IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 15, 2018, accepted July 30, 2018, date of publication August 6, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2863557

An Improved Adaptive Subspace Tracking Algorithm Based on Approximated Power Iteration

QIANG WU[1,2], JIAN-SHENG ZHENG[1], ZHI-CHENG DONG[2], (Member, IEEE), ERDAL-PANAYIRCI[3,4], (Life Fellow, IEEE), ZHI-QIANG WU[2,5], (Senior Member, IEEE), AND REN QINGNUOBU[2]

[1]Electronic Information School, Wuhan University, Wuhan 430072, China
[2]School of Engineering, Tibet University, Lhasa 850000, China
[3]Department of Electrical and Electronics Engineering, Kadir Has University, 34230 Istanbul, Turkey
[4]Department of Electrical Engineering, Princeton University, Princeton, NY 08544, USA
[5]Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA

Corresponding author: Jiansheng Zheng (jszhengwhu@qq.com)

This work was supported in part by the National Science Foundation of China (NSFC) under Grant 61561046, in part by the Key Project of Science Foundation of Tibet Autonomous Region under Grant 2015ZR-14-3, in part by the 2015 Outstanding Youth Scholars of Everest Scholars Talent Development Support Program of Tibet University, and in part by the 2017 Tibet autonomous region university research and innovation team Plateau Communication Research and Innovation Team.

ABSTRACT A subspace tracking technique has drawn a lot of attentions due to its wide applications. The main objective of this approach is to estimate signal or noise subspace basis for the sample covariance matrix. In this paper, we focus on providing a fast, stable, and adaptive subspace tracking algorithm that is implemented with low computational complexity. An alternative realization of the fast approximate power iteration (FAPI) method, termed modified FAPI (MFAPI), is also presented. Rather than solving an inverse square root of a matrix employed in the FAPI, the MFAPI applies the matrix product directly to ensure the orthonormality of the subspace basis matrix at each recursion. This approach yields a simpler derivation and is numerically stable while maintaining a similar computational complexity as compared with that of the FAPI. Furthermore, we present a detailed mathematical proof of the numerical stability of our proposed algorithm. Computer simulation results indicate that the MFAPI outperforms many classical subspace tracking algorithms, particularly at the transient-state step.

INDEX TERMS Adaptive subspace tracking, approximated power iteration, orthonormal iteration, projection approximation.

I. INTRODUCTION

The subspace tracking methods, that are employed to track the change of structure in signals, intend to divide the observed data into two parts: signal subspace and noise subspace. Since a more distinct representation of useful signals could be obtained in this way, these methods have drawn lots of interests recently. There are many subspace-based application in practice such as sensor networks [1], signal detection [2]–[4], machine learning [5], array processing [6], etc. In some applications, these methods can detect very weak signals in a very noisy environment at signal-to-noise ratio (SNR) as low as to -23dB [4].

The subspace tracking methods can be described as follows: let $\boldsymbol{x}(t)$ be a N-dimensional received data vector with sample covariance matrix $\boldsymbol{C}_{xx}(t) = E[\boldsymbol{x}(t)\boldsymbol{x}(t)^H]$. The main objective is then to estimate the largest L-dimensional signal subspace or the smallest $(N-L)$-dimensional noise subspace spanned by the basis matrix $\boldsymbol{W}(t) \in \mathbb{C}^{N\times L}$ or $\boldsymbol{W}(t) \in \mathbb{C}^{N\times\,(N-L)}$ of $\boldsymbol{C}_{xx}(t)$, respectively, where L is the number of the useful signals in a noisy environment.

There exist many different methods related with subspace tracking in literature. Computational complexity measure is a commonly employed indicator to classify and distinguish these algorithms. According to [17], the computational complexity of an algorithm can be classified as high, medium, or low based on the number of operations, $O(N^3)$, $O(N^2L)$ and $O(NL)$, respectively. Direct decomposition methods, such as batch singular value decomposition (SVD) and eigenvalue decomposition (EVD), are not suitable techniques in real applications due to their high computational complexity in order of $O(N^3)$, even though they yield quite accurate eigenvalue and eigenvector estimations. Presently, almost all mainstream subspace tracking methods have been able to reduce the computational complexity to the order $O(NL)$ operations. In practice, these algorithms find wide range of applications [7], especially with massive data computations. Therefore, in this paper, we are mainly concerned with these mainstream low-complexity schemes.

Subspace-based algorithms are designed to track the signal (principal) or noise (minor) subspace. Generally, according to the tracking type, these methods can be categorized into two types: the algorithms for tracking only one single subspace or tracking both subspaces simultaneously. Also, most subspace tracking algorithms can be viewed as a constrained or unconstrained optimization problem [7]. Benefiting from using recursive least squares (RLS) approach, the popular types of subspace tracking methods, such as projection approximation subspace tracking (PAST) [8], orthonormal PAST (OPAST) [9], fast approximated power iteration (FAPI) [10], have very fast convergence rates in estimating the signal or noise subspace. However, these methods are not without their drawbacks. Although the method based on PAST convergences to the basis matrix, orthonormalization can not be guaranteed in the initial stage. Comparing with PAST, the OPAST has a faster convergence rate due to its orthonormalization to the signal subspace matrix at each iterative step. Both PAST and OPAST can be regarded as a first order approximation of the FAPI. Hence, FAPI outperforms PAST and OPAST. However, the main weakness of the FAPI algorithm is that its derivation is relatively difficult and its numerical instability at the transient-state step (details will be described in the following sections). For the PAST-type algorithms, a more detailed description can be found in [11]. Comparison with RLS employed in those algorithms, the FRANS algorithm can track noise subspace basis based on Rayleigh's quotient [12]. In order to overcome the instability and slow convergence rate of FRANS, a algorithm called HFRANS is proposed using Householder transformation [13] whose detailed analysis is presented in [14]. Note that, these algorithms belong to the first type.

The second type of methods, mainly inherited from the data projection method (DPM) [15], can track dual signal or noise subspace matrix simultaneously with a sign change. Popular algorithms include ODKA [16], FDPM [17], YAST [18], SGYAST [19], and algorithm proposed in [20]. Although most of the DPM-based algorithms have simple code structure and ability to track dual subspace basis matrix, they have slower convergence rate than that of the first type of algorithms.

Considering the above mentioned problems, this paper presents a new implementation of the tracking signal subspace approach based on FAPI for the exponential window case. Relying on the direct matrix product, a simple but more stable method is provided to guarantee orthonormalization of subspace basis matrix. Additionally, we discuss convergence rates of the two types of algorithms mentioned above through the results obtained by detail computer simulations. Finally, a thorough mathematical proof of the numerical stability is given which makes it easier to understand our algorithm. To the best of our knowledge, similar description has not been proposed in the literature yet.

The rest of the paper is organized as follows. In Section II, problem definition is provided; In Section III, we first review the exponential window estimate of the sample covariance matrix, then present a detailed derivation of our MFAPI algorithm; performance analyses, including computational complexity, convergence rate and numerical stability of proposed MFAPI method are presented in Section IV; finally, simulation results and conclusion are provided in Section V and Section VI, respectively.

II. PROBLEM DEFINITION AND NOTATION

We first provide the notations used in this paper. Scalars are denoted by lower case Greek letters, the symbol $\mathbb{C}^{m\times n}$ represents the complex matrix with m rows and n columns, $\boldsymbol{I}_r$ represents the $r \times r$ identity matrix. The transpose of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^T$ and the conjugate-transpose by $\boldsymbol{A}^H$. The symbol $\|\cdot\|$ denotes the 2-norm of a vector, $\|\cdot\|_F^2$ denotes the matrix Frobenius norm, $\mathrm{diag}\{\cdot\}$ is a diagonal matrix.

In this paper, the subspace basis matrix is estimated by our proposed subspace-based algorithm, which can be applied to the various scenarios described in the introduction. However, in order to evaluate the performance of the proposed algorithm, a special example of detecting the frequencies of signals is considered.

