and Electrical Engineering*, vol.37, no.6, pp.916-926, 2011.

[11] E. Candes, J. Romberg and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. Information Theory*, vol.52, no.2, pp.489-509, 2006.

[12] Z. H. Wu, Y. Shen, Q. Wang and J. Liu, Blind source separation based on compressed sensing, *Proc. of the 6th International ICST Conference on Communications and Networking*, Harbin, China, pp.794-798, 2011.

[13] E. Candes, M. Wakin and S. Boyd, Enhancing sparsity by reweighted l_1 minimization, *Journal of Fourier Analysis and Applications*, vol.14, no.5, pp.877-905, 2008.

[14] T. T. Do, G. Lu, N. Nguyen and T. D. Tran, Sparsity adaptive matching pursuit algorithm for practical compressed sensing, *Proc. of the 42nd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, pp.581-587, 2008.

[15] S. S. Chen, D. L. Donoho and M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM Journal on Scientific Computing*, vol.20, no.1, pp.33-61, 1999.