Time-varying Mixing Matrix Identification for Underdetermined Blind Source Separation Based on Online Tensor Decomposition

Sunan Ge, Tao Xue, Rui Zhang, and Zhenzhong Hou

***Abstract*—Given that the mixing matrix of underdetermined blind source separation (UBSS) changes with the recording environment, offline UBSS methods encounter difficulty in satisfying the time-varying estimation demand. Therefore, in this work, an online tensor algorithm has been proposed to estimate the time-varying mixing matrix for separating an instantaneous linear underdetermined mixture. First, we construct a canonical polyadic tensor model by assuming individually correlated sources. Second, an online tensor algorithm is applied to decompose the canonical polyadic tensor model to ensure the accuracy of the time-varying mixing matrix. Finally, two types of data, including speech and biomedical signals, have been used to substantiate the effectiveness of the proposed algorithm in estimating the time-varying mixing matrix for UBSS. The results show that the developed online tensor algorithm is significantly superior to the conventional offline UBSS methods in terms of time consumption and accuracy.**

***Index Terms*—underdetermined blind source separation; online tensor decomposition; canonical polyadic decomposition; time-varying mixing matrix.**

I. INTRODUCTION

BLIND source separation (BSS) focuses on recovering the sources from observed signals without any prior information about the mixing process [1]. It is widely applied for modal property estimation [2], speech signal processing [3], and electroencephalographic (EEG) artifact removal [4]. According to the dimensionality of the sources and observations, BSS algorithms are always classified into three types: available blind source separation (ABSS) with the same numbers of sources and observations, overdetermined blind source separation (OBSS) with fewer sources than observations, and underdetermined blind source separation (UBSS) with larger numbers of sources than observations [5-6]. Compared with OBSS and ABSS, it is difficult to solve UBSS problems because of the inestimable mixing matrix and sources.

Usually, UBSS contains two essential tasks, i.e., estimating the mixing matrix, which is a critical step in UBSS, and recovering the sources. It is known that a perfectly estimated mixing matrix could contribute to improving the source recovery. Initially, under the hypothesis of the sparsity of sources in the time-frequency (TF) plane, sparse component analysis (SCA) was first proposed to estimate the mixing matrix for UBSS [7]. This sparsity of sources should be under W-disjoint orthogonality conditions [8-9] or emerging alone in a small set of adjacent TF windows [10] after a short-time Fourier transform (STFT). Based on the above conditions, some clustering algorithms, such as K-means [11] and fuzzy c-means [12], have been widely adopted to obtain the mixing matrix in the TF plane. Unfortunately, the robustness of these clustering algorithms is not well-suited for high-dimensional data or nonstationary signals with noise. Therefore, Dong et al. [13] proposed a modified similarity-based robust clustering method (MSCM) to improve the robustness of clustering algorithms. To reduce the sensitivity of clustering algorithms, Sun and colleagues [14] proposed the Hough transform to modify the cluster center to enhance the estimation accuracy of a mixing matrix. In addition, most of the SCA approaches require the automatic determination of a single-source point in the TF plane [15]. However, detecting a single or dominant source at every TF point in practice is difficult; hence, the mixing matrix cannot be accurately estimated, especially for low-sparsity signals.

To overcome this drawback, a tensor method without single-source point detection was proposed by Common [16]; this method estimates the mixing matrix for UBSS for a specific case with two existing observed signals and three sources. In his work, canonical polyadic (CP) decomposition, also called parallel factor analysis (PARAFAC), was exploited to obtain a mixing matrix because of its uniqueness under extremely mild conditions, especially its uniqueness in the "underdetermined" case. Ferreol et al. [17] suggested the fourth-order blind identification of underdetermined mixtures of sources (FOBIUM) method that does not require

Manuscript received March 2, 2020, revised December 11, 2020. This work is supported by the National Natural Science Foundation of China (Project No. 61473223 and No. 61873107), the Natural Science Foundation of Jiangsu Higher Education Institutions of China (Project No. 17KJB413001), the Youth Innovation Team of Shaanxi Universities, and the Shaanxi Provincial Special (Fund) Plan for Technological Innovation Guidance in 2020 (Project No. 2020CGXNG-012)

Sunan Ge is a Postdoctoral in the School of Mathematics of Northwest University, and a Lecturer in the School of Computer Science of Xi'an Polytechnic University, Shaanxi, 710048, P. R. China. (phone: +86-18805237886; e-mail: gesunan@ xpu.edu.cn).

Xue Tao is a Professor in the School of Computer Science of Xi'an Polytechnic University, Shaanxi, 710048, P. R. China. (phone: +86-13201771093; e-mail: xuetao@xpu.edu.cn)

Rui Zhang is a Professor in the School of Mathematics of Northwest University, Xi'an, Shaanxi, 710069, P. R. China. (phone: +86-18165386699; e-mail: rzhang@nwu.edu.cn).

Zhenzhong Hou is a Lecturer in the College of Materials Science and Engineering of Xi'an University of Science and Technology, Shaanxi, 710054, P. R. China. (phone: +86-18710309082; e-mail: hzzhong1981@yeah.net).

limitations on the number of the sources and observations when the individual source signals are dependent over some time interval. To improve the precision in the joint diagonalization matrix for UBSS, Lathauwer et al. [18] proposed the fourth-order-only blind identification (FOOBI) algorithm to decompose the fourth-order CP model with nonzero kurtosis of the sources. Furthermore, the second-order blind identification of underdetermined mixtures (SOBIUM) method was adopted by Lathauwer and his colleagues to estimate a mixing matrix and to consider the individual correlation of sources in time [19]. Tichavsky et al. revealed a weight-adjusted tensor method (UDSEP) to obtain the mixing matrix and to decompose a third-order CP modal with nonstationary signals [20]. In addition, assuming that the tensor elements are all nonnegative, Cichocki proposed the hierarchical alternating least squares (HALS) algorithm for acquiring the nonnegative mixing matrix [21-22]. In addition, an alternating least squares (ALS) algorithm was proposed to decompose the joint CP model to estimate the mixing matrix of two datasets when the data are involved in multiple datasets [23]. However, the ALS method always has a relatively poor convergence when handling a high-order tensor. To improve the convergence, an enhanced line search (ELS) method was used to estimate the mixing matrix instead of the ALS method [24].

Most of the above methods are offline algorithms for estimating the mixing matrix for UBSS. However, considering that the underlying data-generation process varies with time, the offline algorithm consumes considerable time when dealing with a time-varying mixing matrix. In this study, we focus on solving the time-varying mixing matrix identification for the UBSS problem with an online tensor method [25] to enhance efficiency. The second-order cumulant tensor is set by the spatial covariance matrices from the observations. A modified online tensor algorithm is proposed to decompose the time-varying second-order cumulant tensor to obtain the time-varying mixing matrix.

The rest of this paper is organized as follows. In Section 2, an online tensor method for estimating the mixing matrix for UBSS is introduced. Simulation results based on speech and biomedical signals are discussed in Section 3. Finally, conclusions are provided in Section 4.