Let $\boldsymbol{x}(t)$ be a received data vector. $\boldsymbol{x}(t)$ can be written as $\boldsymbol{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$ obtained from the output of N different sensors of an array at the tth snapshot, or $\boldsymbol{x}(t) = [x(t), x(t-1), \ldots, x(t-N+1)]^T$ obtained from the N sequentially received data of a time series. The signal model, including L complex exponential signals, can be expressed as

$$\boldsymbol{x}(t) = \sum_{l=1}^{L} p_l(t)\boldsymbol{e}(\omega_l) + \boldsymbol{n}(t) \tag{1}$$

where $p_l(t)$ is the random signal, ω_l denotes the lth frequency to be detected, $\boldsymbol{n}(t)$ is a stationary zero-mean additive white Gaussian noise (AWGN) vector with variance σ^2. $\boldsymbol{e}(\omega_l) = [1\ \exp(j\omega_l)\ \exp(j2\omega_l)\ \cdots\ \exp(j(N-1)\omega_l)]^{\mathrm{T}}$ is the frequency vector. Let $\boldsymbol{E}(\omega) = [\boldsymbol{e}(\omega_1), \boldsymbol{e}(\omega_2), \ldots, \boldsymbol{e}(\omega_L)]$ be the matrix of ω and $\boldsymbol{P} = [p_1, p_2, \ldots, p_L]^T$ be a vector of p. Equation (1) can then be rewritten as

$$\boldsymbol{x}(t) = \boldsymbol{E}(\omega)\boldsymbol{P} + \boldsymbol{n}(t). \tag{2}$$

Here, we assume that the signal is independent of the noise. Then, the covariance matrix of $\boldsymbol{x}(t)$ can be obtained as [15]

$$\boldsymbol{C}_{xx}(t) = \mathrm{E}[\boldsymbol{x}(t)\boldsymbol{x}(t)^H] = \boldsymbol{E}\boldsymbol{P}_s\boldsymbol{E}^H + \sigma^2\boldsymbol{I}_N, \tag{3}$$

where $\boldsymbol{P}_s = \mathrm{E}[\boldsymbol{P}\boldsymbol{P}^H]$ is the covariance matrix of the signal components, $E[\cdot]$ denotes the expectation operator. By applying EVD, $\boldsymbol{C}_{xx}(t)$ can be further expressed as the sum of the signal and the noise covariance matrices as

$$\boldsymbol{C}_{xx}(t) = \boldsymbol{W}_s\boldsymbol{\Sigma}_s\boldsymbol{W}_s^H + \boldsymbol{W}_\psi\boldsymbol{\Sigma}_\psi\boldsymbol{W}_\psi^H, \tag{4}$$

where $\boldsymbol{W}_s$ and $\boldsymbol{W}_\psi$ denote the signal and noise subspace basis matrices, respectively, $\boldsymbol{\Sigma}_s = \mathrm{diag}\{\lambda_1, \cdots, \lambda_L\}$ and $\boldsymbol{\Sigma}_\psi = \mathrm{diag}\{\lambda_{L+1}, \cdots, \lambda_N\}$ denote signal and noise eigenvalues, respectively. $\boldsymbol{C}_{xx}(t)$ is semi-positive definite matrix whose eigenvalues satisfy

$$\lambda_1 \geq \cdots \geq \lambda_L \geq \lambda_{L+1} = \lambda_{L+2} \cdots = \lambda_N = \sigma^2 \geq 0. \tag{5}$$

The basis matrix $\boldsymbol{W}_\psi$ is employed to estimate the frequencies ω_l of the signal components. Some algorithms such as MUSIC [21] and ESPRIT [22] can be used for this purpose. Th MUSIC algorithms is described as follows. Equation (4) can be rewritten as

$$\boldsymbol{C}_{xx}(t) = [\boldsymbol{W}_s, \boldsymbol{W}_\psi]\begin{bmatrix}\boldsymbol{\Sigma}_s & \boldsymbol{0}\\ \boldsymbol{0} & \sigma^2\boldsymbol{I}_{N-L}\end{bmatrix}\begin{bmatrix}\boldsymbol{W}_s^H\\ \boldsymbol{W}_\psi^H\end{bmatrix}. \tag{6}$$

Considering $\boldsymbol{W}_s^H\boldsymbol{W}_\psi = \boldsymbol{0}$ and $\boldsymbol{W}_\psi^H\boldsymbol{W}_\psi = \boldsymbol{I}$, we have

$$\boldsymbol{C}_{xx}(t)\boldsymbol{W}_\psi = [\boldsymbol{W}_s, \boldsymbol{W}_\psi]\begin{bmatrix}\boldsymbol{\Sigma}_s & \boldsymbol{0}\\ \boldsymbol{0} & \sigma^2\boldsymbol{I}_{N-L}\end{bmatrix}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}\end{bmatrix} = \sigma^2\boldsymbol{W}_\psi. \tag{7}$$

Using (3) we derive

$$\boldsymbol{C}_{xx}(t)\boldsymbol{W}_\psi = \boldsymbol{E}\boldsymbol{P}_s\boldsymbol{E}^H\boldsymbol{W}_\psi + \sigma^2\boldsymbol{W}_\psi. \tag{8}$$

And from (7), (8) we have

$$\boldsymbol{E}\boldsymbol{P}_s\boldsymbol{E}^H\boldsymbol{W}_\psi = \boldsymbol{0}. \tag{9}$$

It then follows that

$$\boldsymbol{W}_\psi^H\boldsymbol{E}\boldsymbol{P}_s\boldsymbol{E}^H\boldsymbol{W}_\psi = \boldsymbol{0}. \tag{10}$$

Note that when $\boldsymbol{Q}$ is nonsingular, $\boldsymbol{t}^H\boldsymbol{Q}\boldsymbol{t} = \boldsymbol{0}$ if, and only if, $\boldsymbol{t} = \boldsymbol{0}$ holds. Hence, from (10)

$$\boldsymbol{E}^H\boldsymbol{W}_\psi = \boldsymbol{0}. \tag{11}$$

Substituting $\boldsymbol{E}(\omega) = [\boldsymbol{e}(\omega_1), \boldsymbol{e}(\omega_2), ..., \boldsymbol{e}(\omega_L)]$ into (11) yields

$$\boldsymbol{e}(\omega)^H\boldsymbol{W}_\psi = \boldsymbol{0}, \quad \omega = \omega_1, \omega_2, ..., \omega_L. \tag{12}$$

Finally, for the given $\boldsymbol{W}_\psi$, let

$$p(\omega) = \frac{1}{\boldsymbol{e}(\omega)^H\boldsymbol{W}_\psi\boldsymbol{W}_\psi^H\boldsymbol{e}(\omega)} \tag{13}$$

be a spectral function of the parameter ω. We can then obtain L largest peaks of $p(\omega)$ by searching over the values of ω. The ω's corresponding to the L largest peaks are the estimated frequencies.

III. PROPOSED MFAPI ALGORITHM

FAPI has excellent capability as compared to other existing subspace-based algorithms in tracking the subspace basis matrix and, especially, in achieving better convergence rate [10]. However, the main weakness of FAPI is that it needs to evaluate the inverse square root of matrix to implement orthonormalization for subspace basis matrix. It is known that this evaluation may cause numerical instability with a substantially higher computational complexity. On the other hand, the proposed MFAPI method makes use of direct matrix multiplication for orthonormalization to obtain a faster convergence rate than the FAPI and guarantees numerical stability. In the rest of this section, a detail derivation of MFAPI algorithm based on FAPI is presented.

A. ESTIMATE OF SAMPLE COVARIANCE MATRIX

Subspace tracking methods need to estimate their sample covariance matrix from the received data. Usually, the adapted schemes for covariance matrix estimation is based on using the exponential and truncated windowings whose details can be found in [10].

