IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 8, 2018, accepted February 13, 2018, date of publication February 27, 2018, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2808951

Fast Dual Purpose Algorithm Based on Novel Unified Cost Function

DONG HAI-DI, GAO YING-BIN, HE BING, AND LIU GANG
Xi'an Research Institute of High Technology, Xi'an 710025, China

Corresponding author: Gao Ying-Bin (545436624@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673387 and Grant 61374120 and in part by the Shaanxi Province Natural Science Foundation of China under Grant 2016JM6015.

ABSTRACT Principal subspace analysis (PSA) and minor subspace analysis (MSA) are considered two robust instruments in many fields. The dual purpose algorithm is capable of solving the PSA and MSA by simply switching the sign of one algorithm. Until today, there have been few dual purpose algorithms that are able to find their corresponding cost functions. In this paper, a novel unified cost function (NUCF) is proposed that possesses a global maximum, which is achieved only in the case where the weight matrix encompasses the desired principal or minor subspace. With the use of the gradient ascent method in the NUCF, we propose a novel dual purpose algorithm, which possesses lower computational complexity when compared with some existing algorithms. Numerical simulations and real applications illustrate that the proposed dual purpose algorithm is capable of tracking the desired subspace, and it converges faster than some similar types of algorithms.

INDEX TERMS Novel unified cost function, principal subspace analysis, minor subspace analysis, dual purpose algorithm.

I. INTRODUCTION

Principal subspace analysis (PSA) and minor subspace analysis (MSA) perform essential functions in numerous signal processing applications. For instance, PSA has been implemented in machine learning [1], face recognition .[2] and direction of arrival (DOA) estimation [3]. Likewise, MSA has been implemented in array signal processing [4], radio frequency interference (RFI) mitigation [5], total least squares (TLS) [6], and parameter estimation [7]. Recently, the neural network based PSA and MSA algorithms have garnered significant attention. When compared to conventional algebraic methodologies such as singular value decomposition (SVD) and eigenvalue decomposition (EVD), the benefits offered by neural network algorithms are that they are capable of estimating the covariance matrix from sample signals and tracking non-stationary distributions [8]. Large numbers of algorithms have been reported with respect to PSA [9]–[11] and MSA [12]–[15].

Dual purpose algorithms have capabilities for PSA and MSA through simply switching the sign in the same algorithm [16]. Thus, they have emerged as one of the hot research topics in this field. When compared to other PSA or MSA algorithms, dual purpose algorithms offer numerous benefits, including lower computation complexity and lower hardware costs [17]. Tianping [16] proposed the first dual purpose learning rule using neural networks. Following this innovative work, numerous dual purpose algorithms have been proposed with respect to both PSA and MSA, such as Hasan algorithm [18], Peng algorithm [19] and the stable data projection method (SDPM) algorithm [7]. Nevertheless, every algorithm stated above is proposed on the basis of heuristic reasoning. Furthermore, it is difficult to assess their matching cost functions. In [20], a unified information criterion (UIC) was proposed and a dual purpose algorithm (UIC algorithm) was developed. However, simulation experiments discovered that the UIC algorithm is limited in fast tracking. Since the convergence speed is an important property and the cost function can affect the convergence speed of neural network algorithms [21], it is valuable to develop new cost functions for dual purpose algorithms. In this paper, through the addition of a penalty term to the Rayleigh Quotient, we propose a novel unified cost function (NUCF) and analyze its landscape.

The main contributions of this paper are 1) A novel unified cost function is proposed for PSA and MSA. 2) The landscape of NUCF is analyzed using the stable point method, which provides a new way to analyze cost functions. 3) A dual purpose algorithm is derived, which has fast convergence speed and low computation complexity.

The rest of this paper is organized as follows. In Section II, we propose a novel cost function and analyze its landscape with the use of the stationary point method. In Section III, a dual purpose algorithm is proposed based on the proposed cost function and we also compare it with some prevalent existing algorithms. Section IV presents a numerical simulation and two application experiments. We conclude this research in Section V.

II. NOVEL COST FUNCTION AND ITS LANDSCAPE

A. NEURAL NETWORK MODEL AND NOVEL COST FUNCTION

Consider the following linear neural network framework

$$\boldsymbol{y}(k) = \boldsymbol{W}^T(k)\boldsymbol{x}(k), \quad k = 0, 1, \cdots \tag{1}$$

where $\boldsymbol{W}(k) \in \mathbb{R}^{n\times r}$ indicates the weight matrix of this neuron. $\boldsymbol{x}(k) \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{y}(k) \in \mathbb{R}^{r\times 1}$ are the input and output of this model, respectively. The key aim of dual purpose algorithms is the construction of an appropriate learning rule with respect to the weight matrix for the purpose of tracking the principal subspace (PS) or the minor subspace (MS) of input signals [9].

Inspired by the Rayleigh Quotient, we propose following dual purpose cost function:

$$\begin{aligned} \boldsymbol{W}^* &= \arg\max_{\boldsymbol{W}\in\Omega} J_{NUCF}(\boldsymbol{W}) \\ J_{NUCF}(\boldsymbol{W}) &= \pm\frac{1}{2}tr[(\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W})(\boldsymbol{W}^T\boldsymbol{W})^{-1}] \\ &\quad + \frac{1}{2}tr[\ln(\boldsymbol{W}^T\boldsymbol{W}) - \boldsymbol{W}^T\boldsymbol{W}] \end{aligned} \tag{2}$$

where $\boldsymbol{R} = E[\boldsymbol{x}(k)\boldsymbol{x}^T(k)]$ is the autocorrelation matrix of the input signal. Clearly, $\boldsymbol{R}$ is a symmetric positive definite matrix. When "+" is used, (2) indicates a PSA cost function. When "−" is utilized, (2) suggests an MSA cost function. Clearly, (2) is an unconstraint optimization function. With respect to this function, we pay attention to the following three questions.

a) Does the maximum of this function exist?
b) What is the relationship between the desired subspace and the maximum?
c) Is there any other local maximum?

The answers to the above questions will be provided in the subsequent subsections. These answers merely accomplish the landscape analysis of this function.

B. LANDSCAPE OF NUCF

Suppose that the eigenvalues and eigenvectors of $\boldsymbol{R}$ are stated as λ_i and $\boldsymbol{u}_i$ $(i = 1, 2, \cdots, n)$, respectively. Then arrange the eigenvectors in two different manners.

In the first manner, the arrangement of the orthonormal eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n$ is performed in a way that their respective eigenvalues have a non-ascending sequence such as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Subject to this scenario, $\boldsymbol{R}$ can be decomposed as:

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = \boldsymbol{U}_1\boldsymbol{\Lambda}_1\boldsymbol{U}_1^T + \boldsymbol{U}_2\boldsymbol{\Lambda}_2\boldsymbol{U}_2^T \tag{3}$$

where $\boldsymbol{U} = [\boldsymbol{U}_1, \boldsymbol{U}_2]$ and

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{bmatrix} \tag{4}$$

In (3) and (4), $\boldsymbol{U}_1 = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r]$ is constituted by the first r eigenvectors and $\boldsymbol{U}_2 = [\boldsymbol{u}_{r+1}, \boldsymbol{u}_{r+2}, \cdots, \boldsymbol{u}_n]$ is constituted by the remaining $n - r$ eigenvectors. $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ represent two diagonal matrices whose diagonal elements are the respective matching eigenvalues. Therefore, (3) is merely the EVD of $\boldsymbol{R}$.

