Plane Rotation-Based EVD Updating Schemes for Efficient Subspace Tracking

Benoît Champagne and Qing-Guang Liu

***Abstract*— We present new algorithms based on plane rotations for tracking the eigenvalue decomposition (EVD) of a time-varying data covariance matrix. These algorithms directly produce eigenvectors in orthonormal form and are well suited for the application of subspace methods to nonstationary data. After recasting EVD tracking as a simplified rank-one EVD update problem, computationally efficient solutions are obtained in two steps. First, a new kind of *parametric* perturbation approach is used to express the eigenvector update as an unimodular orthogonal transform, which is represented in exponential matrix form in terms of a reduced set of small, unconstrained parameters. Second, two approximate decompositions of this exponential matrix into products of plane (or Givens) rotations are derived, one of which being previously unknown. These decompositions lead to new plane rotation-based EVD-updating schemes (PROTEUS), whose main feature is the use of plane rotations for updating the eigenvectors, thereby preserving orthonormality. Finally, the PROTEUS schemes are used to derive new EVD trackers whose convergence and numerical stability are investigated via simulations. One algorithm can track all the signal subspace EVD components in only $O(LM)$ operations, where L and M, respectively, denote the data vector and signal subspace dimensions while achieving a performance comparable to an exact EVD approach *and* maintaining perfect orthonormality of the eigenvectors. The new algorithms show no signs of error buildup.**

I. Introduction

SUBSPACE-BASED signal analysis methods play a major role in contemporary signal processing, with applications including direction-of-arrival estimation in array processing and frequency estimation of sinusoidal signals in spectral analysis. As their distinguishing feature, these methods seek to extract the desired information about the signal and noise by first estimating either a part or all of the eigenvalue decomposition (EVD) of the data covariance matrix. For example, knowledge of the eigenvalues can be used in connection with a criterion such as AIC or MDL to estimate the number of dominant signal sources present in the observed data [48]. Additional knowledge of the eigenvectors can be used in a high-resolution procedure such as MUSIC to estimate unknown parameters of these dominant sources [3], [40].

Manuscript received August 29, 1996; revised November 17, 1997. This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ananthram Swami.

B. Champagne is with INRS-Télécommunications, Université du Québec, Verdun, P.Q., Canada H3E 1H6 (e-mail: champagne@inrs-telecom.uquebec.ca).

Q.-G. Liu is with Nortel Inc., Verdun, P.Q., Canada.

Publisher Item Identifier S 1053-587X(98)04423-7.

In on-line applications of subspace-based methods to nonstationary signals, it is desirable to continually update the EVD estimates in real time as new data vectors become available in order to permit the detection of abrupt changes and/or the tracking of nonstationarities in the signal environment. In recent years, several computationally efficient techniques in the form of recursive algorithms have thus been proposed for sequential estimation and tracking of some, or all, of the EVD components of a time-varying data covariance matrix. These algorithms, which are collectively referred to here as subspace trackers, rely on different approaches for their derivations and, accordingly, may differ considerably in terms of complexity and performance.

A commonly used approach for the derivation of subspace trackers is to formulate the determination of the desired EVD components as the optimization (possibly constrained) of a specific cost function involving the unknown data covariance matrix. To arrive at a recursive algorithm, the optimization is accomplished adaptively via an appropriate stochastic search procedure. Algorithms of this type have been derived based on the constrained gradient search [16], [20], [33], [37], [46], [52], the conjugate gradient iteration [10], [20], [19], [43], the Gauss-Newton search [27], [28], [36], and the recursive least-squares [51]. Another type of approach consists of using classical algorithms from numerical analysis to compute exactly, at regular intervals, the EVD of a time-varying sample covariance matrix or, equivalently, the singular value decomposition (SVD) of a corresponding data matrix. Such a technique based on orthogonal iterations is proposed in [33]. Within this framework, specific attempts have been made to exploit the low-rank nature of practical recursive covariance matrix estimates. In [23], a classical algorithm for SVD computation [21] is tailored to the rank-one update situation; noise eigenvalue smoothing (i.e., noise subspace sphericalization) is also introduced to reduce the computational load and the estimation variance. In [41], a similar solution is proposed based on an algorithm for exact rank-one updating of the EVD [5]. Further related contributions include numerical stabilization techniques [12], [30], extension to rank-k updates $(k > 1)$ [53], SVD and forward-backward versions [32], and joint signal and noise subspaces sphericalization [13], [14]. Another, closely related approach consists of interlacing the recursive update of a sample covariance or data matrix with only a few steps of certain standard iterations for EVD or SVD computation so that the desired EVD information is only updated approximately. Subspace trackers of this type have been derived based on the inverse power method [47], the

1053–587X/98$10.00 © 1998 IEEE

orthogonal iteration [24], [45], the QR iteration [17], [39], the Lanczos method [11], [49], and the Jacobi-type SVD iterations [18], [25], [31]. Specific algorithms have also been developed that track only basis vectors of the signal and/or noise subspaces based on the URV decomposition [35], [44], the rank-revealing QR factorization [4], and classical invariant subspace updating techniques [26].