1) EXPONENTIAL WINDOW ESTIMATE

In general, exponential windowing, from which $\boldsymbol{C}_{xx}(t)$ is estimated, is defined as

$$\boldsymbol{C}_{xx}(t) = \sum_{k=-\infty}^{t} \beta^{t-k}\boldsymbol{x}(k)\boldsymbol{x}(k)^H, \tag{14}$$

where β is a forgetting factor with $0 < \beta \leq 1$. Equation (14) usually has another form for the recursive purpose and it can be written as [8]

$$\boldsymbol{C}_{xx}(t) = \beta\boldsymbol{C}_{xx}(t-1) + \boldsymbol{x}(t)\boldsymbol{x}(t)^H. \tag{15}$$

Exponential windowing-based covariance estimation puts emphasis on the new data so it fits in non-stationary scenarios when β is small enough. Meanwhile, the estimate of covariance matrix using exponential windowing can be computed simply by an iterative algorithm without storing the historical sample data, that is very useful in big data applications.

2) TRUNCATED WINDOW ESTIMATE

Truncated windowing-based covariance estimation can be defined as [10]

$$\boldsymbol{C}_{xx}(t) = \sum_{k=t-l+1}^{t} \beta^{t-k}\boldsymbol{x}(k)\boldsymbol{x}(k)^H, \tag{16}$$

where $\beta = 1$ corresponds to the case of sliding windowing, l is the window length. For iterative purpose, truncated windowing has the following form:

$$\boldsymbol{C}_{xx}(t) = \beta\boldsymbol{C}_{xx}(t-1) + \boldsymbol{x}(t)\boldsymbol{x}(t)^H - \beta^l\boldsymbol{x}(t-l)\boldsymbol{x}(t-l)^H. \tag{17}$$

In this paper, for simplicity, we only consider implementation of MFAPI using an exponential windowing due to the following reasons. First, our derivation method for MFAPI can be realized easily with the truncated windowing case. Secondly, for the FAPI, computational complexity is much lower in case of the exponential windowing than the truncated windowing. Namely, it is about $3NL$ ($3NL$ is the lowest complexity among the current low-complexity subspace tracking algorithms) for the exponential windowing while $6NL$ for the truncated windowing.

B. POWER ITERATION

Power iteration [7] is a commonly employed method to compute dominant subspace basis matrix. Since FAPI algorithm has a similar code structure with power iteration, we describe the power iteration method first. With sequentially received data $\boldsymbol{x}(t)$, conventional power iteration method computes the subspace basis matrix $\boldsymbol{W}(t)$ by the following formulas

$$\boldsymbol{y}(t) = \boldsymbol{W}(t-1)^H \boldsymbol{x}(t), \tag{18}$$

$$\boldsymbol{C}_{xy}(t) = \boldsymbol{C}_{xx}(t)\boldsymbol{W}(t-1), \tag{19}$$

$$\boldsymbol{W}(t)\boldsymbol{R}(t) = \boldsymbol{C}_{xy}(t), \tag{20}$$

where L-dimensional $\boldsymbol{y}(t)$ can be viewed as data compressing step by projecting N-dimensional $\boldsymbol{x}(t)$ onto $\boldsymbol{W}(t-1)$ (in general, $L \ll N$), and $\boldsymbol{C}_{\boldsymbol{xy}}(t)$ as correlation matrix between $\boldsymbol{x}(t)$ and $\boldsymbol{y}(t)$. $\boldsymbol{W}(t)$ can be computed by the factorization step (20) by means of the QR decomposition or other orthonormalization methods. $\boldsymbol{R}(t)$ is a matrix factor of $\boldsymbol{C}_{xy}(t)$. It is an upper triangular matrix if QR decomposition applied. Hence, the power iteration method can eventually converge to the principal subspace basis matrix.

C. FAPI ALGORITHM

FAPI algorithm is designed for estimating subspace basis matrix with low-computational complexity and fast convergence rate. In the following, we describe the implementation of this algorithm. As discussed in [10], the following approximation is employed for projection of $\boldsymbol{W}(t-1)$ into $\boldsymbol{W}(t)$:

$$\boldsymbol{W}(t) \simeq \boldsymbol{W}(t-1)\boldsymbol{\Theta}(t), \tag{21}$$

where $\boldsymbol{\Theta}(t) \triangleq \boldsymbol{W}(t-1)^H\boldsymbol{W}(t)$ is an orthonormal matrix. Applying the exponential window estimate and substituting (15) into (19), we obtain

$$\boldsymbol{C}_{xy}(t) = \beta\boldsymbol{C}_{xx}(t-1)\boldsymbol{W}(t-1) + \boldsymbol{x}(t)\boldsymbol{y}(t)^H. \tag{22}$$

Using the approximation (21) of $\boldsymbol{\Theta}(t)$ at time $(t-1)$, Equation (22) can be expressed as

$$\boldsymbol{C}_{xy}(t) = \beta\boldsymbol{C}_{xy}(t-1)\boldsymbol{\Theta}(t-1) + \boldsymbol{x}(t)\boldsymbol{y}(t)^H. \tag{23}$$

Equation (23) is one of the three important formulas that are used in [10]. Other two main iterative expressions, that are needed to estimate the subspace basis matrix $\boldsymbol{W}(t)$ in the FAPI algorithm [10], are given as

$$\boldsymbol{Z}(t) = \frac{1}{\beta}\boldsymbol{\Theta}(t)^H(\boldsymbol{I}_L - \boldsymbol{g}(t)\boldsymbol{y}(t)^H)\boldsymbol{Z}(t-1)\boldsymbol{\Theta}(t)^{-H}, \tag{24}$$

$$\boldsymbol{W}(t) = \left(\boldsymbol{W}(t-1) + \boldsymbol{e}(t)\boldsymbol{g}(t)^H\right)\boldsymbol{\Theta}(t), \tag{25}$$

where $\boldsymbol{Z}(t)$ is an auxiliary matrix, $\boldsymbol{e}(t)$ is projection error of $\boldsymbol{x}(t)$, and $\boldsymbol{g}(t)$ is defined in [10] as

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \boldsymbol{W}(t-1)\boldsymbol{y}(t), \tag{26}$$

$$\boldsymbol{g}(t) = \frac{\boldsymbol{h}(t)}{\beta + \boldsymbol{y}(t)^H\boldsymbol{h}(t)}, \tag{27}$$

where

$$\boldsymbol{h}(t) = \boldsymbol{Z}(t-1)\boldsymbol{y}(t). \tag{28}$$

Proper selection of $\boldsymbol{\Theta}(t)$ in (25) is critical for subspace tracking algorithms. The key point is to make $\boldsymbol{W}(t)$ orthonormal at each iterative step, with low computational complexity and numerically stable way. Namely

$$\boldsymbol{W}(t)^H\boldsymbol{W}(t) = \boldsymbol{I}_L. \tag{29}$$

For the FAPI algorithm, $\boldsymbol{\Theta}(t)$ is selected as an inverse square root as follows,

$$\boldsymbol{\Theta}(t)\boldsymbol{\Theta}(t)^H = (\boldsymbol{I}_L + \boldsymbol{g}(t)(\boldsymbol{e}(t)^H\boldsymbol{e}(t))\boldsymbol{g}(t)^H)^{-1}. \tag{30}$$

However, derivation of $\boldsymbol{\Theta}(t)$ is based on the assumption that $\beta + \boldsymbol{y}(t)^H\boldsymbol{h}(t)$ is nonsingular and solving an inverse square root requires higher computational complexity and some instabilities are inevitable.

Based on the above mentioned discussions, we now present the derivation of the MFAPI algorithm.

D. DERIVATION OF MFAPI

In this paper, we propose another way for selecting $\boldsymbol{\Theta}(t)$ that guarantees orthonormalization of $\boldsymbol{W}(t)$. This approach is based on using a direct matrix production. As a direct results, it provides more simpler derivation than FAPI and avoids any instability in the algorithm.