In the second manner, the arrangement of the eigenvalues is carried out in an arbitrary sequence, but not the non-ascending sequence. Subject to the second scenario, the decomposition of $\boldsymbol{R}$ can be implemented in the following formula

$$\boldsymbol{R} = \boldsymbol{U}'\boldsymbol{\Lambda}'(\boldsymbol{U}')^T = \boldsymbol{U}_r\boldsymbol{\Lambda}_r\boldsymbol{U}_r^T + \boldsymbol{U}_{n-r}\boldsymbol{\Lambda}_{n-r}\boldsymbol{U}_{n-r}^T \tag{5}$$

where $\boldsymbol{U}' = [\boldsymbol{U}_r, \boldsymbol{U}_{n-r}]$ and

$$\boldsymbol{\Lambda}' = \begin{bmatrix} \boldsymbol{\Lambda}_r & \\ & \boldsymbol{\Lambda}_{n-r} \end{bmatrix} \tag{6}$$

In (5) and (6), $\boldsymbol{U}_r$ houses r arbitrary eigenvectors and $\boldsymbol{U}_{n-r}$ possesses the residual $n-r$ eigenvectors. $\boldsymbol{\Lambda}_r$ and $\boldsymbol{\Lambda}_{n-r}$ indicate the two diagonal matrices. Moreover, their diagonal elements constitute their respective eigenvalues.

Given $\boldsymbol{W}$ in domain $\Omega = \{\boldsymbol{W}|0 < \boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W} < \infty, \boldsymbol{W}^T\boldsymbol{W} \neq 0\}$, we assessed the following PSA cost function

$$\begin{aligned} \boldsymbol{W}^* &= \arg\max_{\boldsymbol{W}\in\Omega} E_1(\boldsymbol{W}) \\ E_1(\boldsymbol{W}) &= \frac{1}{2}tr[(\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W})(\boldsymbol{W}^T\boldsymbol{W})^{-1}] \\ &\quad + \frac{1}{2}tr[\ln(\boldsymbol{W}^T\boldsymbol{W}) - \boldsymbol{W}^T\boldsymbol{W}] \end{aligned} \tag{7}$$

Theorem 1: $\boldsymbol{W}$ indicates a stationary point of $E_1(\boldsymbol{W})$ in Ω if $\boldsymbol{W} = \boldsymbol{U}_r\boldsymbol{Q}$, where $\boldsymbol{Q}$ is an arbitrary orthogonal matrix.

Proof: From (7), we observe that in the region Ω, $E_1(\boldsymbol{W})$ is a differentiable function since $\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}$ and $\boldsymbol{W}^T\boldsymbol{W}$ are two positive definite matrices. Through the matrix differential method, we can conveniently attain the gradient of $E_1(\boldsymbol{W})$ with respect to $\boldsymbol{W}$, which is provided by

$$\begin{aligned} \nabla E_1(\boldsymbol{W}) &= \left[\boldsymbol{R}\boldsymbol{W} - \boldsymbol{W}\{\boldsymbol{W}^T\boldsymbol{W}\}^{-1}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\right] \\ &\quad \times \{\boldsymbol{W}^T\boldsymbol{W}\}^{-1} + \left[\boldsymbol{W}\{\boldsymbol{W}^T\boldsymbol{W}\}^{-1} - \boldsymbol{W}\right] \end{aligned} \tag{8}$$

If $\boldsymbol{W} = \boldsymbol{U}_r\boldsymbol{Q}$, we then have

$$\begin{aligned} &\nabla E_1(\boldsymbol{W})|_{\boldsymbol{W}=\boldsymbol{U}_r\boldsymbol{Q}} \\ &= \left[\boldsymbol{R}\boldsymbol{U}_r\boldsymbol{Q} - \boldsymbol{U}_r\boldsymbol{Q}\left\{\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{U}_r\boldsymbol{Q}\right\}^{-1}\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{R}\boldsymbol{U}_r\boldsymbol{Q}\right] \\ &\quad \times \left\{\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{U}_r\boldsymbol{Q}\right\}^{-1} + \left[\boldsymbol{U}_r\boldsymbol{Q}\left\{\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{U}_r\boldsymbol{Q}\right\}^{-1} - \boldsymbol{U}_r\boldsymbol{Q}\right] \\ &= \left[\boldsymbol{R}\boldsymbol{U}_r\boldsymbol{Q} - \boldsymbol{U}_r\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{R}\boldsymbol{U}_r\boldsymbol{Q}\right] + [\boldsymbol{U}_r\boldsymbol{Q} - \boldsymbol{U}_r\boldsymbol{Q}] \\ &= \boldsymbol{0} \end{aligned} \tag{9}$$

Moreover, through the use of $\nabla E_1(\boldsymbol{W}) = 0$, if $\boldsymbol{W}$ indicates a stationary point of $E_1(\boldsymbol{W})$, we have

$$\left[\boldsymbol{R}\boldsymbol{W} - \boldsymbol{W}\left\{\boldsymbol{W}^T\boldsymbol{W}\right\}^{-1}\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}\right]\left\{\boldsymbol{W}^T\boldsymbol{W}\right\}^{-1} = -\left[\boldsymbol{W}\left\{\boldsymbol{W}^T\boldsymbol{W}\right\}^{-1} - \boldsymbol{W}\right] \quad (10)$$

Multiplying both sides of (10) by $\boldsymbol{W}^T$, we can attain that

$$\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_r \quad (11)$$

As suggested by (11), we can observe that the columns of $\boldsymbol{W}$ are orthonormal at the stationary point of $E_1(\boldsymbol{W})$. Suppose that the EVD of $\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}$ can be written as

$$\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W} = \left(\boldsymbol{Q}''\right)^T\boldsymbol{\Lambda}_r''\boldsymbol{Q}'' \quad (12)$$

where $\boldsymbol{Q}''$ is an arbitrary orthogonal matrix. By substituting (12) into (10) and simplifying the statement, we have

$$\boldsymbol{R}\boldsymbol{U}_r'' = \boldsymbol{U}_r''\boldsymbol{\Lambda}_r'' \quad (13)$$

where $\boldsymbol{U}_r'' = \boldsymbol{W}\boldsymbol{Q}''^T$ and it satisfies $(\boldsymbol{U}_r'')^T\boldsymbol{U}_r'' = \boldsymbol{I}_r$. Since $\boldsymbol{\Lambda}_r''$ indicates a diagonal matrix and $\boldsymbol{U}_r''$ denotes a column full rank matrix, we can conclude that $\boldsymbol{U}_r$ and $\boldsymbol{\Lambda}_r$ are respectively equivalent to $\boldsymbol{U}_r''$ and $\boldsymbol{\Lambda}_r''$. In other words, at the stationary point of $E_1(\boldsymbol{W})$, we have $\boldsymbol{W} = \boldsymbol{U}_r\boldsymbol{Q}$.

This accomplishes the proof.

Theorem 2: In Ω, $E_1(\boldsymbol{W})$ possesses just one global maximum, which is attained if and only if $\boldsymbol{W} = \boldsymbol{U}_1\boldsymbol{Q}$, where $\boldsymbol{U}_1 = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r]$. Every other stationary point is merely a saddle point of $E_1(\boldsymbol{W})$. Moreover, at this global maximum, we have

$$E_1(\boldsymbol{W}) = \frac{1}{2}\sum_{i=1}^{r}\lambda_i - \frac{r}{2} \quad (14)$$

Proof: From Theorem 1, we can observe that any $\boldsymbol{W} = \boldsymbol{U}_r\boldsymbol{Q}$ constitutes a stationary point of $E_1(\boldsymbol{W})$. Thus, we denote $\boldsymbol{U}_r = [\boldsymbol{u}_{i_1}, \boldsymbol{u}_{i_2}, \cdots, \boldsymbol{u}_{i_r}]$, where $i_1, i_2, \cdots, i_r$ indicates the indexes of these eigenvectors. Furthermore, the set, which comprises these indexes, is denoted as $L_1 = \{i_1, i_2, \cdots, i_r\}$. In the same manner, the indexes of the vectors in $\boldsymbol{U}_1 = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r]$ constitute another set, which is denoted as $L_2 = \{1, 2, \cdots, r\}$.