Recently, efficient EVD trackers have been obtained based on the application of classical perturbation methods to the rank-one update for the sample covariance matrix [6]. One of these algorithms (PC) can track the M-dimensional signal subspace (eigenvalues and eigenvectors) of an L-dimensional data vector in only $O(LM)$ operations while achieving a performance comparable with an exact EVD approach. The main limitation of this approach, which is common to many of the above subspace trackers, is its inability to directly produce perfectly orthonormal eigenvector estimates. Yet, several of the most popular subspace-based signal analysis methods assume or require the use of an orthonormal basis. For instance, random deviations from orthonormality ultimately limits the resolution capability and estimation accuracy of the root-MUSIC method [1]. Some form of orthonormality is also important in maintaining long-term numerical stability of the perturbation algorithms of [6] as well as of other subspace trackers [12], [31]. In all these situations, further orthonormalization of the dominant eigenvectors is necessary. This entails additional computational costs of $O(LM^2)$ that set a lower bound on reachable operation counts.

In this paper, we derive and evaluate new EVD tracking algorithms that overcome this limitation by directly producing eigenvector estimates that are orthonormal at all times. The derivation involves two steps: First, a new kind of improved, *parametric* perturbation approach is used to express the eigenvector update as a unimodular orthogonal transformation that is represented in exponential matrix form in terms of a reduced set of small, unconstrained parameters. Second, two approximate decompositions of this exponential matrix into products of plane (or Givens) rotations are derived, one of which being previously unknown. These decompositions then lead to new plane rotation-based EVD-updating schemes (PROTEUS) for the rank-one problem. These schemes, which are noniterative (i.e., close form), use specific sequences of plane rotations to update the eigenvectors, thereby preserving orthonormality. In the paper, the PROTEUS schemes are used to derive various subspace trackers whose convergence and numerical stability are investigated via computer experiments. One particular algorithm (PROTEUS-2) can track all the signal subspace EVD components in only $O(LM)$ operations while achieving a performance comparable with an exact approach *and* maintaining perfect orthonormality of the eigenvectors. Furthermore, the new algorithms show no sign of numerical instability or error buildup.

The paper is organized as follows. In Section II, a mathematical formulation of subspace tracking is provided, and a series of preprocessing steps are described to put the problem in a normalized form. In Section III, the parametric perturbation approach is exposed, and the decompositions of the associated exponential matrix into products of plane rotations are derived, thus leading to the formulation of the PROTEUS schemes. The new subspace trackers based on these schemes are presented in Section IV, where convergence and numerical stability issues are discussed briefly. The comparative performance of these new algorithms is investigated via computer experiments in Section V. Some final remarks are provided in Section VI.

The following notations are used:

$\mathbb{R}$	set of real numbers;
$\mathbb{C}$	set of complex numbers;
$E[\cdot]$	mathematical expectation;
^{T}	plain conjugate transpositions;
^{H}	complex conjugate transpositions;
$\text{diag}(\cdot)$	diagonal matrix with entries given by the arguments;
I_L	$L \times L$ identity matrix;
$\mathbf{0}$	zero vectors and matrices of appropriate dimensions;
$\det(\cdot)$	determinant;
$\|\cdot\|_2$	matrix-2 norm of its argument
$\leftarrow$	overwriting.

II. Preliminary Notions

A. Problem Formulation

In a typical application of subspace-based signal processing, an L-dimensional complex data vector $\mathbf{x}(k) \in \mathbb{C}^L$ is observed at the kth sampling instant, where $k = 1, 2, \ldots$. The sequence $\mathbf{x}(k)$ is modeled as a zero-mean, random vector process whose covariance matrix at time k is shown by

$$R(k) = E[\mathbf{x}(k)\mathbf{x}(k)^H] \tag{1}$$

where it is implicitly assumed that the process $\mathbf{x}(k)$ can be nonstationary. The eigenvalues and corresponding orthonormalized eigenvectors of the matrix $R(k)$ are denoted by $\lambda_i(k)$ and $\mathbf{q}_i(k)$, $i = 1, \ldots, L$, respectively. That is, the matrices

$$\Lambda(k) = \text{diag}(\lambda_1(k), \ldots, \lambda_L(k)) \tag{2}$$

$$Q(k) = [\mathbf{q}_1(k), \ldots, \mathbf{q}_L(k)] \tag{3}$$

satisfy

$$R(k) = Q(k)\Lambda(k)Q(k)^H \tag{4}$$

$$Q(k)^H Q(k) = I_L. \tag{5}$$

Equation (4) is known as the eigenvalue decomposition (EVD) of $R(k)$. Without loss of generality, it is convenient to assume that $\lambda_1(k) \geq \lambda_2(k) \geq \cdots \geq \lambda_L(k) \geq 0$.