1) DERIVATION FOR $\theta(t)$

Let

$$\boldsymbol{T}(t) \triangleq \boldsymbol{W}(t-1) + \boldsymbol{e}(t)\boldsymbol{g}(t)^H. \tag{31}$$

It then follows that

$$\boldsymbol{T}(t)^H\boldsymbol{T}(t) = \boldsymbol{I}_L + \|\boldsymbol{e}(t)\|^2\boldsymbol{g}(t)\boldsymbol{g}(t)^H. \tag{32}$$

In deriving (32), we have assumed that $\boldsymbol{W}(t-1)$ is orthonormal at time $(t-1)$, that is, $\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1) = \boldsymbol{I}_L$. Additionally, it can be easily seen by simple manipulations that $\boldsymbol{W}(t-1)^H\boldsymbol{e}(t) = \boldsymbol{0}_{L\times 1}$. Let

$$\boldsymbol{Y}(t) \triangleq \left[\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} \vdots \boldsymbol{G}(t)\right] \tag{33}$$

be a $L \times L$ column orthonormal matrix, satisfying: $\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2} + \boldsymbol{G}(t)\boldsymbol{G}(t)^H = \boldsymbol{I}_L$, $\boldsymbol{G}(t)^H\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} = \boldsymbol{0}_{(L-1)\times 1}$ and $\boldsymbol{G}(t)^H\boldsymbol{G}(t) = \boldsymbol{I}_{L-1}$. It then follows that

$$\begin{aligned}(\boldsymbol{T}(t)\boldsymbol{Y}(t))^H&\boldsymbol{T}(t)\boldsymbol{Y}(t)\\ &= \boldsymbol{Y}(t)^H\boldsymbol{T}(t)^H\boldsymbol{T}(t)\boldsymbol{Y}(t)\\ &= \boldsymbol{I}_L + \mathrm{diag}\left\{\|\boldsymbol{e}(t)\|^2\|\boldsymbol{g}(t)\|^2, 0, \ldots, 0\right\}\end{aligned}$$

$$
\begin{aligned}
&= \mathrm{diag}\left\{1 + \|\boldsymbol{e}(t)\|^2\|\boldsymbol{g}(t)\|^2, 1, \ldots, 1\right\} \\
&= \mathrm{diag}\left\{\|\boldsymbol{v}(t)\|^2, 1, \ldots, 1\right\}, \qquad (34)
\end{aligned}
$$

where $\boldsymbol{v}(t)$is defined as $\boldsymbol{v}(t) = \boldsymbol{W}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} + \boldsymbol{e}(t)\|\boldsymbol{g}(t)\|$, which satisfies

$$
\|\boldsymbol{v}(t)\|^2 = 1 + \|\boldsymbol{e}(t)\|^2\|\boldsymbol{g}(t)\|^2. \qquad (35)
$$

Note that the right hand side of (34) is a diagonal matrix, and can be transformed into an identity matrix by applying a normalization on the first column. A common method for orthonormalization is to multiply it by an inverse matrix, as employed in [10]. With a simple but rather more stable way, we implement normalization here by multiplying a suitable matrix having the following form

$$
\boldsymbol{U}(t) = \mathrm{diag}\left\{\frac{1}{\|\boldsymbol{v}(t)\|}, 1, \ldots, 1\right\}. \qquad (36)
$$

For implementing orthonormalization, $\boldsymbol{\Theta}(t)$ can be selected according to the following relation

$$
\begin{aligned}
\boldsymbol{\Theta}(t) &= \boldsymbol{Y}(t)\boldsymbol{U}(t)\boldsymbol{Y}(t)^H \\
&= \left[\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} \vdots \boldsymbol{G}(t)\right]\boldsymbol{U}(t)\begin{bmatrix}\frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|} \\ \cdots \\ \boldsymbol{G}(t)^H\end{bmatrix} \\
&= \frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{v}(t)\|\,\|\boldsymbol{g}(t)\|^2} + \boldsymbol{G}(t)\boldsymbol{G}(t)^H. \qquad (37)
\end{aligned}
$$

Since $\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2} + \boldsymbol{G}(t)\boldsymbol{G}(t)^H = \boldsymbol{I}_L$ holds, eliminating the term $\boldsymbol{G}(t)\boldsymbol{G}(t)^H$ in (37) yields

$$
\boldsymbol{\Theta}(t) = \boldsymbol{I}_L + (\frac{1}{\|\boldsymbol{v}(t)\|} - 1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}. \qquad (38)
$$

2) UPDATE FOR $Z(t)$ USING $\Theta(t)$

For updating $\boldsymbol{Z}(t)$ in (24), we apply the matrix inversion lemma to $\boldsymbol{\Theta}(t)$ as

$$
\begin{aligned}
\boldsymbol{\Theta}(t)^{-1} &= \left[\boldsymbol{I}_L + \left(\frac{1}{\|\boldsymbol{v}(t)\|} - 1\right)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}\right]^{-1} \\
&= \boldsymbol{I}_L - \frac{\left(\frac{1}{\|\boldsymbol{v}(t)\|} - 1\right)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}}{1 + \left(\frac{1}{\|\boldsymbol{v}(t)\|} - 1\right)\frac{\boldsymbol{g}(t)^H\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|^2}} \\
&= \boldsymbol{I}_L + \frac{(\|\boldsymbol{v}(t)\| - 1)}{\|\boldsymbol{g}(t)\|^2}\boldsymbol{g}(t)\boldsymbol{g}(t)^H \\
&= \boldsymbol{I}_L + \delta(t)\boldsymbol{g}(t)\boldsymbol{g}(t)^H, \qquad (39)
\end{aligned}
$$

where $\delta(t) = \frac{\|\boldsymbol{v}(t)\|-1}{\|\boldsymbol{g}(t)\|^2}$. Substituting (38), (39) into (24) yields

$$
\boldsymbol{Z}(t) = \frac{1}{\beta}[\boldsymbol{Z}(t-1) - \boldsymbol{g}(t)\boldsymbol{h}'(t)^H + \boldsymbol{\epsilon}(t)\boldsymbol{g}(t)^H], \qquad (40)
$$

where $\boldsymbol{h}'(t)$ and $\boldsymbol{\epsilon}(t)$ are defined by

$$
\boldsymbol{y}'(t) = \frac{1}{\|\boldsymbol{v}(t)\|}\left[\delta(t)\boldsymbol{g}(t) + \boldsymbol{y}(t)\right], \qquad (41)
$$

$$
\boldsymbol{h}'(t) = \boldsymbol{Z}(t-1)^H\boldsymbol{y}'(t), \qquad (42)
$$

$$
\boldsymbol{\epsilon}(t) = \delta(t)\left[\boldsymbol{Z}(t-1)\boldsymbol{g}(t) - \boldsymbol{g}(t)\boldsymbol{h}'(t)^H\boldsymbol{g}(t)\right]. \qquad (43)
$$

3) UPDATE FOR $W(t)$ USING $\Theta(t)$

Orthonormalization of $\boldsymbol{W}(t)$ needs to be implemented exactly at each iterative step. Using $\boldsymbol{\Theta}(t)$ in (38), $\boldsymbol{W}(t)$ in (25) can be computed as follows. Substituting (38) into (25) yields

$$
\begin{aligned}
\boldsymbol{W}(t) &= \boldsymbol{W}(t-1) + \frac{1}{\|\boldsymbol{v}(t)\|}\boldsymbol{e}(t)\boldsymbol{g}(t)^H \\
&\quad + (\frac{1}{\|\boldsymbol{v}(t)\|} - 1)\boldsymbol{W}(t-1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2} \\
&= \boldsymbol{W}(t-1) + \boldsymbol{e}'(t)\boldsymbol{g}(t)^H, \qquad (44)
\end{aligned}
$$

where

$$
\boldsymbol{e}'(t) = \frac{1}{\|\boldsymbol{v}(t)\|}\boldsymbol{e}(t) + (\frac{1}{\|\boldsymbol{v}(t)\|} - 1)\frac{\boldsymbol{W}(t-1)\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|^2}. \qquad (45)
$$

Substituting (41) into (45) yields

$$
\boldsymbol{e}'(t) = \frac{1}{\|\boldsymbol{v}(t)\|}\boldsymbol{x}(t) - \boldsymbol{W}(t-1)\boldsymbol{y}'(t). \qquad (46)
$$

After some algebra, it can be shown that

$$
\begin{aligned}
\boldsymbol{W}(t)^H\boldsymbol{W}(t) &= \boldsymbol{\Theta}(t)^H\boldsymbol{T}(t)^H\boldsymbol{T}(t)\boldsymbol{\Theta}(t) \\
&= \boldsymbol{Y}(t)\boldsymbol{U}(t)^H\boldsymbol{Y}(t)^H\boldsymbol{T}(t)^H\boldsymbol{T}(t)\boldsymbol{Y}(t)\boldsymbol{U}(t)\boldsymbol{Y}(t)^H \\
&= \boldsymbol{I}_L. \qquad (47)
\end{aligned}
$$

TABLE 1. Pseudo-code of the MFAPI algorithm.