In respect of any $L_1 (L_1 \neq L_2)$, there must exist a j that caters to $j \in L_1$ and $j \notin L_2$. Thereafter, we replace the component $\boldsymbol{u}_j$ of the matrix $\boldsymbol{U}_r$ using $\boldsymbol{u}_j + \varepsilon\boldsymbol{u}_k$, where $k \in L_2, k \notin L_1$ and $\forall\varepsilon > 0$. Suppose that $\tilde{\boldsymbol{U}}_r$ indicates the resulting new matrix, where $\tilde{\boldsymbol{W}} = \tilde{\boldsymbol{U}}_r\boldsymbol{Q}$. Then we have

$$\tilde{\boldsymbol{U}}_r = \boldsymbol{U}_r + \varepsilon\boldsymbol{M} \quad (15)$$

where $\boldsymbol{M} = [\boldsymbol{0}, \cdots, \boldsymbol{u}_k, \cdots, \boldsymbol{0}]$ is an $n \times r$ matrix. The jth column of $\boldsymbol{M}$ is $\boldsymbol{u}_k$ and the other columns are null vectors.

Substituting (15) into (7), we have

$$\begin{aligned}
E_1(\boldsymbol{W})|_{\boldsymbol{W}=\tilde{\boldsymbol{U}}_r\boldsymbol{Q}} &= \frac{1}{2}tr\left[\left(\boldsymbol{Q}^T\tilde{\boldsymbol{U}}_r^T\boldsymbol{R}\tilde{\boldsymbol{U}}_r\boldsymbol{Q}\right)\left(\boldsymbol{Q}^T\tilde{\boldsymbol{U}}_r^T\tilde{\boldsymbol{U}}_r\boldsymbol{Q}\right)^{-1}\right] \\
&\quad + \frac{1}{2}tr\left[\ln(\boldsymbol{Q}^T\tilde{\boldsymbol{U}}_r^T\tilde{\boldsymbol{U}}_r\boldsymbol{Q}) - \left(\boldsymbol{Q}^T\tilde{\boldsymbol{U}}_r^T\tilde{\boldsymbol{U}}_r\boldsymbol{Q}\right)\right] \\
&= \frac{1}{2}tr\left[\left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})^T\boldsymbol{R}(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})\boldsymbol{Q}\right)\right. \\
&\quad \left.\times\left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})\boldsymbol{Q}\right)^{-1}\right] \\
&\quad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})\boldsymbol{Q}\right)\right. \\
&\quad \left.- \left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})\boldsymbol{Q}\right)\right] \\
&= \frac{1}{2}tr\left[\left(\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q} + \varepsilon^2\boldsymbol{Q}^T\boldsymbol{M}^T\boldsymbol{R}\boldsymbol{M}\boldsymbol{Q}\right)\right. \\
&\quad \left.\times\left(\boldsymbol{Q}^T(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{Q}\right)^{-1}\right] \\
&\quad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{Q}^T(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{Q}\right)\right. \\
&\quad \left.- \left(\boldsymbol{Q}^T(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{Q}\right)\right] \quad (16)
\end{aligned}$$

By using $\boldsymbol{M}^T\boldsymbol{M} = diag\,(0, 0, \cdots, 1, \cdots 0)$, we have

$$\left(\boldsymbol{Q}^T(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{Q}\right)^{-1} = \boldsymbol{I} + \boldsymbol{Q}^T\boldsymbol{C}_1\boldsymbol{Q} \quad (17)$$

where $\boldsymbol{C}_1 = diag(0, 0, \cdots, 1/(1+\varepsilon^2) - 1, \cdots 0)$.

Denote $\boldsymbol{C}_2 = \boldsymbol{M}^T\boldsymbol{R}\boldsymbol{M} = diag(0, \cdots, \lambda_k, \cdots 0)$ and substitute it into (7). Subsequently, we have

$$\begin{aligned}
E_1(\boldsymbol{W})|_{\boldsymbol{W}=\tilde{\boldsymbol{U}}_r\boldsymbol{Q}} &= \frac{1}{2}tr\left[\left(\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q} + \varepsilon^2\boldsymbol{Q}^T\boldsymbol{C}_2\boldsymbol{Q}\right)\left(\boldsymbol{I} + \boldsymbol{Q}^T\boldsymbol{C}_1\boldsymbol{Q}\right)\right] \\
&\quad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})\boldsymbol{Q}\right)\right. \\
&\quad \left.- \left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{M})\boldsymbol{Q}\right)\right] \\
&= \frac{1}{2}tr\left[\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q} + \varepsilon^2\boldsymbol{Q}^T\boldsymbol{C}_2\boldsymbol{Q}\right. \\
&\quad \left.+\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{C}_1\boldsymbol{Q} + \varepsilon^2\boldsymbol{Q}^T\boldsymbol{C}_2\boldsymbol{C}_1\boldsymbol{Q}\right] \\
&\quad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{Q}^T(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{Q}\right)\right. \\
&\quad \left.- \left(\boldsymbol{Q}^T(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{Q}\right)\right] \quad (18)
\end{aligned}$$

In the same way, we obtain

$$\begin{aligned}
E_1(\boldsymbol{W})|_{\boldsymbol{W}=\boldsymbol{U}_r\boldsymbol{Q}} &= \frac{1}{2}tr\left[\left(\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{R}\boldsymbol{U}_r\boldsymbol{Q}\right)\left(\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{U}_r\boldsymbol{Q}\right)^{-1}\right] \\
&\quad + \frac{1}{2}tr\left[\ln(\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{U}_r\boldsymbol{Q}) - \left(\boldsymbol{Q}^T\boldsymbol{U}_r^T\boldsymbol{U}_r\boldsymbol{Q}\right)\right] \\
&= \frac{1}{2}tr\left[\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q}\right] + \frac{1}{2}tr\left[\ln(\boldsymbol{I}) - \boldsymbol{I}\right] \quad (19)
\end{aligned}$$

Through the use of (16) and (19), we obtain

$$\begin{aligned}
&E_1(\boldsymbol{W})|_{\boldsymbol{W}=\tilde{\boldsymbol{U}}_r\boldsymbol{Q}} - E_1(\boldsymbol{W})|_{\boldsymbol{W}=\boldsymbol{U}_r\boldsymbol{Q}} \\
&= \frac{1}{2}tr\left[\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q} + \varepsilon^2\boldsymbol{Q}^T\boldsymbol{C}_2\boldsymbol{Q} + \boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{C}_1\boldsymbol{Q}\right. \\
&\quad \left.+\varepsilon^2\boldsymbol{Q}^T\boldsymbol{C}_2\boldsymbol{C}_1\boldsymbol{Q}\right] - \frac{1}{2}tr\left[\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q}\right] - \frac{1}{2}tr\left[\ln(\boldsymbol{I}) - \boldsymbol{I}\right] \\
&\quad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{Q}^T(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{Q}\right)\right. \\
&\quad \left.- \left(\boldsymbol{Q}^T(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M})\boldsymbol{Q}\right)\right] \\
&= \frac{1}{2}tr\left[\varepsilon^2\boldsymbol{C}_2 + \boldsymbol{\Lambda}_r\boldsymbol{C}_1 + \varepsilon^2\boldsymbol{C}_2\boldsymbol{C}_1\right] \\
&\quad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{I} + \varepsilon^2\boldsymbol{M}^T\boldsymbol{M}\right) - \varepsilon^2\boldsymbol{M}^T\boldsymbol{M}\right] \\
&= \frac{1}{2}\left[\varepsilon^2\lambda_k - \frac{\varepsilon^2}{1+\varepsilon^2}\lambda_j - \frac{\varepsilon^4}{1+\varepsilon^2}\lambda_k + \ln\left(1+\varepsilon^2\right) - \varepsilon^2\right] \\
&= \frac{1}{2}\left(\lambda_k - \lambda_j\right)\varepsilon^2 + o(\varepsilon^2) \qquad (20)
\end{aligned}$$

It is easily observed that, along the direction of component $\boldsymbol{u}_k$, $E_1(\boldsymbol{W})$ is expected to rise. $\hat{\boldsymbol{U}}_r$ indicates the resulting new matrix that is obtained through the replacement of the jth column of $\boldsymbol{U}_r$ with $\boldsymbol{u}_j + \varepsilon\boldsymbol{u}_j$ such that

$$\hat{\boldsymbol{U}}_r = \boldsymbol{U}_r + \varepsilon\boldsymbol{P} \qquad (21)$$

where $\boldsymbol{P} = [\boldsymbol{0}, \cdots, \boldsymbol{u}_j, \cdots, \boldsymbol{0}]$ indicates an $n \times r$ matrix. The jth column of $\boldsymbol{P}$ is $\boldsymbol{u}_j$ and the other columns are all null vectors.