Within this framework, the most exhaustive form of subspace tracking consists of performing sequential estimation of all the EVD parameters of $R(k)$ as new observations $\mathbf{x}(k)$ become available. To this end, recursive algorithms are needed that can compute the EVD estimates at time k, i.e., estimates of $\Lambda(k)$ and $Q(k)$ in (4), given estimates of $\Lambda(k-1)$ and $Q(k-1)$ and the new data vector $\mathbf{x}(k)$. To be of practical value, these algorithms must have low complexity so that their real-time implementation is conceivable. In addition, they must possess the following statistical properties: In a stationary environment, the EVD estimates converge to the true EVD for k sufficiently large; whereas in a nonstationary environment, the estimates have the ability to track variations in the true EVD. Finally, numerical stability over long periods of operation is also a practical necessity.

While in certain specific applications it may be necessary to track all the EVD information, very often, only a subset thereof is required, such as, for example, the M largest or $L - M$ smallest eigenvalues and corresponding eigenvectors of $R(k)$, for some integer $M < L$, or only a basis of one of the corresponding eigensubspaces. We refer to the first class of problems as *complete* and to the second, larger class of problems as *partial* subspace tracking. In this paper, we shall be concerned at first with deriving efficient algorithms for the complete problem. Later, in Section IV, we describe some further simplifications that result for partial problems of particular interest.

Let the sequential estimates of $\Lambda(k)$ of $Q(k)$ in (4) be represented by

$$\Gamma(k) = \text{diag}(\gamma_1(k), \ldots, \gamma_L(k)) \quad (6)$$

$$U(k) = [\mathbf{u}_1(k), \ldots, \mathbf{u}_L(k)] \quad (7)$$

respectively, where $\gamma_1(k) \geq \gamma_2(k) \geq \cdots \geq \gamma_L(k) \geq 0$. In this work, we seek computationally efficient recursions for updating these estimates so that within a good degree of approximation

$$U(k)\Gamma(k)U(k)^H = (1-\epsilon)U(k-1)\Gamma(k-1)U(k-1)^H + \epsilon\mathbf{x}(k)\mathbf{x}(k)^H \quad (8)$$

where ϵ is a constant parameter with $0 < \epsilon < 1$, subject to the constraint

$$U(k)^H U(k) = I_L \quad (9)$$

which must be enforced exactly at all time. The underlying motivations are discussed below.

When (8) is satisfied exactly for all k, the factors of $U(k)\Gamma(k)U(k)^H$ indeed provide the EVD of a sample covariance matrix with exponential window, i.e., defined through the recursion

$$\hat{R}(k) = (1-\epsilon)\hat{R}(k-1) + \epsilon\mathbf{x}(k)\mathbf{x}(k)^H \quad (10)$$

which is the basis of several subspace trackers. Note that ϵ in (10) is a forgetting parameter that determines the effective length of the exponential window (namely, $1/\epsilon$). In nonstationary environments, a larger value of ϵ typically results in better tracking capabilities at the expense of increased estimator variance. In practical applications of (10), ϵ is usually much smaller than one.

In the area of matrix analysis, the computational problem specified by (8) and (9) is known as a rank-one EVD update. Several subspace trackers have been proposed that seek an *exact* solution to this problem, e.g., [12], [41], [53]). However, sequential EVD estimates obtained in this way are still subject to statistical errors and are generally not optimal in a nonstationary environment. For this reason, we do not require that (8) be satisfied exactly. Our intent is to allow for approximations that can significantly reduce the computational complexity of the resulting subspace trackers without adversely affecting their statistical performance. Clearly, such approximations will be legitimate if they are masked by the estimation errors in $\hat{R}(k)$ (10).

A fundamental limitation common to many of the existing subspace tracking algorithms is their inability to produce perfectly orthonormal eigenvector estimates. Thus, if such a basis of eigenvectors is required by the postprocessing method, further orthonormalization is necessary, adding a significant computational load to the overall process. Here, to overcome this limitation, we further require that the orthonormality constraint (9) be strictly enforced for all $k \in \{1, 2, \ldots\}$.