Initialization:*		
$\boldsymbol{W}(0) = \begin{bmatrix}\boldsymbol{I}_L \\ \boldsymbol{0}_{(N-L)\times L}\end{bmatrix}, \boldsymbol{Z}(0) = \boldsymbol{I}_L$		
For each iteration, do		
Input sequential data vector:$\boldsymbol{x}(t)$		Cost
$\boldsymbol{y}(t) = \boldsymbol{W}(t-1)^H\boldsymbol{x}(t)$	(18)	NL
$\boldsymbol{h}(t) = \boldsymbol{Z}(t-1)\boldsymbol{y}(t)$	(28)	L^2
$\boldsymbol{g}(t) = \frac{\boldsymbol{h}(t)}{\beta+\boldsymbol{y}(t)^H\boldsymbol{h}(t)}$	(27)	$2L$
$\|\boldsymbol{e}(t)\|^2 = \|\boldsymbol{x}(t)\|^2 - \|\boldsymbol{y}(t)\|^2$	(26)	$N+L$
$**\ \|\boldsymbol{v}(t)\| = \sqrt{1 + \|\boldsymbol{e}(t)\|^2\|g(t)\|^2}$		L
$\delta(t) = \frac{\|\boldsymbol{v}(t)\|-1}{\|g(t)\|^2}$		1
$\boldsymbol{y}'(t) = \frac{1}{\|\boldsymbol{v}(t)\|}\left[\delta(t)\boldsymbol{g}(t) + \boldsymbol{y}(t)\right]$	(41)	L
$\boldsymbol{h}'(t) = \boldsymbol{Z}(t-1)^H\boldsymbol{y}'(t)$	(42)	L^2
$\boldsymbol{\epsilon}(t) = \delta(t)\left[\boldsymbol{Z}(t-1)\boldsymbol{g}(t) - \boldsymbol{g}(t)\boldsymbol{h}'(t)^H\boldsymbol{g}(t)\right]$	(43)	L^2+3L
$\boldsymbol{Z}(t) = \frac{1}{\beta}\left[\boldsymbol{Z}(t-1) - \boldsymbol{g}(t)\boldsymbol{h}'(t)^H + \boldsymbol{\epsilon}(t)\boldsymbol{g}(t)^H\right]$	(40)	$2L^2$
$\boldsymbol{e}'(t) = \frac{1}{\|\boldsymbol{v}(t)\|}\boldsymbol{x}(t) - \boldsymbol{W}(t-1)\boldsymbol{y}'(t)$	(46)	$NL+N$
$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) + \boldsymbol{e}'(t)\boldsymbol{g}(t)^H$	(44)	NL

* The values of $\boldsymbol{W}(0)$ and $\boldsymbol{Z}(0)$ should be assigned properly in order to have a fast convergence rate. In general,$\boldsymbol{W}(0)$ should be orthonormal and $\boldsymbol{Z}(0)$ be positive definite.

** The step without equation number can be found in the paper.

IV. PSEUDO-CODE AND PERFORMANCE ANALYSIS

The computational cost for each of the detailed pseudo-code of the MFAPI is presented in Table 1. As depicted in Table 1, the computational complexity of MFAPI is approximately $(3NL + 2N)$. Hence, it is in the low-complexity class of algorithms that requires $O(NL)$ operations. Compared to the computational costs of the FAPI in [10, Table 3], the MFAPI algorithm has the similar computation complexity to the FAPI.

In summary, we emphasize that, from the implementation point of view, the MFAPI is simpler and more stable than FAPI. This is mainly because that the two algorithms realize orthonormalization of $\boldsymbol{W}(t)$ in different ways.

A. CONVERGENCE RATE ANALYSIS

As mentioned in the introduction section, almost all subspace tracking algorithms can be categorized into two types based on the way of their subspace tracking techniques are implemented. In this section, a comparison on the convergence rate of the both types is presented. The first kind of subspace tracking algorithms, that can only track single subspace basis, have similar form of the power iteration algorithm as mentioned in Sect. III-B. The second kind of methods, which can simultaneously track dual subspace basis matrix with a simple sign change, can be expressed as:

$$\boldsymbol{y}(t) = \boldsymbol{W}(t-1)^H \boldsymbol{x}(t), \tag{48}$$

$$\boldsymbol{M}(t) = \boldsymbol{W}(t-1) \pm \mu \boldsymbol{x}(t)\boldsymbol{y}(t)^H, \tag{49}$$

$$\boldsymbol{W}(t) = \text{orthonorm}\{\boldsymbol{M}(t)\}, \tag{50}$$

where "+" represents that the algorithm can track signal subspace while "−" represents tracking noise subspace. μ is the step size with small value for numerical stability and orthonorm$\{\cdot\}$ denotes the orthonormalization of $\boldsymbol{M}(t)$.

Each kind of subspace tracking algorithms has a different exponential convergence rate at the transient-state step. Here we do not analyze the performance of algorithms at the steady-state step and recommend the interested readers to read [23] for more detail information. As given in [7], the exponential convergence rates of the first and second types of subspace tracking methods can be expressed as $(\lambda_{L+1}/\lambda_L)^n$ and $((1+\mu\lambda_{L+1})/(1+\mu\lambda_L))^n$, respectively, where $\lambda_L > \lambda_{L+1}$. When the algorithm converges to a steady-state point, we have $\lim_{n\to\infty}(\lambda_{L+1}/\lambda_L)^n = 0$ or $\lim_{n\to\infty}(((1+\mu\lambda_{L+1})/(1+\mu\lambda_L))^n = 0$. Since $\mu > 0$ is very small and $0 \leq \lambda_{L+1} < \lambda_L$, we conclude easily that the first kind has a faster convergence rate than the second kind. The results presented in the simulation section also agree with this analysis.

B. NUMERICAL STABILITY ANALYSIS

In this section, we analyze the numerical stability of MFAPI. We particularly focus on the deviations between the product $\boldsymbol{W}(t)^H\boldsymbol{W}(t)$ and the identity matrix $\boldsymbol{I}_L$. Similar analysis is also given in [17]. Now, assume that $\boldsymbol{W}(t)^H\boldsymbol{W}(t) = \boldsymbol{I}_L + \boldsymbol{\xi}(t)$, where $\boldsymbol{\xi}(t)$ is difference from $\boldsymbol{I}_L$. Then, by means of $\boldsymbol{W}(t)^H\boldsymbol{e}(t) = \boldsymbol{0}$ and $\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1) = \boldsymbol{I}_L + \boldsymbol{\xi}(t-1)$, it follows that (51), as shown at the bottom of this page. Detail derivations can be found in Appendix A.

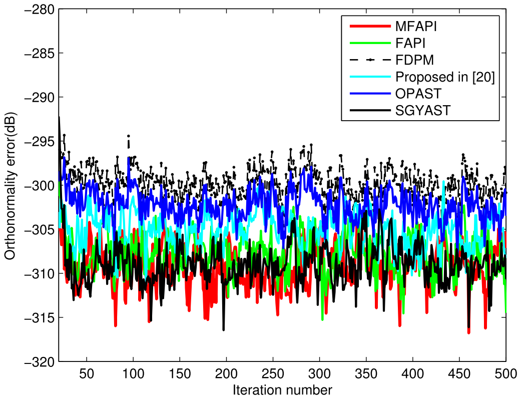

FIGURE 1. Orthonormality error analysis of MFAPI with other algorithms.