If $\boldsymbol{W} = \hat{\boldsymbol{U}}_r\boldsymbol{Q}$, by substituting (21) into (7), we obtain

$$\begin{aligned}
&E_1(\boldsymbol{W})|_{\boldsymbol{W}=\hat{\boldsymbol{U}}_r\boldsymbol{Q}} \\
&\quad = \frac{1}{2}tr\left[\left(\boldsymbol{Q}^T\hat{\boldsymbol{U}}_r^T\boldsymbol{R}\hat{\boldsymbol{U}}_r\boldsymbol{Q}\right)\left(\boldsymbol{Q}^T\hat{\boldsymbol{U}}_r^T\hat{\boldsymbol{U}}_r\boldsymbol{Q}\right)^{-1}\right] \\
&\qquad + \frac{1}{2}tr\left[\ln(\boldsymbol{Q}^T\hat{\boldsymbol{U}}_r^T\hat{\boldsymbol{U}}_r\boldsymbol{Q}) - \left(\boldsymbol{Q}^T\hat{\boldsymbol{U}}_r^T\hat{\boldsymbol{U}}_r\boldsymbol{Q}\right)\right] \\
&\quad = \frac{1}{2}tr\left[\left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{P})^T\boldsymbol{R}(\boldsymbol{U}_r + \varepsilon\boldsymbol{P})\boldsymbol{Q}\right)\right. \\
&\qquad \left.\times\left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{P})^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{P})\boldsymbol{Q}\right)^{-1}\right] \\
&\qquad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{P})^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{P})\boldsymbol{Q}\right)\right. \\
&\qquad \left.- \left(\boldsymbol{Q}^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{P})^T(\boldsymbol{U}_r + \varepsilon\boldsymbol{P})\boldsymbol{Q}\right)\right] \\
&\quad = \frac{1}{2}tr\left[\left(\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q} + 2\varepsilon\boldsymbol{Q}^T\boldsymbol{C}_2''\boldsymbol{Q} + \varepsilon^2\boldsymbol{Q}^T\boldsymbol{C}_2''\boldsymbol{Q}\right)\right. \\
&\qquad \left.\times\left(\boldsymbol{Q}^T(\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}_1')\boldsymbol{Q}\right)^{-1}\right] \\
&\qquad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{Q}^T(\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}_1')\boldsymbol{Q}\right)\right. \\
&\qquad \left.- \left(\boldsymbol{Q}^T(\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}_1')\boldsymbol{Q}\right)\right] \qquad (22)
\end{aligned}$$

where $\boldsymbol{C}_2'' = diag(0, \cdots, \lambda_j, \cdots 0)$ and $\boldsymbol{C}_1' = diag(0, \cdots, 1, \cdots 0)$. By using $\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}' = diag(1, \cdots, 1 + 2\varepsilon + \varepsilon^2, \cdots, 1)$, we can obtain

$$\left(\boldsymbol{Q}^T(\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}_1')\boldsymbol{Q}\right)^{-1} = \boldsymbol{I} + \boldsymbol{Q}^T\boldsymbol{C}_1''\boldsymbol{Q} \qquad (23)$$

where $\boldsymbol{C}_1'' = diag(0, \cdots, 1/(1 + 2\varepsilon + \varepsilon^2) - 1, \cdots 0)$. Substituting (23) into (7), we obtain

$$\begin{aligned}
E_1(\boldsymbol{W})|_{\boldsymbol{W}=\hat{\boldsymbol{U}}_r\boldsymbol{Q}} &= \frac{1}{2}tr\left[\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q} + 2\varepsilon\boldsymbol{C}_2'' + \varepsilon^2\boldsymbol{C}_2''\right. \\
&\quad \left.+\boldsymbol{\Lambda}_r\boldsymbol{C}_1'' + 2\varepsilon\boldsymbol{C}_2''\boldsymbol{C}_1'' + \varepsilon^2\boldsymbol{C}_2''\boldsymbol{C}_1''\right] \\
&\quad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}_1'\right)\right. \\
&\quad \left.- \left(\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}_1'\right)\right] \qquad (24)
\end{aligned}$$

From (22) and (24), we obtain

$$\begin{aligned}
&E_1(\boldsymbol{W})|_{\boldsymbol{W}=\hat{\boldsymbol{U}}_r\boldsymbol{Q}} - E_1(\boldsymbol{W})|_{\boldsymbol{W}=\boldsymbol{U}_r\boldsymbol{Q}} \\
&\quad = \frac{1}{2}tr\left[\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q} + 2\varepsilon\boldsymbol{C}_2'' + \varepsilon^2\boldsymbol{C}_2''\right. \\
&\qquad \left.+\boldsymbol{\Lambda}_r\boldsymbol{C}_1'' + 2\varepsilon\boldsymbol{C}_2''\boldsymbol{C}_1'' + \varepsilon^2\boldsymbol{C}_2''\boldsymbol{C}_1''\right] - \frac{1}{2}tr\left[\boldsymbol{Q}^T\boldsymbol{\Lambda}_r\boldsymbol{Q}\right] \\
&\qquad + \frac{1}{2}tr\left[\ln\left(\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}_1'\right)\right. \\
&\qquad \left.- \left(\boldsymbol{I} + 2\varepsilon\boldsymbol{C}_1' + \varepsilon^2\boldsymbol{C}_1'\right)\right] - \frac{1}{2}tr\left[\ln(\boldsymbol{I}) - \boldsymbol{I}\right] \\
&\quad = 2\varepsilon\lambda_j + \varepsilon^2\lambda_j + \left(\frac{1}{1+2\varepsilon+\varepsilon^2} - 1\right)\lambda_j \\
&\qquad + 2\varepsilon\left(\frac{1}{1+2\varepsilon+\varepsilon^2} - 1\right)\lambda_j + \ln(1+\varepsilon)^2 \\
&\qquad + \varepsilon^2\left(\frac{1}{1+2\varepsilon+\varepsilon^2} - 1\right)\lambda_j + 2\varepsilon + \varepsilon^2 \\
&\quad = -2\varepsilon^2 + o(\varepsilon^2) \qquad (25)
\end{aligned}$$

From (25), we observe that $E_1(\boldsymbol{W})$ is expected to decrease along the direction of $\boldsymbol{u}_j$. It implies that the stationary point $\boldsymbol{W} = \boldsymbol{U}_r'\boldsymbol{Q}$ is not stable. Since $E_1(\boldsymbol{W})$ is expected to decrease whenever any constituent of $\boldsymbol{U}_r'$ gets affected by $\boldsymbol{u}_k (1 \le k \le r)$, $\boldsymbol{W} = \boldsymbol{U}_1\boldsymbol{Q}$ indicates the exclusive global maximum. This suggests that $E_1(\boldsymbol{W})$ possesses a global maximum with no other local maximum. By substituting $\boldsymbol{W} = \boldsymbol{U}_1\boldsymbol{Q}$ into $E_1(\boldsymbol{W})$, we obtain the global maximum,

$$E_1(\boldsymbol{W}) = \frac{1}{2}\sum_{i=1}^{r}\lambda_i - \frac{r}{2} \qquad (26)$$

This accomplishes the proof.