II. TIME-VARYING MIXING MATRIX IDENTIFICATION FOR UBSS

A. Preliminaries

In this section, a brief review of the tensor and CP decomposition is provided to understand the online tensor algorithm. Simply speaking, a tensor is a multiway array or multidimensional matrix. The order of a tensor is the number of dimensions that are considered ways or modes. A tensor $\bar{X} \in £^{I_1 \times I_2 \times \mathrm{L} \times I_N}$ of order *N* indicates an *N*-way array with its $(i_1, i_2, \mathrm{L}\ i_N)$ th entry denoted by $\bar{X}_{i_1 i_2 \mathrm{L}\ i_N}$, where $i_n \in \{1, 2, \mathrm{L}\ , I_n\}$, for $1 \le n \le N$. For the *N*-th order tensors, the two popular tensor decomposition methods are the Tucker model and the more restricted CP decomposition method. Compared with the Tucker model, the CP is more suitable in the UBSS case and can be denoted as a linear combination of rank-1 tensors without core parameters. The CP decomposition method is formulated as

$$\bar{X} = \sum_{r=1}^{R} \lambda_r a_r^{(1)} \circ a_r^{(2)} \circ \mathrm{L}\ \circ a_r^{(N)} \tag{1}$$

where $a_r^{(n)} \in £^{I_n}$, for $n = 1, 2, \mathrm{L}\ , N$, and $a_r^{(1)} \circ a_r^{(2)} \circ \mathrm{L}\ \circ a_r^{(N)}$ denote a rank-one tensor. The minimal number of rank-1 terms R is referred to as the rank of the tensor, which is the number of sources for UBSS. $\circ$ is the tensor outer, which can be written as the elementwise form of CP:

$$\bar{X}_{i_1 i_2 \mathrm{L}\ i_N} = \sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \mathrm{L}\ a_{i_N r}^{(N)} \tag{2}$$

Unfolding is known as matricization or flattening and represents tensors as matrices or multiway relationships. Unfolding refers to a tensor fiber and slice that are formed as a subset when the indices are fixed. A tensor fiber is a one-dimensional fragment of a tensor obtained by fixing all indices except for one. A tensor slice is a two-dimensional section or fragment of a tensor obtained by fixing all indices except for two indices and includes horizontal, lateral, and frontal slices. The mode-*n* unfolding of tensor $\bar{X}$ is denoted as $X_{(n)}$, and the mode-n fibers are arranged as the columns of a matrix. The matrix is expressed as

$$X_{(n)} = A^{(n)} \Lambda (A^{(n)} \odot \mathrm{L} \odot A^{(n+1)} \odot A^{(n-1)} \odot \mathrm{L} \odot A^{(1)})^T \tag{3}$$

in which, $A^{(n)} = [a_1^{(n)} \quad a_2^{(n)} \quad \mathrm{L} \quad a_R^{(n)}] \in ?^{I_n \times R}$, and $\Lambda = \mathrm{diag}(\lambda_1 \quad \lambda_2 \quad \mathrm{L} \quad \lambda_R)$. $\odot$ is the Khatri-Rao product, which is defined in [22].

As special cases, the third-order rank-one CP tensor is written as (see Figure 1)

$$\bar{T} = \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)} \in ?^{I_1 \times I_2 \times I_3} \tag{4}$$

where

$$\bar{T}_{i_1 i_2 i_3} = \sum_{r=1}^{R} a_{i_1 r} a_{i_2 r} a_{i_3 r} \tag{5}$$

The tensor can be represented in compact matrix forms by applying the unfolding representations of tensor $\bar{T}$

$$\begin{aligned} \bar{T}_{(1)} &= A^{(1)} (A^{(3)} \odot A^{(2)})^T \\ \bar{T}_{(2)} &= A^{(2)} (A^{(3)} \odot A^{(1)})^T \\ \bar{T}_{(3)} &= A^{(3)} (A^{(2)} \odot A^{(1)})^T \end{aligned} \tag{6}$$

The third-order CP, as a sum of rank-one tensors, is represented in Figure 1.

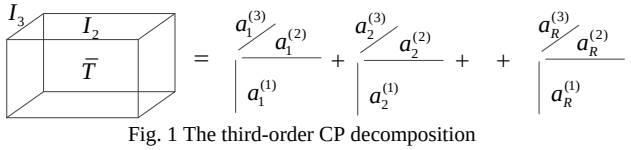

Fig. 1 The third-order CP decomposition

B. Problem Description

Here, we will consider the time-varying instantaneous linear mixture model

$$X(t) = A(t)S(t), \tag{7}$$

where $A(t)$ is a time-varying mixing matrix of $m \times R$, where

$m < R$. In (7), $S(t) = \left[s_1(t) \quad s_2(t) \quad \mathrm{L} \quad s_R(t) \right]^T$ denotes the sources that are R-dimensional vectors, and observations $X(t) = \left[x_1(t) \quad x_2(t) \quad \mathrm{L} \quad x_m(t) \right]^T$ are m-dimensional vectors. In (7), the number of observations is less than the number of sources ($m < R$); hence, it is a so-called UBSS problem. Accurately estimating the time-varying mixing matrix and then utilizing it to achieve the sources is the primal problem of UBSS. Here, we assume that the sources are individually correlated in time. The spatial covariance matrices of the observations satisfy

$$\begin{aligned} C_1(t) &= E\left\{ X_t X_{t+\tau_1}^H \right\} = A(t) D_1(t) A^H(t) \\ &\mathrm{M} \\ C_K(t) &= E\left\{ X_t X_{t+\tau_K}^H \right\} = A(t) D_K(t) A^H(t) \end{aligned} \tag{8}$$

where $D_k(t) = E\left\{ S_t S_{t+\tau_k}^H \right\}$ is the diagonal matrix. τ_k is the time delay and is equal to zero in the simple case, but here, it is denoted as $k = 1, 2, \mathrm{L}\ , K$. To acquire the unique time-varying mixing matrix, the covariance matrix $C_1(t), \mathrm{L}\ , C_K(t) \in R^{m \times m}$ is stacked in tensor $\bar{C}(t) \in R^{m \times m \times K(t)}$ and interpreted as a tensor decomposition.

Combined with (8), an online tensor model is constructed as follows:

$$\bar{C}(t) = \sum_{r=1}^{R} a_r(t) \mathrm{o} a_r^*(t) \mathrm{o} d_r(t) \tag{9}$$

where $a_r(t)$ is the mixing matrix vectors, and $a_r^*(t)$ is the complex conjugate of the mixing matrix vectors. $d_r(t)$ is the diagonal vector of $D_k(t)$. Equation (9) can be written as

$$\bar{C}_{ijk}(t) = \sum_{r=1}^{R} a_{ir}(t) a_{jr}^*(t) d_{kr}(t) \tag{10}$$

where $i = 1, 2, \mathrm{L}\ m$ and $j = 1, 2, \mathrm{L}\ m$. However, considering the time-varying mixing matrix in this situation, tensor decomposition is an online problem. Traditional offline tensor decomposition is not suitable for online problems. Thus, we propose an online UBSS method to decompose the time-varying tensor model. Depending on the definition of the tensor in matrix form, a tensor can be represented by

$$C(t) = (\mathrm{A}(t) \mathrm{e}\ \mathrm{A}^*(t)) D^T(t) \tag{11}$$

where superscripts g^T and g^H represent transposition and the complex conjugated transposition, respectively.