Assuming $\boldsymbol{\xi}(t)$ is small at steady-state step (as seen in Fig. 1, the value of $\|\boldsymbol{\xi}(t)\|_F^2$ is approximately 10^{-30}), it follows that $\|\boldsymbol{v}(t)\|^2 + \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} \approx \|\boldsymbol{v}(t)\|^2$. Consequently,

$$\boldsymbol{Y}(t)^H\boldsymbol{T}(t)^H\boldsymbol{T}(t)\boldsymbol{Y}(t) = \begin{bmatrix} \|\boldsymbol{v}(t)\|^2 & \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{\xi}(t-1)\boldsymbol{G}(t) \\ \boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} & \boldsymbol{I}_{L-1} + \boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\boldsymbol{G}(t) \end{bmatrix}. \tag{52}$$

Equations (36), (47) and (52) yield

$$\begin{aligned} \boldsymbol{W}(t)^H\boldsymbol{W}(t) = & \frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2} \\ & + \frac{1}{\|\boldsymbol{v}(t)\|}\boldsymbol{G}(t)\boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2} \\ & + \frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{v}(t)\|\|\boldsymbol{g}(t)\|^2}\boldsymbol{\xi}(t-1)\boldsymbol{G}(t)\boldsymbol{G}(t)^H \\ & + \boldsymbol{G}(t)\boldsymbol{G}(t)^H \\ & + \boldsymbol{G}(t)\boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\boldsymbol{G}(t)\boldsymbol{G}(t)^H. \end{aligned} \tag{53}$$

Replacing $\boldsymbol{G}(t)\boldsymbol{G}(t)^H = \boldsymbol{I}_L - \frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}$, (53) can be rewritten as

$$\begin{aligned} \boldsymbol{W}(t)^H\boldsymbol{W}(t) = \boldsymbol{I}_L + \boldsymbol{\xi}(t-1) & \\ & + (1 - \frac{2}{\|\boldsymbol{v}(t)\|})\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2} \\ & + (\frac{1}{\|\boldsymbol{v}(t)\|} - 1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}\boldsymbol{\xi}(t-1) \\ & + (\frac{1}{\|\boldsymbol{v}(t)\|} - 1)\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}. \end{aligned} \tag{54}$$

$$\boldsymbol{Y}(t)^H\boldsymbol{T}(t)^H\boldsymbol{T}(t)\boldsymbol{Y}(t) = \begin{bmatrix} \|\boldsymbol{v}(t)\|^2 + \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} & \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{\xi}(t-1)\boldsymbol{G}(t) \\ \boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} & \boldsymbol{I}_{L-1} + \boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\boldsymbol{G}(t) \end{bmatrix}. \tag{51}$$

Substituting $\boldsymbol{W}(t)^H\boldsymbol{W}(t) = \boldsymbol{I}_L + \boldsymbol{\xi}(t)$ into (54) yields

$$\begin{aligned}\boldsymbol{\xi}(t) &= \boldsymbol{\xi}(t-1)\\ &\quad+(1-\frac{2}{\|\boldsymbol{v}(t)\|})\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}\\ &\quad+(\frac{1}{\|\boldsymbol{v}(t)\|}-1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}\boldsymbol{\xi}(t-1)\\ &\quad+(\frac{1}{\|\boldsymbol{v}(t)\|}-1)\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}. \qquad (55)\end{aligned}$$

With reference to the similar derivation in [17], we discuss the numerical stability in two cases in the presence of different SNR values:

Case 1: In high SNR region, since all the noise eigenvalues of sample covariance matrix are sufficiently small, $\|\boldsymbol{e}(t)\|^2$ becomes sufficiently small (according to Karhunen-Loève transform (KLT), the error of projection $\|\boldsymbol{e}(t)\|^2 = \sum_{k=L+1}^{N}\lambda_k$ is used). From Equation (35) then it follows that $\|\boldsymbol{v}(t)\| \approx 1$ and hence (55) can be rewritten as

$$\boldsymbol{\xi}(t) = \boldsymbol{\xi}(t-1) - \frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2} + \boldsymbol{\Gamma}(t), \qquad (56)$$

where $\boldsymbol{\Gamma}(t)$ is the numerical precision error and here we assume $\boldsymbol{\Gamma}(t) = \boldsymbol{0}$. Applying the column-wise version $\mathrm{vec}\{\boldsymbol{\xi}(t)\}$ of $\boldsymbol{\xi}(t)$, (56) takes the form

$$\mathrm{vec}\{\boldsymbol{\xi}(t)\} = \boldsymbol{\Omega}\mathrm{vec}\{\boldsymbol{\xi}(t-1)\}, \qquad (57)$$

where $\boldsymbol{\Omega} = \boldsymbol{I}_{L^2} - \frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2} \otimes \frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}$ and $\otimes$ denotes the Kronecker product. In order to achieve the numerical stability, we need $\lim_{t\to\infty}\mathrm{vec}\{\boldsymbol{\xi}(t)\} = 0$ in (57). In this case all eigenvalues of matrix $\boldsymbol{\Omega}$ should be in the unit circle. Since $\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}$ is positive definite and $\mathrm{tr}\{\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}\} = 1$, the eigenvalues, $\gamma_i(i = 1, 2, ..., L)$, of $\frac{\boldsymbol{g}(t)\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|^2}$ satisfy $0 < \gamma_i < 1(i = 1, 2, ..., L)$. Meanwhile, by using properties of the Kronecker product eigenvalues, we can show the eigenvalues of $\boldsymbol{\Omega}$ satisfying

$$0 < (1-\gamma_i\gamma_j) < 1, \quad i, j = 1, 2, ..., L. \qquad (58)$$

Case 2: In low SNR region, $\|\boldsymbol{e}(t)\|^2$ is not sufficiently small. However, according to Fig. 1, $\|\boldsymbol{\xi}(t)\|_F^2$ is small enough to be taken approximately equal to zero. Hence, the resulting derivation becomes very simple by directly taking $\boldsymbol{\xi}(t-1) = \boldsymbol{0}$. In this case, (55) can be written as $\boldsymbol{\xi}(t) = \boldsymbol{\xi}(t-1) \approx \boldsymbol{0}$.

Consequently, from the above discussions, we conclude that the MFAPI is numerically stable.

V. SIMULATION RESULTS

In this section, computer simulation results are presented to test the performance of the proposed MFAPI algorithm. We note that the subspace tracking algorithms are extremely rich in the literature and hence it is difficult to assess and compare the performances of all the algorithms. In order to have a relatively complete comparison, five well-known algorithms (FAPI, OPAST, FDPM, SGYAST, and algorithm proposed in [20]) are used for comparison. Among those algorithms, FAPI and OPAST belong to the first type while FDPM and the algorithm in [20] belong to the second type (see the introduction section). As a stable version of GYAST, SGYAST, is included since this method is newer in the literature despite its higher computational complexity $O(6NL)$ than that of other algorithms $O(3NL)$ (here we do not consider GYAST as a test bench due to its numerical instability, as discussed in [19]). We compare the performance of these algorithms from two aspects: orthonormality error and convergence rate.

A. ORTHONORMALITY ANALYSIS

Performance of subspace tracking methods can be measured in part by orthonormality error of $\boldsymbol{W}(t)$ and thus the following criteria is commonly used to estimate this error.

$$10\log_{10}\|\boldsymbol{W}(t)^H\boldsymbol{W}(t) - \boldsymbol{I}_L\|_F^2. \qquad (59)$$

Fig. 1 shows the orthonormality errors of OPAST, FAPI, MFAPI, SGYAST, FDPM, and algorithm proposed in [20]. We observe that these algorithms all have superior performance for orthonormality and the value reduces to about −300 dB.

TABLE 2. Parameters used in subspace tracking methods.