Adhering to the confirmed procedures in Theorems 1-2, it appears quite convenient to analyze the landscape of the following MSA cost function.

$$\begin{aligned}
\boldsymbol{W}^* &= \arg\max_{\boldsymbol{W}\in\Omega} E_2(\boldsymbol{W}) \\
E_2(\boldsymbol{W}) &= -\frac{1}{2}tr[(\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W})(\boldsymbol{W}^T\boldsymbol{W})^{-1}] \\
&\quad + \frac{1}{2}tr[\ln(\boldsymbol{W}^T\boldsymbol{W}) - \boldsymbol{W}^T\boldsymbol{W}] \qquad (27)
\end{aligned}$$

Theorem 3 and Theorem 4 illustrate the landscape of $E_2(\boldsymbol{W})$. Since the proven procedure is very similar to that of $E_1(\boldsymbol{W})$, we do not include the derivation steps.

Theorem 3: $\boldsymbol{W}$ is a stationary point of $E_2(\boldsymbol{W})$ in Ω if $\boldsymbol{W} = \boldsymbol{U}_r'\boldsymbol{Q}$, where $\boldsymbol{Q}$ represents an arbitrary orthogonal matrix.

Theorem 4: In Ω, $E_2(\boldsymbol{W})$ possesses a global maximum, which is obtainable if and only if $\boldsymbol{W} = \tilde{\boldsymbol{U}}_1\boldsymbol{Q}$, in which $\tilde{\boldsymbol{U}}_1 = [\boldsymbol{u}_{n-r+1}, \boldsymbol{u}_{n-r+2}, \cdots, \boldsymbol{u}_n]$. Every other stationary point is a saddle point of $E_2(\boldsymbol{W})$. Furthermore, at this global maximum, we have

$$E_2(\boldsymbol{W}) = -\frac{1}{2}\sum_{i=n-r+1}^{n}\lambda_i - \frac{r}{2} \tag{28}$$

Theorem 1 and Theorem 2 suggest that $E_1(\boldsymbol{W})$ possesses a global maximum with no local ones. Therefore, iterative approaches (such as the gradient method) can be utilized for the purpose of searching for the global maximum of $E_1(\boldsymbol{W})$. The same kinds of arguments are applicable for $E_2(\boldsymbol{W})$.

III. DUAL PURPOSE ALGORITHM AND COMPARISONS

A. DUAL PURPOSE ALGORITHM

The gradient of $J_{NUCF}(\boldsymbol{W})$ is given by (8). By utilizing this gradient as the iteration step size, we obtain the following algorithm

$$\begin{aligned}\boldsymbol{W}(t+1) = \boldsymbol{W}(t) \pm \mu\left[\boldsymbol{R}\boldsymbol{W}(t) - \frac{\boldsymbol{W}(t)\boldsymbol{W}^T(t)\boldsymbol{R}\boldsymbol{W}(t)}{\boldsymbol{W}^T(t)\boldsymbol{W}(t)}\right] \\ \times\left\{\boldsymbol{W}^T(t)\boldsymbol{W}(t)\right\}^{-1} + \mu\left[\frac{\boldsymbol{W}(t)}{\boldsymbol{W}^T(t)\boldsymbol{W}(t)} - \boldsymbol{W}(t)\right]\end{aligned} \tag{29}$$

Discretizing (29) and substituting the instant estimation of $\boldsymbol{R} = \boldsymbol{x}_k\boldsymbol{x}_k^T$ into it, we obtain the following discrete algorithm:

$$\begin{aligned}\boldsymbol{W}_{k+1} = \boldsymbol{W}_k \pm \mu\left[\boldsymbol{x}\boldsymbol{y}_k^T - \boldsymbol{W}_k\left(\boldsymbol{W}_k^T\boldsymbol{W}_k\right)^{-1}\boldsymbol{y}_k\boldsymbol{y}_k^T\right]\left(\boldsymbol{W}_k^T\boldsymbol{W}_k\right)^{-1} \\ + \mu\left[\boldsymbol{W}_k - \boldsymbol{W}_k\left(\boldsymbol{W}_k^T\boldsymbol{W}_k\right)\right]\left(\boldsymbol{W}_k^T\boldsymbol{W}_k\right)^{-1}\end{aligned} \tag{30}$$

where $0 < \mu \leq 1$ is the learning rate. If "+" is utilized, then (30) indicates a PSA algorithm, and "−" (30) indicates an MS tracking algorithm. Therefore, we denote equation (30) as the NUCF algorithm.

B. REMARKS AND COMPARISONS

In this part, we provide some statements on the NUCF algorithm and compare it to some other algorithms.

Remark 1: From [20], we know that the landscape of UIC is analyzed by the similar matrix methodology. In this paper, the landscape of NUCF is analyzed by the stable point methodology, which provides us with another viewpoint of the cost function. Moreover, this method also has general applicability and can be used to analyze other cost functions.

Remark 2: From (7), we can conclude that the NUCF are constituted by two terms. The first term is a Rayleigh Quotient, which can guarantee that the algorithms converge to the desired subspace (PS or MS). The second term is a penalty term that impacts the attributes of the algorithms, such as convergence speed. From [20], we see that the second term of the UIC is a quadratic formula of the weight matrix, while this part in the NUCF is a nonquadratic of the weight matrix. According to [9], the nonquadratic term can greatly improve the performance of neural network algorithms.

Remark 3: From (30), it is evident that the NUCF algorithm introduces the mechanism for the purpose of adaptively adjusting the step size at all steps. It emerges as quite apparent when the subspace dimension is one and $\boldsymbol{w}^T\boldsymbol{w}$ is scalar. In this scenario, the learning rate μ is a time varying step size $\mu_k = \mu/\boldsymbol{w}^T\boldsymbol{w}$ rather than a fixed value μ as in the UIC algorithm. In accordance with the research findings in [9], this adaptive learning rate is able to extensively promote the convergence speed of neural network algorithms.

Remark 4: The computation complexity of the NUCF algorithm is $3nr^2 + 4r^3/3 + 4nr$ flops per update, which poses to be the same as that of the UIC algorithm in [20]. It is also cheaper than $8nr^2 + o(nr^2)$ of the algorithm in [22] and $12nr^2 + o(nr^2)$ of the algorithm in [23]. Although the NUCF algorithm has the same computational complexity as the UIC algorithm, it has a faster convergence speed than the UIC algorithm, which will be proven by the following experiments.

Remark 5: Self stability (which studies the dynamic behaviors of the weight matrix norm) and global convergence analysis (which researches the convergence regions of algorithms) are two essential properties for neural network algorithms. Through further analysis, it is easy to prove that the NUIC algorithm is self-stabilizing and its convergence domain is $\Omega = \left\{\boldsymbol{W}|0 < \boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W} < \infty, \boldsymbol{W}^T\boldsymbol{W} \neq 0\right\}$. Since the proof procedures are similar to those in [20], they are omitted herein.

IV. NUMERICAL SIMULATIONS AND REAL APPLICATIONS

In this part, we have developed three experiments to illustrate the superiority of the NUCF algorithm. The first experiment presents some contrasting observations with other algorithms using numerical simulations. The second experiment illustrates the applicability of the NUCF algorithm for the DOA estimation. The last experiment applies the NUCF algorithm to solve the linear system detection issue.

For the purpose of measuring the attributes of these algorithms, we calculated the norm of the weight matrix

$$\rho(\boldsymbol{W}_k) = \|\boldsymbol{W}_k\|_F \tag{31}$$

and the deviation parameter [24]

$$dist(\boldsymbol{W}_k) = \left\|\boldsymbol{W}_k^T\boldsymbol{W}_k\left[diag\left(\boldsymbol{W}_k^T\boldsymbol{W}_k\right)\right]^{-1} - \boldsymbol{I}_r\right\|_F \tag{32}$$

where $diag(\boldsymbol{M})$ is a diagonal matrix and is obtained by setting all the off-diagonal elements of $\boldsymbol{M}$ to zero.