C. Sketch of Online Tensor Decomposition

In this section, the renewal sketch of the tensor is described as follows. First, the observed signals are updated by

$$X(t+1) = [X(t), x(t+1)] \tag{12}$$

where $X(t)$ denotes the observation at time t and $x(t+1)$ is the new vector at time $t+1$. Here, to simplify the representation of $A(t) \mathrm{e}\ A^*(t)$, we define $H(t) = A(t) \mathrm{e}\ A^*(t)$, where $H(t) \in \mathrm{£}^{mm \times R}$. Then, the matrix $C(t+1)$ unfolded by tensor $\bar{C}(t+1) \in \mathrm{£}^{m \times m \times K(t+1)}$ is described by

$$\begin{aligned} C(t+1) &= [C(t), c(t+1)] \\ &= (\mathrm{A}(t+1) \mathrm{e}\ \mathrm{A}^*(t+1)) D^T(t+1) \\ &= H(t+1) D^T(t+1) \end{aligned} \tag{13}$$

where $C(t) = H(t) D^T(t)$ is the matrix, which is unfolded by tensor $\bar{C}(t)$ at time t, and $c(t+1)$ is a new slice that is represented by a vector.

According to (11), we assume that $A(t) \mathrm{e}\ A^*(t)$ is a smooth variation between t and $t+1$, i.e., $A(t) \mathrm{e}\ A^*(t)$; $A(t+1) \mathrm{e}\ A^*(t+1)$. Thus, the diagonal matrix has been changed from t to $t+1$ by

$$D^T(t+1);\ [D^T(t), d^T(t+1)] \tag{14}$$

where $D(t+1)$ has an approximate time-shift structure and $D(t) \in \mathrm{£}^{K(t)+1}$ has a dimension that grows with time.

Then, $d^T(t+1)$ can be initially estimated by

$$d^T(t+1) = H^{\dagger}(t) c(t+1) \tag{15}$$

where $H(t) \in \mathrm{£}^{mm \times R}$ is replaced by $A(t) \mathrm{e}\ A^*(t)$. Then, the online algorithm is simplified by ignoring the Khatri-Rao product structure. The least squares update of $H(t+1)$ is given by

$$H(t+1) = C(t+1)(D^T(t+1))^{\dagger}. \tag{16}$$

Then, when the matrix $H(t+1)$ replaces $H(t)$, $d^T(t+1)$ can reupdate in (15). Finally, according to

$$\begin{aligned} &A(t+1) \mathrm{e}\ A^*(t+1) = H(t+1) \\ &= [a_1(t+1) \otimes a_1^*(t+1) \quad \mathrm{L} \quad a_R(t+1) \otimes a_R^*(t+1)] \end{aligned} \tag{17}$$

we can obtain $a_r(t+1)$ with the conjugate principal of right singular vector of matrix $H_r(t+1)$, in which $H_r(t+1) = \mathrm{unvec}([H(t+1)]_{:,r})$, $r = 1, 2\mathrm{L}\ , R$.

D. The Estimation of a Time-varying Mixing Matrix

Here, the exponential window time of the least squares criterion is given to evaluate the participation degree of past observations by

$$\min_{\{H(t+1),\, D(t+1)\}} (\phi^{EW}(t+1)) \tag{18}$$

with

$$\phi^{EW}(t+1) = \sum_{\tau=1}^{t+1} \lambda^{t+1-\tau} \left\| c(\tau) - H(t+1) d^T(\tau) \right\|^2 \tag{19}$$

where λ is the forgetting factor. The exponential time window causes all previously observed slices to contain different weight values at any time. Then, the matrix with weight coefficients can be defined by

$$C_{EW}(t+1) = C(t+1) \Lambda(t+1) \tag{20}$$

where $\Lambda(t+1) = diag([\lambda^{t/2}, \lambda^{t-1/2}, \mathrm{L}\ , \lambda^{1/2}, 1])$ is the weight coefficient matrix. Depending on the sketch of the online tensor decomposition, the updating rule of $C_{EW}(t+1)$ will be written as

$$C_{EW}(t+1) = [\lambda^{1/2} C(t), c(t+1)]. \tag{21}$$

Combined with (11), the matrix $C_{EW}(t+1)$ can be factorized as follows:

$$\begin{cases} C_{EW}(t) = (A(t) \text{ e } A^T(t))D^T(t)\Lambda(t) \\ C_{EW}(t) = U_{EW}(t)\Sigma_{EW}(t)V_{EW}^H(t) \end{cases} \quad (22)$$

To guarantee the uniqueness of the decomposition, there is a nonsingular matrix $W_{EW}(t) \in \pounds^{R\times R}$ such that

$$\begin{cases} A(t) \text{ e } A^T(t) = E_{EW}(t)W_{EW}(t) \\ D^T(t)\Lambda(t) = W_{EW}^{-1}(t)V_{EW}^H(t) \end{cases} \quad (23)$$

where $E_{EW}(t) = U_{EW}(t)\Sigma_{EW}(t)$. Once $W_{EW}(t) \in \pounds^{R\times R}$ has been determined, the decomposition of (24) is unique. At time $t+1$, (22) will become

$$\begin{cases} A(t+1) \text{ e } A^T(t+1) = E_{EW}(t+1)W_{EW}(t+1) \\ D^T(t+1)\Lambda(t+1) = W_{EW}^{-1}(t+1)V_{EW}^H(t+1). \end{cases} \quad (24)$$

The time-shift structure of $D(t+1)$ is adopted to link (23) to (24). Hence, the recursive update is explored by $W_{EW}(t+1)$ and $D(t+1)$. The estimations of $E_{EW}(t+1)$ and $V_{EW}(t+1)$ are acquired from $E_{EW}(t)$ and $V_{EW}(t)$. Here, the Bi-SVD1 technique [25-26] is exploited to satisfy the online singular value decomposition (SVD). Then, the matrix $D(t+1)$ can be acquired from

$$\begin{cases} D^T(t)\Lambda(t) = W_{EW}^{-1}(t)V_{EW}^H(t) \\ D^T(t+1)\Lambda(t+1) = W_{EW}^{-1}(t+1)V_{EW}^H(t+1) \end{cases} \quad (25)$$

where the dimensions of $V_{EW}(t)$ and $V_{EW}(t+1)$ are $K(t)\times R$ and $K(t+1)\times R$, respectively. To acquire a recursive update of $W_{EW}(t)$, we should define the following matrices:

$$\begin{aligned} V(t) &= V_{EW}(t) \\ V(t+1) &= [V_{EW}(t+1)]_{1:K(t),:} \\ v(t+1) &= [V_{EW}(t+1)]_{K(t+1),:}. \end{aligned} \quad (26)$$

According to the exponential window, we have

$$\begin{aligned} \lambda^{1/2}D^T(t)\Lambda(t) &= \lambda^{1/2}W_{EW}^{-1}(t)V_{EW}^H(t) \\ &= D^T(t+1)\Lambda(t+1) \\ &= W_{EW}^{-1}(t+1)V_{EW}^H(t+1). \end{aligned} \quad (27)$$

The definition of the matrices is combined with (26) to obtain

$$\lambda^{1/2}W_{EW}^{-1}(t)V^H(t) = W_{EW}^{-1}(t+1)V^H(t+1). \quad (28)$$

Then, $W_{EW}^{-1}(t+1)$ is acquired by

$$W_{EW}^{-1}(t+1) = \lambda^{1/2}W_{EW}^{-1}(t)V^H(t)[V^H(t+1)]^{\dagger}. \quad (29)$$

At the same time, $W_{EW}(t+1)$ can be obtained by

$$W_{EW}(t+1) = \lambda^{-1/2}V^H(t+1)[V^H(t)]^{\dagger}W_{EW}^{-1}(t). \quad (30)$$