Parameters	Value	Algorithms
Frequencies*	0.1Hz,0.2Hz,0.3Hz	all
Sample Number	80	all
SNR	15dB	all
β	0.87	FAPI, MFAPI, OPAST,SGYAST
μ	0.009	FDPM, Proposed in [20]

*The abrupt frequency change of 0.3Hz is at iteration point 100 and 0.2Hz at iteration point 400, as shown in Fig.2.

B. CONVERGENCE RATE ANALYSIS

In this section, to test the convergence rates of the MFAPI and FAPI, we consider estimation of the frequencies of signals embedded in additive noise. We assume that frequencies of three sinusoidal signals are estimated with the rank of signal subspace basis matrix to be $L = 3$. Additionally, in order to improve the performance of the algorithm, Root-MUSIC method is also applied to the estimated subspace basis matrix [24]. The SNR is chosen approximately 15dB as in [17]. More detail configurations are also shown in Table 2. In order to observe the convergence ability during sudden steep rises or drops in frequencies, we set them change abruptly at iteration points 100 and 400. Since MFAPI is a modified version of FAPI, we compare the behavior of these two algorithms first. As seen from the signal tracking performance of MFAPI and FAPI in Fig. 2, both of these two algorithms having capability to track the frequencies of signals, even in the presence of abrupt situations. As Fig. 2 provides an overall description, they have similar performances in the

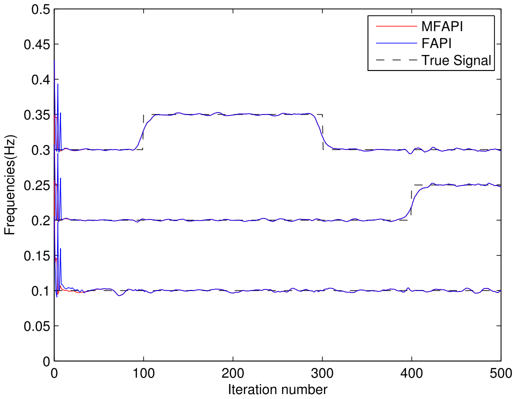

FIGURE 2. Tracking three signals with MFAPI and FAPI algorithm.

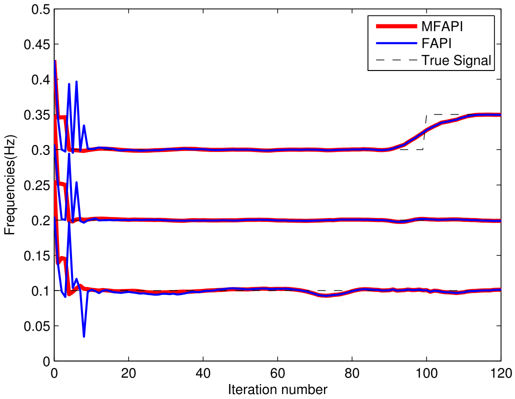

FIGURE 3. Tracking three signals with MFAPI and FAPI for local enlargement.

context of signal detection and tracking. However, a more clear differences between these algorithms can be observed by a local enlargement, as seen in Fig. 3, where the MFAPI has a faster convergence rate than FAPI at the transient-state stage.

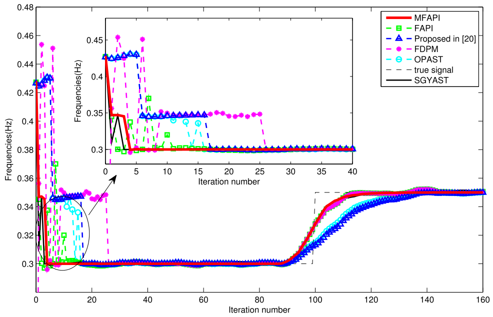

FIGURE 4. Tracking one signal with different algorithms.

Unlike Fig. 3, plotting the tracking curves of all the three frequencies, Fig. 4 provides only a signal with frequency 0.3 Hz for explicit comparison. The MFAPI and other five algorithms (FAPI, OPAST, FDPM, SGYAST, and algorithm proposed in [20]) are chosen as test benches and it is evident that the proposed MFAPI method has a faster convergence rate than all other algorithms at transient-state steps except SGYAST, as depicted in Fig. 4 (Please note although SGYAST has slightly better performance than MFAPI, SGYAST has higher computation complexity). The results clearly verify the conclusion of the analysis, presented in Section IV.

VI. CONCLUSION

In this paper, a fast and stable subspace tracking algorithm with low complexity is provided. The proposed algorithm has the similar computational complexity as FAPI. Instead of solving the inverse square root of matrix as employed in FAPI, the proposed algorithm applies direct matrix product to ensure the orthonormality of subspace basis matrix at each recursion, that has more simpler derivation. Additionally, the new algorithm is proven to be numerically stable as shown by the detail mathematical inference. Finally, regarding the convergence rate at transient-state step, our improved

$$\begin{aligned}\boldsymbol{Y}(t)^H\boldsymbol{T}(t)^H\boldsymbol{T}(t)\boldsymbol{Y}(t) &= \begin{bmatrix}\frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1)+\|\boldsymbol{g}(t)\|\boldsymbol{e}^H(t)\boldsymbol{e}(t)\boldsymbol{g}(t)^H\\ \boldsymbol{G}(t)^H\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1)\end{bmatrix}\begin{bmatrix}\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|}\vdots\boldsymbol{G}(t)\end{bmatrix}\\ &= \begin{bmatrix}\frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|}+\|\boldsymbol{g}(t)\|^2\|\boldsymbol{e}(t)\|^2 & \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1)\boldsymbol{G}(t)\\ \boldsymbol{G}(t)^H\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1)\frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|} & \boldsymbol{G}(t)^H\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1)\boldsymbol{G}(t)\end{bmatrix}\end{aligned}\tag{62}$$

$$\begin{aligned}\boldsymbol{Y}(t)^H\boldsymbol{T}(t)^H\boldsymbol{T}(t)\boldsymbol{Y}(t) &= \begin{bmatrix}\frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|}+1+\|\boldsymbol{g}(t)\|^2\|\boldsymbol{e}(t)\|^2 & \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{\xi}(t-1)\boldsymbol{G}(t)\\ \boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} & \boldsymbol{I}_{L-1}+\boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\boldsymbol{G}(t)\end{bmatrix}\\ &= \begin{bmatrix}\|\boldsymbol{v}(t)\|^2+\frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{\xi}(t-1)\frac{\boldsymbol{g}(t)}{\|\boldsymbol{g}(t)\|} & \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{\xi}(t-1)\boldsymbol{G}(t)\\ \boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\frac{g(t)}{\|\boldsymbol{g}(t)\|} & \boldsymbol{I}_{L-1}+\boldsymbol{G}(t)^H\boldsymbol{\xi}(t-1)\boldsymbol{G}(t)\end{bmatrix}\end{aligned}\tag{63}$$

algorithm yields the fastest convergence rate among existing subspace tracking methods with computation complexity $O(3NL)$ as verified by numerical simulations.

APPENDIX I. DERIVATION OF EQUATION (51)

Equations (31), (33) yield

$$\begin{aligned}
&\boldsymbol{Y}(t)^H\boldsymbol{T}(t)^H \\
&= \begin{bmatrix} \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|} \\ \cdots \\ \boldsymbol{G}(t)^H \end{bmatrix} \left[\boldsymbol{W}(t-1)^H + \boldsymbol{g}(t)\boldsymbol{e}(t)^H\right] \\
&= \begin{bmatrix} \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{W}(t-1)^H + \|\boldsymbol{g}(t)\|\boldsymbol{e}(t)^H \\ \boldsymbol{G}(t)^H\boldsymbol{W}(t-1)^H \end{bmatrix}
\end{aligned} \tag{60}$$

Equation (60) post-multiplied by $\boldsymbol{T}(t)$ yields

$$\begin{aligned}
&\boldsymbol{Y}(t)^H\boldsymbol{T}(t)^H\boldsymbol{T}(t) \\
&= \begin{bmatrix} \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{W}(t-1)^H + \|\boldsymbol{g}(t)\|\boldsymbol{e}(t)^H \\ \boldsymbol{G}(t)^H\boldsymbol{W}(t-1)^H \end{bmatrix} \left[\boldsymbol{W}(t-1) + \boldsymbol{e}(t)\boldsymbol{g}(t)^H\right] \\
&= \begin{bmatrix} \frac{\boldsymbol{g}(t)^H}{\|\boldsymbol{g}(t)\|}\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1) + \|\boldsymbol{g}(t)\|\boldsymbol{e}(t)^H\boldsymbol{e}(t)\boldsymbol{g}(t)^H \\ \boldsymbol{G}(t)^H\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1) \end{bmatrix}
\end{aligned} \tag{61}$$

The next step derivation, Equation (62), can be found at the bottom of the previous page.