A. PS AND MS TRACKING

In this section, we compared the NUCF algorithm with some other prevalent algorithms. With respect to PS tracking, we compared the NUCF algorithm with the UIC [20] and NIC algorithms [9]. The NUCF algorithm is compared to both the UIC algorithm and the OJAm algorithm [24] for

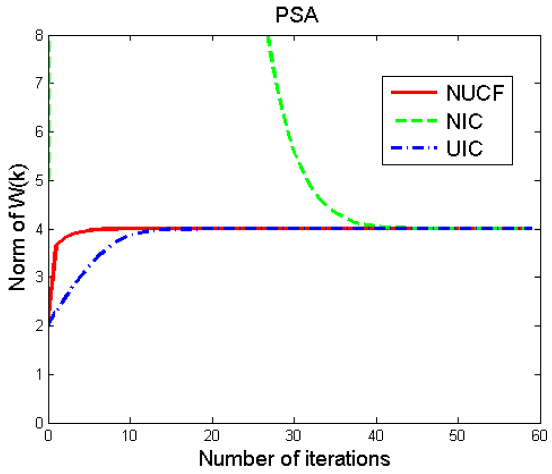


FIGURE 1. Norm curves for PS with $\mu = 0.2$.

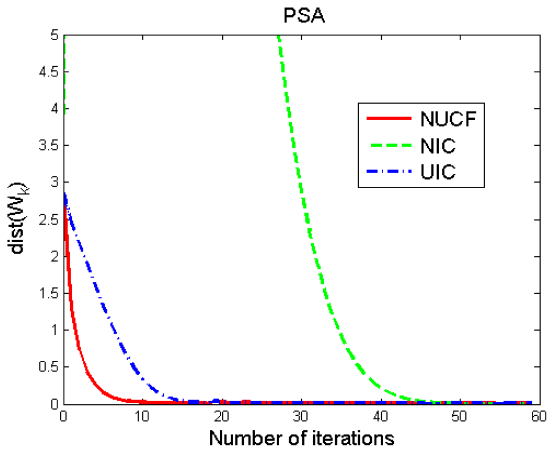


FIGURE 2. Deviation curves for PS with $\mu = 0.2$.

MS tracking. The input signal is generated by $\boldsymbol{x}_k = \boldsymbol{B} \cdot z_k$, where $\boldsymbol{B} = (1/31)\text{randn}(31, 31)$ is generated in the same way as that in [24] and its elements obey Gaussian distribution. $z_k \in \boldsymbol{R}^{31\times 1}$ denotes the zero-mean white noise with the variance $\sigma^2 = 1$. Thereafter, these three algorithms are applied to track the PS or MS of this signal. The dimensions of PS and MS are 16. For the purpose of making an appropriate comparison, these algorithms use the same preliminary weight matrix and the learning rate. The simulation findings are presented in Fig. 1-Fig.4. These simulation results are attained by averaging 100 independent experiments.

From Fig. 1, it is observable that after approximately 45 steps, the weight matrix norm tends to a constant. Furthermore, the deviation curve in Fig. 2 trends to zero. As suggested by these two figures, we can observe that the NUCF algorithm is capable of tracking the PS of a signal. Fig. 3 and Fig. 4 shed light on the simulation findings that are attained when the NUCF algorithm is put to use as an MSA algorithm. As suggested by these two figures, we confirm that the NUCF algorithm can track the MS of a signal.

From both Fig. 1 and Fig. 2, it is also observable that when it is utilized to track the PS of a signal, the

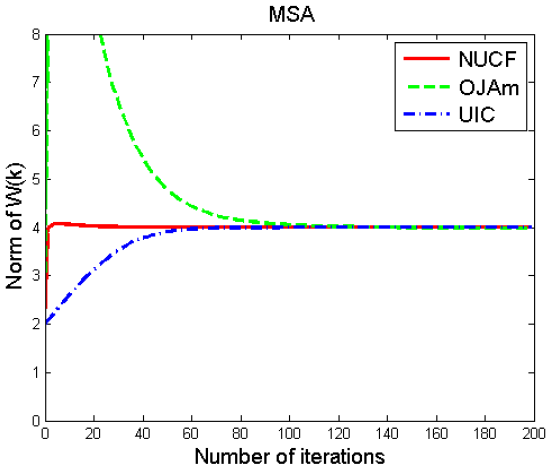


FIGURE 3. Norm curves for MS with $\mu = 0.05$.

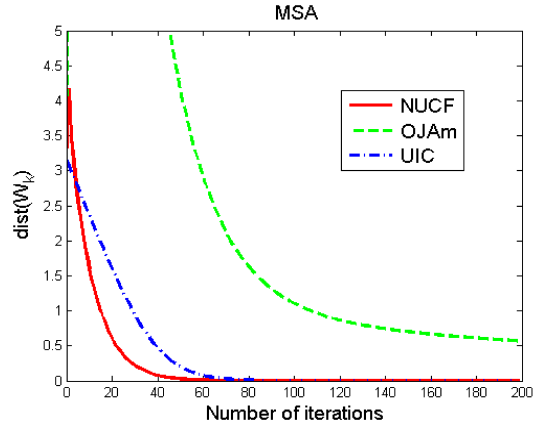


FIGURE 4. Deviation curves for MS with $\mu = 0.05$.

NUCF possesses a quicker convergence speed than both the NIC algorithm and the UIC algorithm for both the weight matrix norm and deviation parameters. The speed benefits of the NUCF algorithm are also apparent to us from Fig. 3 and Fig. 4, where it serves as an MSA algorithm. To conclude, as this experiment reveals, we prove that the NUCF algorithm can track the PS and MS of an input signal and it has a faster convergence speed than some existing algorithms.

B. DOA ESTIMATION

This experiment is a DOA estimation. It is designed to examine the applicability of the NUCF algorithm in tracking the PS from signals. There are K narrow band signals ($\boldsymbol{S} = [s_1(t), s_2(t), \cdots, s_K(t)]^T$) that impinge on a homogeneous linear array having N antennas. The observed signals on this array are stated as $\boldsymbol{y}(t)$ such that

$$\boldsymbol{y}(t) = \boldsymbol{A}\boldsymbol{S}(t) + \boldsymbol{n}(t) \tag{33}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_K]$ indicates the $n \times K$ array response matrix. Its column vector is

$$\boldsymbol{a}_k = \left[1, e^{-j2\pi f_0 \frac{d\sin\theta_k}{c}}, \cdots, e^{-j2\pi f_0 \frac{(N-1)d\sin\theta_k}{c}}\right]^T \tag{34}$$

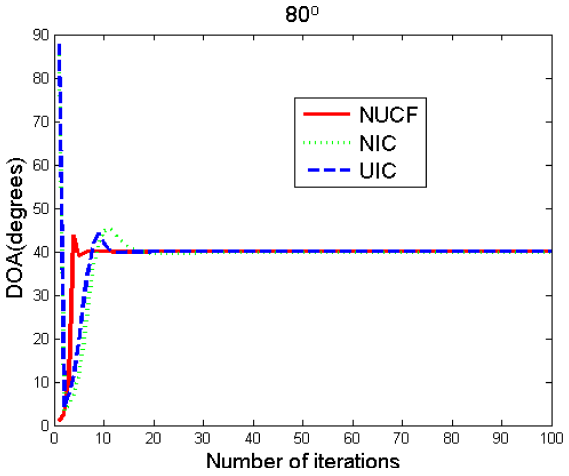


FIGURE 5. DOA curves of three algorithms (80°).