To avoid the pseudoinverse calculation, the update rule of $[V^H(t+1)]^{\dagger}$ can be obtained by the pseudoinverse Lemma for rank-1 updates. It is written as

$$[V^H(t+1)]^{\dagger} = V(t+1)(I_R + \frac{v^H(t+1)v(t+1)}{1-\|v(t+1)\|^2}) \quad (31)$$

Then, depending on (25) and (26), the update rules of $D^T(t+1)$ are

$$d^T(t+1) = W_{EW}^{-1}(t+1)v^H(t+1) \quad (32)$$

and

$$D^T(t+1);\ [D^T(t), d^T(t+1)] \quad (33)$$

Finally, the update rule of the mixing matrix is

$$a_r^*(t+1) = H_r^H(t+1)a_r(t) \quad (34)$$

The online UBSS algorithm for estimating the mixing matrix is given as follows:

TABLE I
ONLINE UBSS ALGORITHM FOR ESTIMATING THE MIXING MATRIX

Input at time t:	The observation, the mixing matrix and the sources represented by $X(t)$, $A(t)$, and $S(t)$, respectively. The online tensor $\bar{C}(t)$ is established with the time-varying mixing matrix and individually correlated sources in (9).
update at time $t+1$:	
Step1:	Online tensor $\bar{C}(t+1)$ is established by (13), when the observation is updated by $X(t+1) = [X(t), x(t+1)]$ at time $t+1$;
Step2:	Bi-SVD1 is exploited to decompose online tensor $\bar{C}(t+1)$ with the exponential time window in order to obtain the update rules of $U_{EW}(t+1)$, $\Sigma_{EW}(t+1)$ and $V_{EW}^H(t+1)$;
Step3:	Combined with the definition matrices of (26), the matrices $[V^H(t+1)]^{\dagger}$ and $W_{EW}^{-1}(t+1)$ are acquired by (29) and (31), respectively;
Step4:	According to $D^T(t+1)\Lambda(t+1) = W_{EW}^{-1}(t+1)V_{EW}^H(t+1)$, the update rule of $D^T(t+1)$ is represented by (32);
Step5:	Finally, the update rule of the mixing matrix is obtained by (34).

III. SIMULATION

In this section, a series of simulations are presented to investigate the performance of the proposed method. Two types of data, which include three sets of speech signals and four sets of biomedical signals, are used in the following experiments to verify the effectiveness of the algorithm. All these data were obtained from public databases [27].

A. Performance Index

The validity of the algorithms can be proven by the error indicator of the mixing matrix as follows:

$$\mathrm{E}_{\mathrm{A1}} = \frac{1}{n}\left\|A - \hat{A}\right\|^2 \quad (35)$$

$$\mathrm{E}_{\mathrm{A2}} = E\left\{\frac{\left\|A - \hat{A}\right\|^2}{\|A\|^2}\right\} \quad (36)$$

B. Epoch Experiments

In this simulation, we illustrate whether the performance of the method is affected by the time of observation. Here, the speech signal and biomedical signal are used to explain the performance. The two groups of data come from the public databases speech4_sin0_05 and Abio5. The total time period of the signal is 5000 s. The observation time points are selected as 200, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000 and 4500.

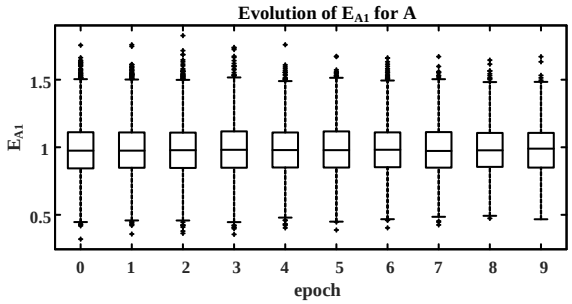

Fig. 2 The performance of E_{A1} for Speech4_sin0_05 at different time points

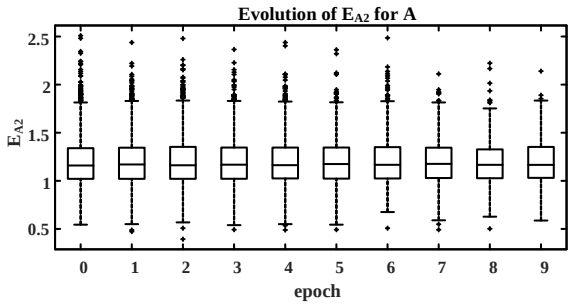

Fig. 3 The performance of E_{A2} for Speech4_sin0_05 at different time points

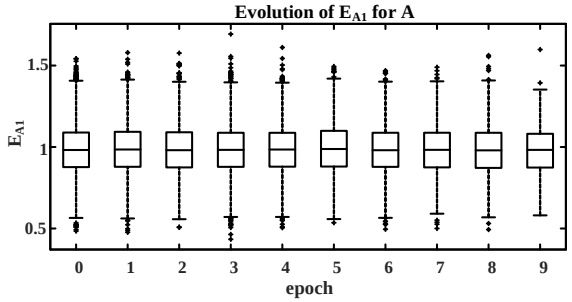

Fig. 4 The performance of E_{A1} for Abio5 at different time points

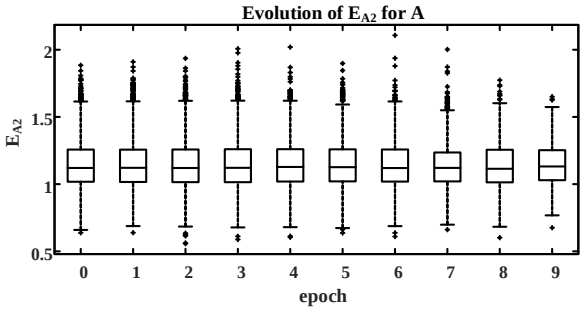

Fig. 5 The performance of E_{A2} for Abio5 at different time points

To better illustrate the statistical performance of the mixing matrix estimation, a box plot is used to show the estimation result in Figures 2-5 at different time points. Figures 2-5 clearly show that the estimated median values are basically on a horizontal line at different time points, which shows that the performance of mixing matrix estimation is independent of the choice of time points. In other words, regardless of how long the signal is observed, it will not affect the mixing matrix estimation. However, it also shows that the proposed method has better adaptability.

C. Underdetermined Experiments

In this simulation, we illustrate whether the performance of the method is affected by the number of observations. Here, the speech signal and biomedical signal are used to explain the performance.

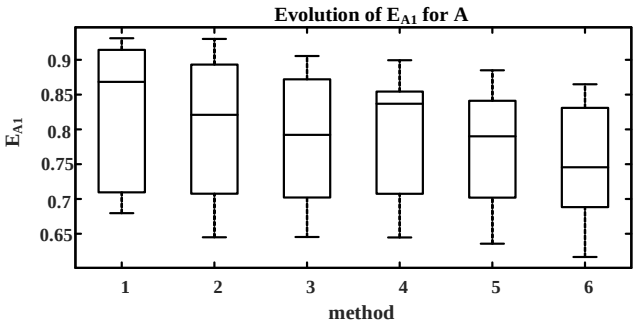

Fig. 6 The performance of E_{A1} for Speech20 at different numbers of observations

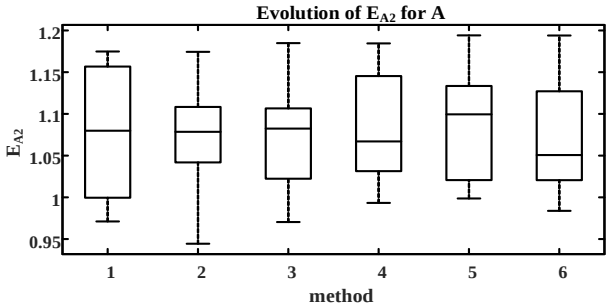

Fig. 7 The performance of E_{A2} for Speech20 at different numbers of observations