By using $\boldsymbol{W}(t-1)^H\boldsymbol{W}(t-1) = \boldsymbol{I}_L + \boldsymbol{\xi}(t-1)$ we obtain (63), as shown at the bottom of the previous page.

REFERENCES

[1] Y. Chi and H. Fu, "Subspace learning from bits," *IEEE Trans. Signal Process.*, vol. 65, no. 17, pp. 4429–4442, Sep. 2017.

[2] V. Golikov and O. Lebedeva, "A robust detection in the presence of clutter and jammer signals with unknown powers," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E94-A, no. 2, pp. 817–822, 2011.

[3] Y. Trachi, E. Elbouchikhi, V. Choqueuse, and M. E. H. Benbouzid, "Induction machines fault detection based on subspace spectral estimation," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5641–5651, Sep. 2016.

[4] A. Szumski, "Finding the interference—The Karhunen–Loève transform as an instrument to detect weak RF signals," *Inside GNSS*, vol. 6, no. 3, pp. 56–64, May/Jun. 2011.

[5] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philos. Trans. Roy. Soc. A, Math. Phys. Eng. Sci.*, vol. 347, no. 2065, p. 20150202, Mar. 2011.

[6] V.-D. Nguyen, K. Abed-Meraim, N. Linh-Trung, and R. Weber, "Generalized minimum noise subspace for array processing," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3789–3802, Jul. 2017.

[7] G. H. Golub and C. F. V. Loan, *Matrix Computation*, 4th ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2013.

[8] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.

[9] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Signal Process. Lett.*, vol. 7, no. 3, pp. 60–62, Mar. 2000.

[10] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2931–2941, Aug. 2005.

[11] T. D. Nguyen and I. Yamada, "A unified convergence analysis of normalized PAST algorithms for estimating principal and minor components," *Signal Process.*, vol. 93, no. 1, pp. 176–184, Jan. 2013.

[12] S. Attallah and K. Abed-Meraim, "Low-cost adaptive algorithm for noise subspace estimation," *Electron. Lett.*, vol. 38, no. 12, pp. 609–611, Jun. 2002.

[13] S. Attallah, "The generalized Rayleigh's quotient adaptive noise subspace algorithm: A householder transformation-based implementation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 1, pp. 3–7, Jan. 2006.

[14] L. Yang, S. Attallah, G. Mathew, and K. Abed-Meraim, "Analysis of orthogonality error propagation for FRANS and HFRANS algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 9, pp. 4515–4521, Sep. 2008.

[15] J.-F. Yang and M. Kaveh, "Adaptive eigensubspace algorithms for direction or frequency estimation and tracking," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-36, no. 2, pp. 241–251, Feb. 1988.

[16] S. C. Douglas, "Numerically-robust adaptive subspace tracking using householder transformations," in *Proc. IEEE Workshop Sensor Array Multichannel Signal Process.*, Mar. 2000, pp. 499–503.

[17] X. G. Doukopoulos and G. V. Moustakides, "Fast and stable subspace tracking," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1452–1465, Apr. 2008.

[18] R. Badeau, G. Richard, and B. David, "Fast and stable YAST algorithm for principal and minor subspace tracking," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3437–3446, Aug. 2008.

[19] M. Arjomandi-Lari and M. Karimi, "Generalized YAST algorithm for signal subspace tracking," *Signal Process.*, vol. 117, pp. 82–95, Dec. 2015.

[20] 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.

[21] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.

[22] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[23] R. Haddadi, "Steady-state statistical performance analysis of subspace tracking methods," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4781–4791, Sep. 2016.

[24] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 8, Sep. 1983, pp. 336–339.

QIANG WU received the M.S. degree from the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, in 2006. He is currently pursuing the Ph.D. degree with the School of Electronic Information, Wuhan University.

Since 2014, he has been an Associate Professor with the School of Engineering, Tibet University, Lhasa. His research interests include signal processing and adaptation technology for wireless communication.

JIAN-SHENG ZHENG received the B.E., M.S., and Ph.D. degrees from the School of Electronic Information, Wuhan University, Wuhan, China, in 1982, 1999, and 2006, respectively.

Since 2000, he has been a Professor with the School of Electronic Information, Wuhan University. His research interests include satellite navigation and positioning, and wireless communication.

ZHI-CHENG DONG (S'12–M'16) was born in Langzhong, China, in 1982. He received the B.E., M.S., and Ph.D. degrees from the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, in 2004, 2008, and 2016, respectively.

Since 2016, he has been an Associate Professor with the School of Engineering, Tibet University, Lhasa, China. He has served as a Technical Program Committee Member for Chinacom (2014), IEEE ICC (2015–2017), TENSYMP (2015), and IEEE GLOBECOM 2017.

His research interests include adaptation technology, performance analysis, and signal processing for high mobility wireless communication.

ERDAL-PANAYIRCI (M'80–SM'91–F'03–LF'06) received the Diploma Engineering degree in electrical engineering from Istanbul Technical University, Istanbul, Turkey, and the Ph.D. degree in electrical engineering and system science from Michigan State University, USA. Until 1998, he was with the Faculty of Electrical and Electronics Engineering, Istanbul Technical University, where he was a Professor and the Head of the Telecommunications Chair. He is currently a Professor of electrical engineering and the Head of the Electronics Engineering Department, Kadir Has University, Istanbul. He has published extensively in leading scientific journals and international conference and co-authored the book *Principles of Integrated Maritime Surveillance Systems* (Boston: Kluwer Academic Publishers, 2000). His recent research interests include communication theory, synchronization, applications of the advanced signal processing techniques to electrical, and optical and underwater acoustic wireless communication.

He spent the academic year 2008–2009 at the Department of Electrical Engineering, Princeton University, NY, USA, involved in new channel estimation and equalization algorithms for high mobility WIMAX and LTE systems. He has been the Principal Coordinator of a 6th and 7th Frame European Project called Network of Excellent on Wireless Communications representing Kadir Has University for five years and WIMAGIC Strep project for two years. He served as a member of the IEEE Fellow Committee in 2005–2008. He was the Technical Program Co-Chair of the IEEE International Conference on Communications, Istanbul, in 2006, and the Technical Program Chair of the IEEE PIMRC, Istanbul, in 2010. He is the Executive Vice Chairman of the IEEE Wireless Communications and Networking Conference, Istanbul, in 2014. He is currently the Head of the Turkish Scientific Commission on Signals and Systems of the International Union of Radio Science. He was an Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS in the areas of synchronization and equalizations in 1995–1999.

ZHI-QIANG WU (M'02–SM'16) received the B.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 1993, the M.S. degree from Peking University, Beijing, in 1996, and the Ph.D. degree from Colorado State University, Fort Collins, in 2002, all in electrical engineering. He was an Assistant Professor with the Department of Electrical Engineering, West Virginia University Institute of Technology, from 2003 to 2005. He joined the Department of Electrical Engineering, Wright State University, in 2005, where he currently serves as a Full Professor.

REN QINGNUOBU was born in 1971. He received the M.S. degree in Tibetan information processing from Tibet University in 2008. He is currently an Associate Professor with Tibet University. His research interests include image inpainting, Tibetan information processing technology, and signal processing algorithms.