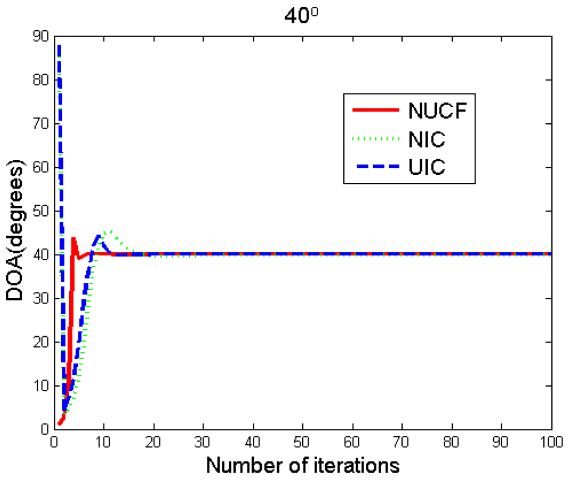


FIGURE 6. DOA curves of three algorithms (40°).

In the field of signal processing, $\boldsymbol{a}_k$ is also stated as the frequency vector. f_0 is the carrier frequency, c is equal to the speed of light (i.e. $c = 3 \times 10^8 (m/s)$), d is the length between the two array antennas, θ_k is the direction angle of the $k$thsignals and $n(t)$ is the observed noise vector. The DOA aims at estimating the direction angle from the observed signals.

The ESPRIT method [25] (which is accomplished on the basis of the PSA approach) is a popular approach for solving the DOA problem. The measurement procedure is available in [25] and we do not include it here. Suppose that two incident signals impinge on an antenna array that has 13 units from the directions of 40° and 80° . The observed noise amounts are zero mean Gaussian noise, which have the variance $\sigma^2 = 1$. Moreover, the signal-to-noise ratio (SNR) is 10dB. Thereafter, we apply the ESPRIT method to solve the problem of estimating the direction angles.

All through the calculation of the ESPRIT method, the PS tracking is accomplished by NUCF, UIC [20] and NIC [9], respectively. Similar to the previous experiment, the same primary parameters are utilized for these algorithms. We set $\mu = 0.05$. Moreover, the primary weight matrix is randomly produced and its elements obey Gaussian distribution. The estimation results of these algorithms are presented in Fig. 5 and Fig. 6.

From the two figures, it is observed that the NUCF algorithm can estimate the direction angles from the observed signals. When comparing the NUCF algorithm to the other two algorithms, it is very easy to observe that the NUCF algorithm possesses the fastest convergence speed among the three PSA algorithms.

C. LINEAR SYSTEM IDENTIFICATION

In this experiment, the use of some MSA algorithms has been made to solve the TLS problem. Considering the linear system detection, the application of the TLS method can be made with respect to the parameter assessment of the adaptive filters [26].

The signal model of the adaptive filter is stated as follows. Suppose that at time k, the input vector of the adaptive filter is $\tilde{\boldsymbol{p}}(k) = \boldsymbol{p}(k) + \boldsymbol{n}_1(k)$, where $\boldsymbol{p}(k)$ is the true input signal and $\boldsymbol{n}_1(k)$ represent the additive noise vector. The output of this filter is given by $\tilde{q}(k) = q(k) + n_2(k)$, where $n_2(k)$ is the additive output noise. Denote $\boldsymbol{h}(k) = [h_1, h_2, \ldots, h_n]^T$ as the estimation parameter vector of the adaptive filter, then the representation of the filter output at time k can be stated as $y(k) = \tilde{\boldsymbol{p}}^T(k)\boldsymbol{h}(k)$. The output error of this filter is $\varepsilon(k) = y(k) - \tilde{q}(k)$. The linear system identification estimates the parameters of the adaptive filter from the observations of both the input and the output.

To address this issue, let us denote $\boldsymbol{z}(k) = [\tilde{\boldsymbol{p}}^T(k), \tilde{q}(k)]^T$ and $\boldsymbol{w}(k) = [\boldsymbol{h}^T(k), -1]^T$. Then we have $\varepsilon(k) = \boldsymbol{z}^T(k)\boldsymbol{w}(k)$. On the basis of the above derivation, estimation of the parameters of this adaptive filter can be transformed into minimizing the following equation:

$$J(k) = \frac{\boldsymbol{w}(k)^T \boldsymbol{R} \boldsymbol{w}(k)}{\boldsymbol{w}(k)^T \boldsymbol{w}(k)} = \frac{E[\varepsilon(k)^2]}{\|\boldsymbol{w}(k)\|_2^2} \tag{35}$$

where $\boldsymbol{R} = E\{\boldsymbol{z}(k)\boldsymbol{z}(k)^T\}$ indicates the autocorrelation matrix of $\boldsymbol{z}(k)$. Clearly, (35) is the Rayleigh Quotient of $\boldsymbol{w}(k)$. Therefore, the minimum point of $J(k)$ can be attained when $\boldsymbol{w}(k)$ is equal to the MC of the vector sequence $\{\boldsymbol{z}(k)\}$.

In this simulation, we use the OJAm [24], UIC [20] and NUCF algorithms to identify this linear system. The parameters of this adaptive filter are given by:

$$\boldsymbol{h} = [-0.3, -0.9, 0.8, -0.7, 0.6]^T \tag{36}$$

The unknown system is triggered by a zero-mean homogeneously allocated random white signal with its variance equal to unity. The additive input noise $\boldsymbol{n}_1(k)$ and output noise $n_2(k)$ appear to be zero-mean white Gaussian noise with the variance $\sigma^2 = 0.01$. The learning rate of the three algorithms is set as $\mu = 0.1$. Moreover, the preliminary weight vector is produced in the same way as that in the above experiment. For the purpose of manifesting the convergence speed of the

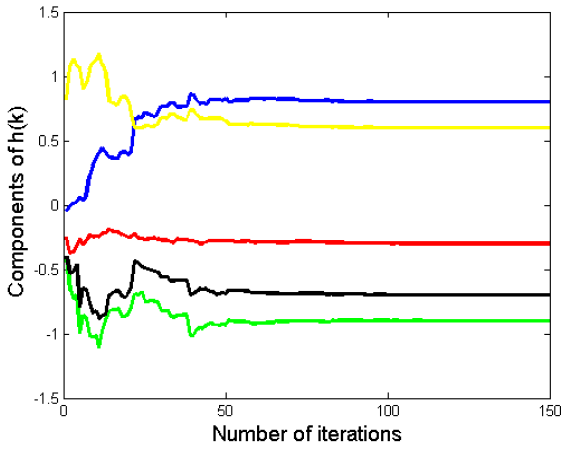


FIGURE 7. Convergence of the components.

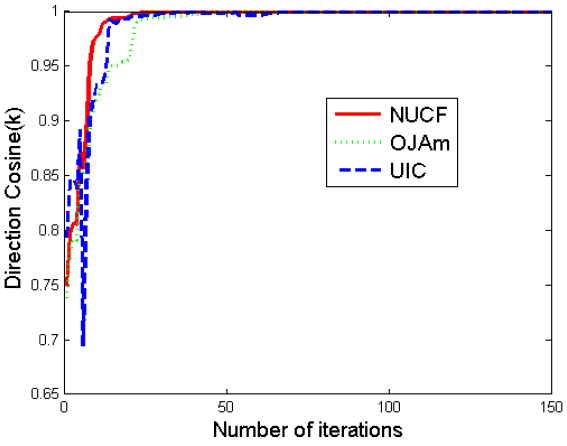


FIGURE 8. Convergence curves of three algorithms.

three algorithms, we calculate the direction cosine (DC) at every step, which is given by

$$\text{DirectionCosine}(k) = \frac{\left|\boldsymbol{h}^T(k)\boldsymbol{h}\right|}{\|\boldsymbol{h}(k)\|\,\|\boldsymbol{h}\|} \tag{37}$$

Fig. 7 illustrates the dynamic conduct of the constituents of the weight vector $\boldsymbol{h}(k)$. As suggested by this figure, we can conclude that the $\boldsymbol{h}(k)$ converges to the weight vector of the linear system. Therefore, the NUCF algorithm can be used to solve the linear system identification problem. Fig. 8 reveals the DC curves of the three algorithms. As is evident in Fig. 8, we can confirm again that the NUCF algorithm possesses faster convergence speed when compared to the other two algorithms for MSA.