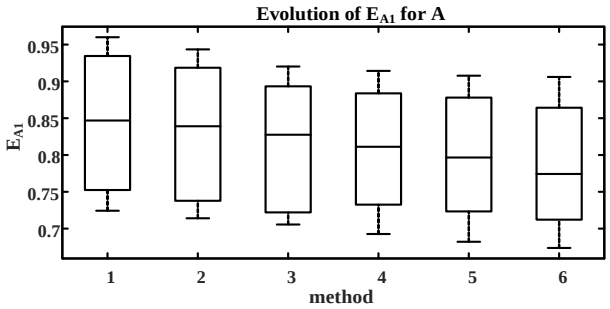

Fig. 8 The performance of E_{A1} for eeg23arti at different numbers of observations

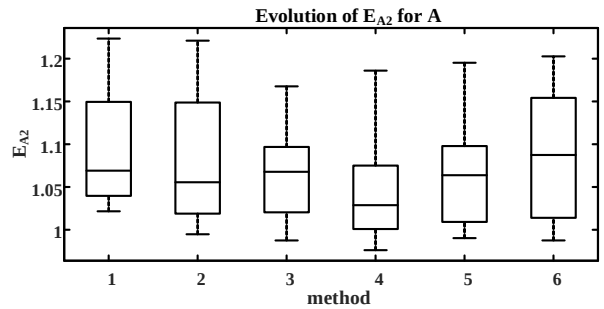

Fig. 9 The performance of E_{A2} for eeg23arti at different numbers of observations

TABLE II
PERFORMANCE OF DIFFERENT NUMBERS OF OBSERVATION

data	number	E_{A1}	E_{A2}
Speech20	18	0.8948	1.0258
	16	0.8687	1.0729
	14	0.7696	0.9970
	12	0.8442	1.1520
	10	0.7029	1.1148
	8	0.6527	1.1046
eeg23arti	18	0.9164	1.0977
	16	0.8971	1.0234
	14	0.8262	1.0349
	12	0.8039	1.0185
	10	0.7041	1.1080
	8	0.7291	1.1707

The two groups of data come from the public databases Speech20 and eeg23arti. Depending on the tensor decomposition theorem [17], the number of observations is set as 18, 16, 14, 12, 10, and 8. In addition, in order to better illustrate the statistical performance of the mixing matrix estimation, a box plot is used to show the estimation result in Figures 6-9 at different numbers of observations. The mean values of E_{A1} and E_{A2} are shown in Table 2.

From Figures 6-9, it is obvious that the median values of the mixing performance are similar regardless of the number of observed signals. These findings suggest that the method is fairly robust for the problem of underdetermined mixing matrix estimation.

D. Speech Signals

In this simulation, three groups of speech signals are adopted to illustrate the effectiveness of the algorithm. For Speech4, Speech8, Speech10 and Acspeech16, their observation numbers are set to 3, 5, 7 and 10, respectively, by the definition of tensor rank [19]. Their delay times are set to 10, 10, 18 and 30. The mixing matrix is a full column rank matrix and is randomly generated in MATLAB with time-varying points. The results are shown in Figures 10-17. To effectively illustrate the capability of the algorithm, the SOBIUM method [19], the underdetermined joint BSS (UJBSS)-1 method [23], the UJBSS-2 method [24] and UDSEP method [20] are used for comparison. Since all of these methods are offline methods, they are recalculated with each update of the mixing matrix with respect to time in order to meaningfully compare the offline and online methods in a fair manner.

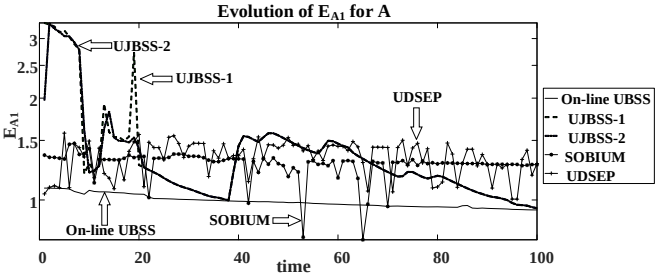

Fig. 10 The performance of E_{A1} for Speech4

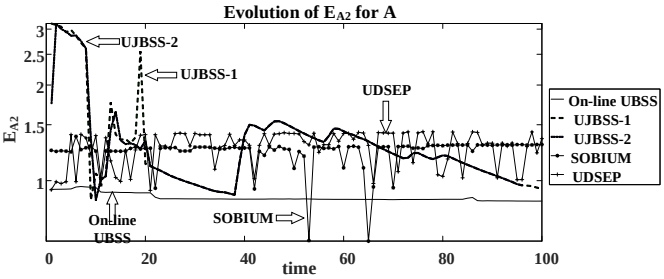

Fig. 11 The performance of E_{A2} for Speech4

Figures 10-11 show that for Speech4, the E_{A1} and E_{A2} values acquired by the UJBSS-1 method and UJBSS-2 method far outweigh those acquired by the other methods. Compared with the other methods, all methods focus on joint CP decomposition for estimating the mixing matrix by the ALS and ELS algorithms; therefore, their estimation performance is dependent on the initial conditions. We find that the E_{A1} and E_{A2} values of the UJBSS-1 and UBJSS-2 methods have large fluctuations with time. The E_{A1} and E_{A2} values of the online UBSS method are the minimum among these methods most of the time. It can be seen from the box plot of Figures 12-13 that the median value of the proposed method is smaller than that of other methods, and the distance between upper and lower limits is shorter, which illustrates that the proposed method achieves better performance results than the other methods in statistics.

Figure 14 denotes the time consumption of the algorithm. The CPU time of the UDSEP method is larger than that of the other methods because the parameter of the UDSEP method needs to be initialized by the SOBIUM method before estimating the mixing matrix. The CPU time of the online UBSS method is less than that of the other methods. This means that the online UBSS method has outstanding performance when estimating the time-varying mixing matrix for UBSS.

Due to space limitations, we only show the performance of the CPUs on Speech8, Speech10 and Acspeech16 in Figures 15-17, respectively.

For Speech8, Speech10 and Acspeech16, the online UBSS still has outstanding performance in time consumption. The mean values of E_{A1} and E_{A2} are shown in Table 3 These values obviously indicate that the online UBSS not only has outstanding performance in accurately estimating the mixing matrix but also enhances the efficiency of the estimation.

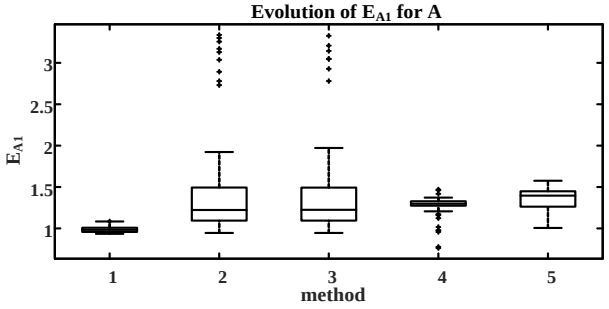

Fig. 12 Box plot of E_{A1} for Speech4

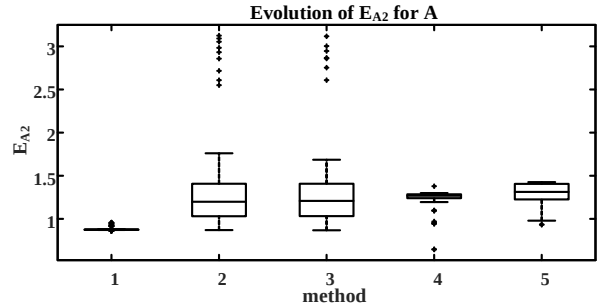

Fig. 13 Box plot of E_{A2} for Speech4

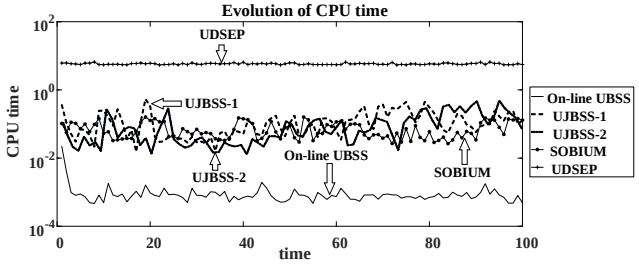

Fig. 14 The performance of CPU time for Speech4

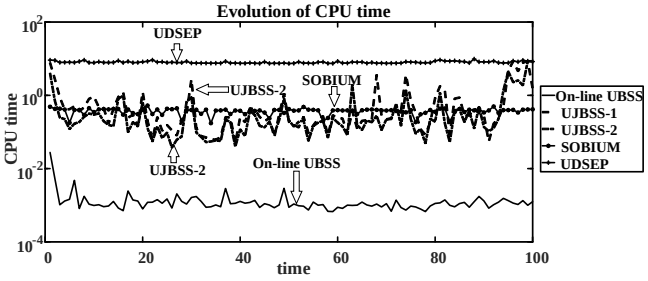

Fig. 15 The performance of CPU time for Speech8

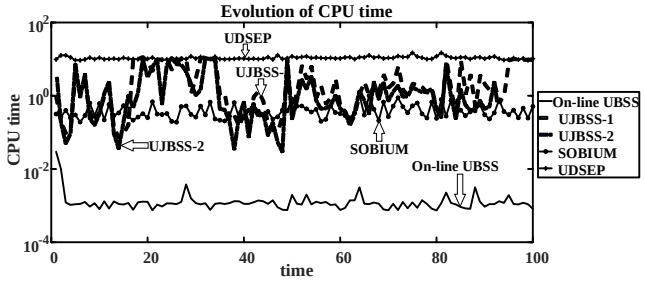

Fig. 16 The performance of CPU time for Speech10

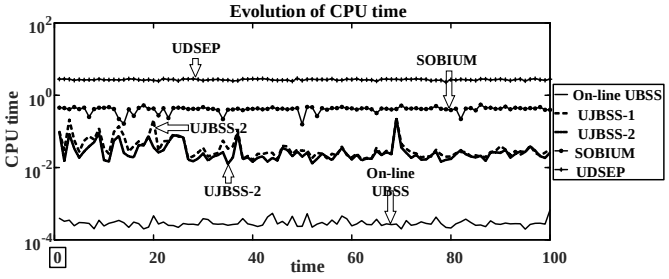

Fig. 17 The performance of CPU time for Acspeech16

TABLE III
PERFORMANCE OF DIFFERENT METHODS ON DIFFERENT SPEECH SIGNALS

data	method	E_{A1}	E_{A2}
Speech4	SOBIUM [1]	1.2783	1.2365
	UJBSS-1 [2]	1.4140	1.3480
	UJBSS-2 [3]	1.3909	1.3231
	UDSEP [4]	1.3368	1.2795
	Online UBSS	**0.9904**	**0.8851**
Speech8	SOBIUM	0.8360	1.1984
	UJBSS-1	3.1801	6.7301
	UJBSS-2	3.2008	6.7746
	UDSEP	0.8813	1.1722
	Online UBSS	**0.8089**	**1.1497**
Speech10	SOBIUM	0.9000	1.0552
	UJBSS-1	4.9026	8.1575
	UJBSS-2	4.7118	7.8323
	UDSEP	0.8853	1.0432
	Online UBSS	**0.8545**	**1.0370**
Acspeech16	SOBIUM	0.7346	1.2096
	UJBSS-1	0.9974	2.0667
	UJBSS-2	0.9836	2.0166
	UDSEP	0.7277	1.1037
	Online UBSS	**0.6943**	**1.0988**

E. Biomedical

Furthermore, public biomedical data are introduced to assess the performance of the online UBSS method. EEG4, Abio5, Abio6, Abio7 and eeg18raw contain 4, 5, 6, 7 and 18 typical biological sources, respectively, from [26]. The number of observed signals is set to 3, 4, 4, 5 and 15. The results are shown in Figures 18-23. Similarly, considering the length of the paper, we only show the CPU time results for Abio5, Abio6, Abio7 and eeg18raw in Figures 24-28.

Figures 18-19 show that for EEG4, the E_{A1} and E_{A2} values acquired by the UJBSS-1 method and UJBSS-2 method far outweigh those acquired by the other methods. At the same time, here, the E_{A1} and E_{A2} values of the UJBSS-1 and UBJSS-2 methods also have large fluctuations with time. The E_{A1} and E_{A2} values of the online UBSS method are the minimum among these methods most of the time. It can be seen from the box plots in Figures 20-23 that the median value of the proposed method is smaller than that of other methods, and the distance between the upper and lower limits is shorter, which illustrates that the proposed method achieves better performance results than the other methods in statistics for biomedical signals.