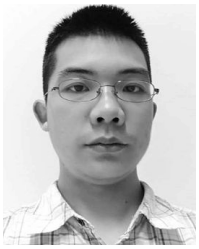
V. CONCLUSION

In this paper, a novel cost function, which is actually an unconstrained optimization formula, is proposed for both the PS and the MS tracking. The theoretical analysis reveals that the proposed function contains only one maximum, which is achieved if and only if the weight matrix produces an orthonormal foundation of the desired subspace (PS or MS) of an input signal. Through the use of gradient ascent method on the proposed function, a dual purpose algorithm is established. In comparison with some prevalent algorithms, the extracted dual purpose algorithm offers lower computational complexity and faster convergence speed. A numerical simulation and real application experiments prove that the derived algorithm is capable of efficiently and satisfactorily tracking the desired subspace.

REFERENCES

[1] T. D. Nguyen and I. Yamada, "Necessary and sufficient conditions for convergence of the DDT systems of the normalized PAST algorithms," *Signal Process.*, vol. 94, pp. 288–299, Jan. 2014.
[2] G. Ghinea, R. Kannan, and S. Kannaiyan, "Gradient-orientation-based PCA subspace for novel face recognition," *IEEE Access*, vol. 2, pp. 914–920, 2014.
[3] Y.-H. Shao, N.-Y. Deng, W.-J. Chen, and Z. Wang, "Improved generalized eigenvalue proximal support vector machine," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 213–216, Mar. 2013.
[4] M. Thameri, K. Abed-Meraim, and A. Belouchrani, "Minor subspace tracking using MNS technique," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2012, pp. 2433–2436.
[5] V.-D. Nguyen, K. Abed-Meraim, N. Linh-Trung, and R. Weber, "Generalized MNS method for parallel minor and principal subspace analysis," in *Proc. 22nd Eur. Signal Process. Conf.*, 2014, pp. 2265–2269.
[6] M. Arjomandi-Lari and M. Karimi, "Generalized YAST algorithm for signal subspace tracking," *Signal Process.*, vol. 117, pp. 82–95, Dec. 2015.
[7] R. Wang, M. Yao, D. Zhang, and H. Zou, "A novel orthonormalization matrix based fast and stable DPM algorithm for principal and minor subspace tracking," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 466–472, Jan. 2011.
[8] C. Pehlevan, T. Hu, and D. B. Chklovskii, "A Hebbian/anti-Hebbian neural network for linear subspace learning: A derivation from multidimensional scaling of streaming data," *Neural Comput.*, vol. 27, no. 7, pp. 1461–1495, 2015.
[9] Y. Miao and Y. Hua, "Fast subspace tracking and neural network learning by a novel information criterion," *IEEE Trans. Signal Process.*, vol. 46, no. 7, pp. 1967–1979, Jul. 1998.
[10] A. Valizadeh, M. Farrokhrooz, and A. Rafiei, "Fast signal subspace tracking using two-layer linear neural network learning," in *Proc. 16th Med. Conf. Control Auto.*, 2008, pp. 1828–1832.
[11] R. Wang, M. Yao, D. Zhang, and H. Zou, "Stable and orthonormal OJA algorithm with low complexity," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 211–214, Apr. 2011.
[12] S. Bartelmaos and K. Abed-Meraim, "fast adaptive algorithms for minor component analysis using householder transformation," *Digit. Signal Process*, vol. 21, pp. 667–678, Dec. 2011.
[13] D. Peng, Y. Zhang, Y. Xiang, and H. Zhang, "A globally convergent MC algorithm with an adaptive learning rate," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 359–365, Feb. 2012.
[14] X. Feng, H. Ma, X. Kong, and C. Zhang, "Fast and stable coupled minor component analysis rules," in *Proc. Adv. Inf. Technol., Electron. Auto. Control Conf. (IAEAC)*, 2015 pp. 54–59.
[15] R. Möller, "Derivation of coupled PCA and SVD learning rules from a Newton zero-finding framework," Dept. Comput. Eng., Faculty Technol., Bielefeld Univ., Bielefeld, Germany, Tech. Rep. 15, 2017.
[16] C. Tianping, "Modified Oja's algorithms for principal subspace and minor subspace extraction," *Neural Process. Lett.*, vol. 5, pp. 105–110, Apr. 1997.
[17] X. Kong, C. Hu, H. Ma, and C. Han, "A unified self-stabilizing neural network algorithm for principal and minor components extraction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 185–198, Feb. 2012.
[18] M. A. Hasan, "Dual systems for minor and principal component computation," in *Proc. Acoustics, Speech Signal Process. (ICASSP)*, 2008, pp. 1901–1904.
[19] D. Peng, Z. Yi, and Y. Xiang, "A unified learning algorithm to extract principal and minor components," *Digit. Signal Process.*, vol. 19, no. 4, pp. 640–649, 2009.
[20] X. Kong, C. Hu, and C. Han, "A dual purpose principal and minor subspace gradient flow," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 197–210, Jan. 2012.

[21] G. Yingbin, K. Xiangyu, H. Changhua, L. Hongzeng, and H. Li'an, "A generalized information criterion for generalized minor component extraction," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 947–959, Feb. 2017.
[22] G. Mathew, V. U. Reddy, and S. Dasgupta, "Adaptive estimation of eigensubspace," *IEEE Trans. Signal Process.*, vol. 43, no. 2, pp. 401–411, Feb. 1995.
[23] Z. Fu and E. M. Dowling, "Conjugate gradient eigenstructure tracking for adaptive spectral estimation," *IEEE Trans. Signal Process.*, vol. 43, no. 5, pp. 1151–1160, May 1995.
[24] D.-Z. Feng, W.-X. Zheng, and Y. Jia, "Neural network learning algorithms for tracking minor subspace in high-dimensional data stream," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 513–521, May 2005.
[25] K. Helwani, *Fundamentals of Adaptive Filter Theory*. Berlin, Germany: Springer, 2015.
[26] G. Cirrincione and M. Cirrincione, *Neural-Based Orthogonal Data Fitting: The EXIN Neural Networks*. Hoboken, NJ, USA: Wiley, 2011.



DONG HAI-DI received the bachelor's degree in control science and engineering and master's degree in control science and engineering from the Xi'an Research Institute of High Technology, Xi'an, China, in 2011 and 2014, respectively, where he is currently pursuing the Ph.D. degree.

His current research interests include adaptive signal processing and feature extraction.

GAO YING-BIN received the bachelor's, master's, and Ph.D. degrees from the Xi'an Research Institute of High Technology, Xi'an, China, in 2009, 2011, and 2016, respectively, all in control science and engineering. He is currently an Engineer with the China Electronics Technology Group Corp 54th Research Institute, Shijiazhuang, China.

His research interests include adaptive signal processing and feature extraction.

HE BING received the bachelor's, master's, and Ph.D. degrees from the Xi'an Research Institute of High Technology, Xi'an, China, in 2005, 2008, and 2012, respectively, all in control science and engineering. He is currently an Assistant Professor with the Xi'an Institute of High Technology, Xi'an, China.

His research interests include fault diagnosis and fault tolerant control.

LIU GANG received the bachelor's and master's degrees from the Xi'an Institute of High Technology, Xi'an, China, in 1988 and 1991, respectively, and the Ph.D. degree from Northwestern Polytechnical University, Xi'an, in 1998. He is currently a Professor with the Xi'an Institute of High Technology.

His research interests include adaptive signal processing, system modeling, and fault diagnosis.