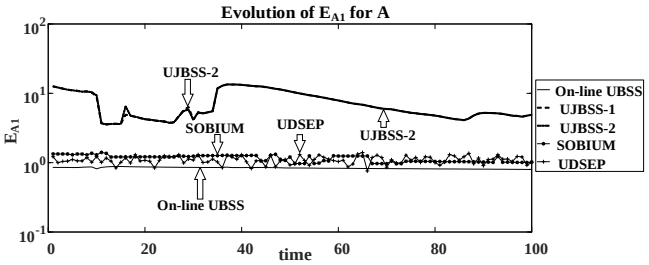

Fig. 18 The performance of E_{A1} for EEG4

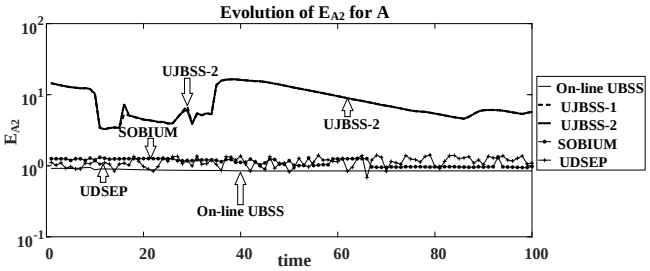

Fig. 19 The performance of E_{A2} for EEG4

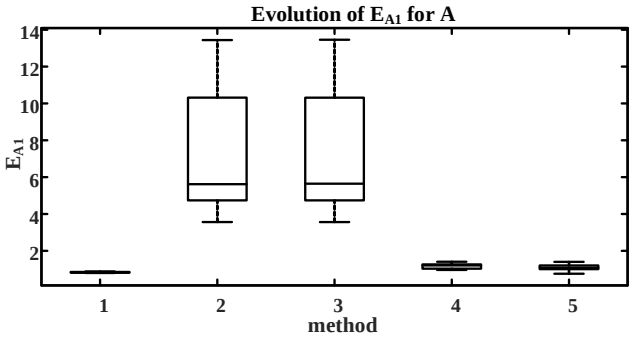

Fig. 20 Box plot of E_{A1} for EEG4

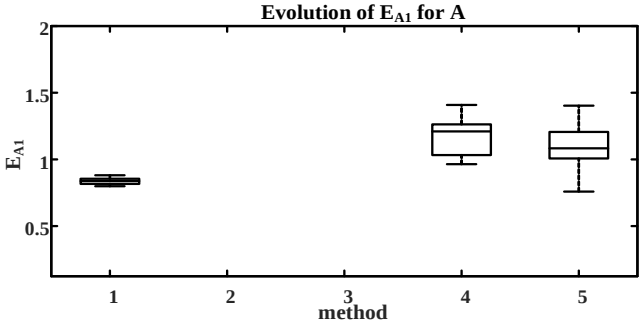

Fig. 21 Box plot of E_{A1} for EEG4 except for the UJBSS-1 and UJBSS-2 methods

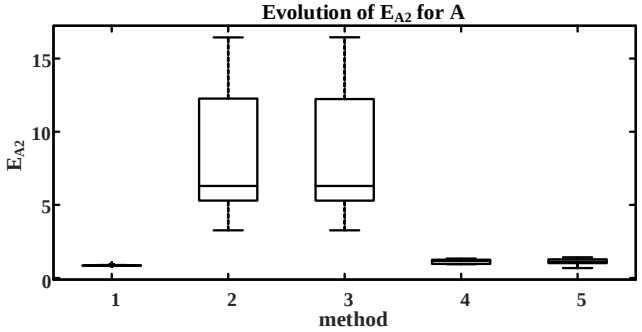

Fig. 22 Box plot of E_{A2} for EEG4

Figure 24 denotes the time consumption of the algorithm. The CPU time of the UDSEP method is larger than that of the other methods because the parameter of the UDSEP method needs to be initialized by the SOBIUM method before estimating the mixing matrix. The CPU time of the online UBSS method is less than that of the other methods. This means that the online UBSS method has outstanding performance when estimating the time-varying mixing matrix for UBSS.

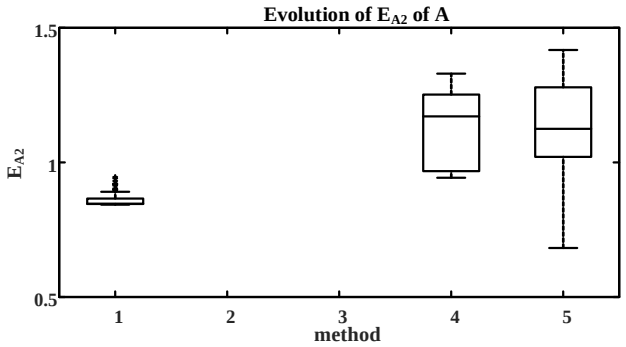

Fig. 23 Box plot of E_{A2} for EEG4 except for the UJBSS-1 and UJBSS-2 methods

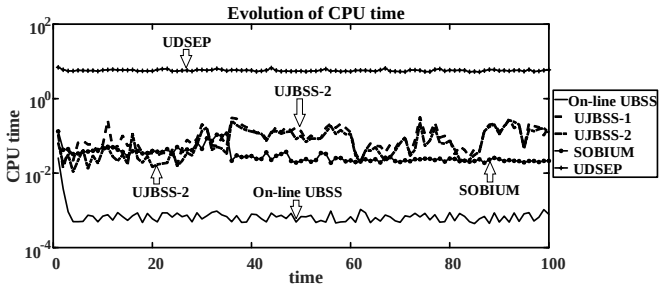

Fig. 24 The performance of CPU time on EEG4

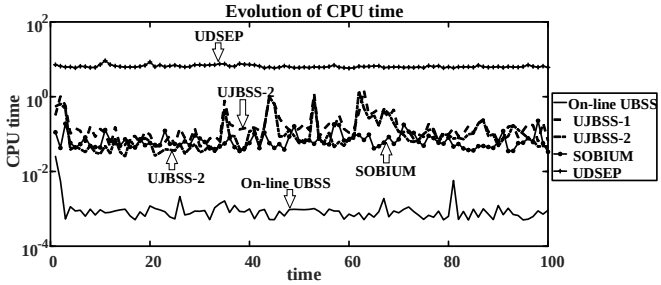

Fig. 25 The performance of CPU time on Abio5

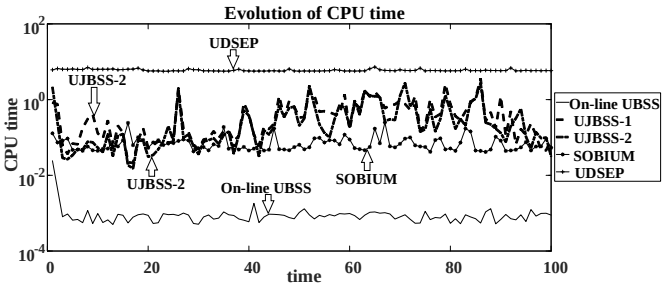

Fig. 26 The performance of CPU time on Abio6

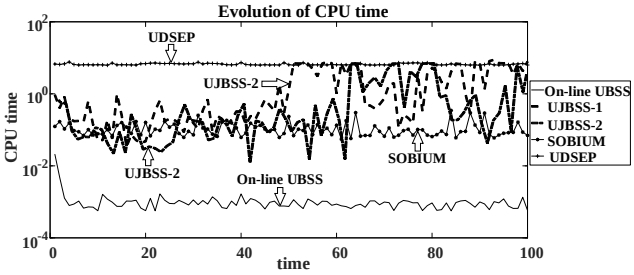

Fig. 27 The performance of CPU time on Abio7

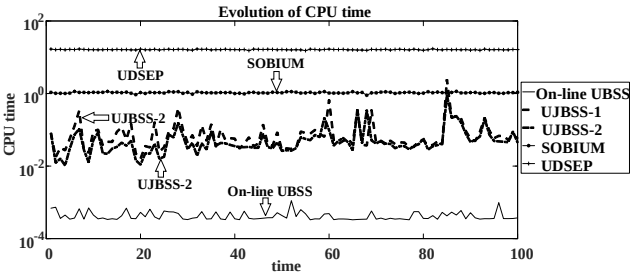

Fig. 28 The performance of CPU time on eeg18raw

TABLE IV
PERFORMANCE OF DIFFERENT METHODS FOR DIFFERENT BIOMEDICAL SIGNALS

data	method	E_{A1}	E_{A2}
EEG4	SOBIUM	1.1565	1.1198
	UJBSS-1	7.1995	8.3800
	UJBSS-2	7.2153	8.4004
	UDSEP	1.0965	1.1331
	Online UBSS	**0.8373**	**0.8609**
Abio5	SOBIUM	0.8350	1.4690
	UJBSS-1	1.2658	2.4672
	UJBSS-2	1.1722	2.2728
	UDSEP	0.7751	1.2327
	Online UBSS	**0.7284**	**1.0962**
Abio6	SOBIUM	0.8405	1.2842
	UJBSS-1	2.2107	4.2116
	UJBSS-2	2.2015	4.1917
	UDSEP	0.7823	1.2162
	Online UJBSS	**0.7645**	**1.0879**
Abio7	SOBIUM	0.8070	1.1583
	UJBSS-1	14.2695	26.5076
	UJBSS-2	14.0827	26.1309
	UDSEP	0.8270	1.1437
	Online UBSS	**0.7504**	**0.9901**
eeg18raw	SOBIUM	0.8965	1.0671
	UJBSS-1	4.4193	8.0484
	UJBSS-2	4.3421	7.8737
	UDSEP	0.8963	1.0485
	Online UJBSS	**0.8720**	**0.9965**

For biomedical data, the online UJBSS method also has outstanding time consumption performance compared to the other methods. The results of the five groups of data in terms of E_{A1} and E_{A2} are reported in Table 4. It is shown that the proposed online UJBSS method outperforms the other methods in all the performance metrics on the biomedical dataset.

IV. Conclusion

This article offers an online method to solve the problem of UBSS when the mixing matrix is time-varying. Considering the individual correlations of the source and time-varying mixing matrix, an online three-order tensor is established. Depending on adaptive tensor decomposition, we propose an online tensor algorithm to estimate the time-varying mixing matrix without any limitation on the properties of the sources.

We have implemented the proposed algorithm on two groups of data, namely, speech signals and biomedical signals. The performance of the proposed algorithms is compared with that of a few offline UBSS methods, especially in regard to time consumption. The simulation results substantiate the performance of our method.

References

[1] Behr M, Munk A, "Identifiability for blind source separation of multiple finite alphabet linear mixtures," *IEEE Transactions on Information Theory*, vol. 63, no. 9, pp5506-5517, 2017.

[2] Sadhu A, Narasimhan S, Antoni J, "A review of output-only structural mode identification literature employing blind source separation methods," *Mechanical Systems and Signal Processing*, vol. 94, pp415-431, 2017.

[3] Koldovský Z, Nesta F, "Performance analysis of source image estimators in blind source separation," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp4166-4176, 2017.

[4] He P J, She T T, Li W H and Yuan W B, "Single channel blind source separation on the instantaneous mixed signal of multiple dynamic sources," *Mechanical systems and signal processing*, vol. 113, pp22-35, 2018.

[5] Qian G B, Wei P, and Liao H S, "Efficient variant of noncircular complex FastICA algorithm for the blind source separation of digital communication signals," *Circuits, Systems, and Signal Processing*, vol. 35, no. 2, pp705-717, 2016.

[6] Ma Q R, Ye J M, "A Novel Derivation of Nc-FastICA and Convergence Analysis of C-FastICA." *IAENG International Journal of Computer Science*, vol. 46, no. 3, pp417-424, 2019.

[7] Su Q, Shen Y, Wei Y, and Deng C, "Underdetermined blind source separation by a novel time–frequency method," *AEU-International Journal of Electronics and Communications*, vol. 77, pp43-49, 2017.

[8] Jourjine A, Rickard S, and Yilmaz O, "Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures," *In Proc IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp2985-2988, 2000.

[9] Rickard, S, Balan, R, Rosca, J, "Real-time time-frequency based blind source separation," *In Proc IEEE International Conference on Independent Component Analysis*, pp651-656, 2001.

[10] Araki S, Sawada H, Mukai R, and Makino S, "Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors," *Signal Processing*, vol. 87, no. 8, pp1833-1847, 2007.

[11] Tseng C C, and Wu H H. "A monitoring system to assess supplier performance of a passive components company in Taiwan by K-Means method," *IAENG International Journal of Computer Science*, vol. 47, no. 2, pp172-180, 2020.

[12] Ryu Y, Park Y, Kim J, Lee S, "Image edge detection using fuzzy c-means and three directions image shift method," *IAENG International Journal of Computer Science*, vol. 45, no. 1, pp1-6, 2018.

[13] Dong T, Lei Y, Yang J, "An algorithm for underdetermined mixing matrix estimation, " *Neurocomputing*, vol. 104, pp26-34, 2013.

[14] Sun J, Li Y, Wen J, and Yan S. "Novel mixing matrix estimation approach in underdetermined blind source separation," *Neurocomputing*, vol. 173, pp623-632, 2016.

[15] Li Y, Nie W, Ye F, and Lin Y, "A mixing matrix estimation algorithm for underdetermined blind source separation," *Circuits, Systems, and Signal Processing*, vol. 35, no. 9, pp3367-3379, 2016.

[16] Comon P, "Blind identification and source separation in 2×3 underdetermined mixtures," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp11-22, 2004.

[17] Ferréol A, Albera L, Chevalier P, "Fourth-order blind identification of underdetermined mixtures of sources (FOBIUM)," *IEEE Transactions on Signal processing*, vol. 53, no. 5, pp1640-1653, 2005.

[18] De Lathauwer L, Castaing J, Cardoso J F, "Fourth-order cumulant-based blind identification of underdetermined mixtures," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp2965-2973, 2007.

[19] De Lathauwer L, Castaing J, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp1096-1105, 2008.

[20] Tichavsky P, Koldovsky Z, "Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp1037-1047, 2011.

[21] Phan A H, Cichocki A, "Extended HALS algorithm for nonnegative Tucker decomposition and its applications for multiway analysis and classification," *Neurocomputing*, vol. 74, no. 11, pp1956-1969, 2011.

[22] Cichocki A, Zdunek R, Phan A H, and Amari S, "Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation," *John Wiley & Sons*, 2009.

[23] Zou L, Wang Z J, Chen X, and Ji X, "Underdetermined joint blind source separation based on tensor decomposition," *In Proc IEEE International Conference on Electrical and Computer Engineering*, pp1-4, 2016.

[24] Zou L, Chen X, Wang Z J, "Underdetermined joint blind source separation for two datasets based on tensor decomposition," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp673-677, 2016

[25] Nion D, Sidiropoulos N D, "Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp2299-2310, 2009.

[26] Wang X Z, Liang L, and Che M L. "Iterative Algorithms for Computing the Takagi Factorization of Complex Symmetric Matrices." *IAENG International Journal of Applied Mathematics*, vol. 48, no.3, pp297-305, 2018.

[27] ICALAB, http://www.bsp.brain.riken.jp/ICALAB.

Sunan Ge received a Ph.D. from the School of Electronic Information and Electrical Engineering of Dalian University of Technology in 2016. She is a Lecturer in the School of Computer Science, Xi'an Polytechnic University, and a Postdoc in the School of Mathematics, Northwest University, China. Her current research interests include tensor decomposition, artifact removal of EEGs, and underdetermined blind source separation.

Xue Tao received a Ph.D. from the School of Electronic and Information Engineering of Xi'an Jiaotong University in 2005. He is a professor in the School of Computer Science, Xi'an Polytechnic University, China. His current research interests include distributed computation, cloud computing, big data and the internet of things.

Rui Zhang received a Ph.D. from the School of Electrical and Electronic Engineering of Nanyang Technological University and the School of Mathematics and Statistics of Xi'an Jiaotong University in 2012 and 2011, respectively. She is a professor in the School of Mathematics, Northwest University, China. Her current research interests include artificial neural networks, machine learning, and automatic seizure detection using EEGs.

Zhenzhong Hou received a Ph.D. in Materials Science from Tongji University in 2010. He is a supervisor of Master in the College of Materials Science and Engineering, Xi'an University of Science and Technology, Shaanxi, China. His current research interests include polymer hydrogel electrode materials and applications of artificial intelligence in material design.