B. Preprocessing and Normalization

We next describe a series of preprocessing steps that are used to put the rank-one EVD update (8) into a normalized form. This will not only simplify subsequent analysis but also result in important computational savings in various situations of practical interest, e.g., complex data and/or repeated eigenvalues. For brevity, let $U \equiv U(k-1)$, $\Gamma \equiv \Gamma(k-1)$, and $\mathbf{x} \equiv \mathbf{x}(k)$ denote the information available at time k, prior to the update, and let $U' \equiv U(k)$ and $\Gamma' \equiv \Gamma(k)$ denote the updated EVD estimates. Preprocessing consists of the following four steps, in which $\mathbf{x}$, U, and Γ are transformed so that (8) is gradually brought into the desired form:

1) *Diagonalization*: Rank-one EVD update of a diagonal matrix is obtained by setting

$$\boldsymbol{\xi} = U^H\mathbf{x}. \quad (11)$$

2) *Mapping into Real Vector Space* [41]: To modify the updating problem so that only real quantities are involved, define the diagonal unitary matrix

$$D = \text{diag}(\xi_i/|\xi_i|) \quad (12)$$

where ξ_i denotes the ith entry of $\boldsymbol{\xi}$, and let

$$\boldsymbol{\xi} \leftarrow D^H\boldsymbol{\xi} \text{ (i.e., } \xi_i \leftarrow |\xi_i|), \quad U \leftarrow UD. \quad (13)$$

3) *Deflation* [5]: The dimensionality of the problem can be reduced whenever some of the diagonal elements of Γ are repeated.[1] Specifically, suppose that there are $K < L$ distinct eigenvalues among the $\gamma_i(k-1)$, $i = 1, \ldots, L$. Then, by using an appropriate block Householder matrix $H = \text{diag}(H_1, \ldots, H_K)$ (see [5] for a definition of the block matrices H_i), it is possible to zero out $L - K$ entries of the vector $\boldsymbol{\xi}$. Thus, we have

$$\boldsymbol{\xi} \leftarrow H^T\boldsymbol{\xi}, \quad U \leftarrow UH. \quad (14)$$

4) *Reordering*: Using an appropriate permutation matrix P_1 (see [22]), reorder the entries of $\boldsymbol{\xi}$, the columns of U, and the diagonal entries of Γ so that the last $L - K$ entries of $\boldsymbol{\xi}$ are zero and the first K diagonal entries of Γ are in decreasing order

$$\boldsymbol{\xi} \leftarrow P_1^T\boldsymbol{\xi}, \quad U \leftarrow UP_1, \quad \Gamma \leftarrow P_1^T\Gamma P_1. \quad (15)$$

[1] Deflation is also possible when a particular entry of $\boldsymbol{\xi}$ is zero [5]. However, due to background noise, the probability of this event is null in most applications. This type of deflation is not given further consideration.

TABLE I
SUMMARY OF PREPROCESSING STEPS

Step	Operation
1	$\boldsymbol{\xi} = U^H \mathbf{x}$
2	$D = \mathrm{diag}(\xi_i/\|\xi_i\|)$ $\boldsymbol{\xi} \leftarrow D^H \boldsymbol{\xi}$ $U \leftarrow UD$
3	$H =$ block Householder matrix $\boldsymbol{\xi} \leftarrow H^T \boldsymbol{\xi}$ $U \leftarrow UH$
4	$P_1 =$ permutation matrix $\boldsymbol{\xi} \leftarrow P_1^T \boldsymbol{\xi}$ $U \leftarrow UP_1$ $\Gamma \leftarrow P_1^T \Gamma P_1$

These steps are summarized in Table I for future reference. Their practical realization and computational complexity actually depend on the particular type of subspace tracking problem under consideration; additional explanations are provided in Section IV.

Following these steps, $\boldsymbol{\xi}$ and Γ can be partitionned as

$$\boldsymbol{\xi}^T = \left[\boldsymbol{\xi}_u^T, \mathbf{0}\right], \quad \Gamma = \mathrm{diag}(\Gamma_u, \Gamma_l) \tag{16}$$

where $\boldsymbol{\xi}_u = [\xi_1, \ldots, \xi_K]^T$ with $\xi_i > 0$, $\Gamma_u = \mathrm{diag}(\gamma_1, \ldots, \gamma_K)$ with $\gamma_1 > \gamma_2 > \cdots > \gamma_K \geq 0$, and $\Gamma_l = \mathrm{diag}(\gamma_{K+1}, \ldots, \gamma_L)$ with $\gamma_i \in \{\gamma_1, \ldots, \gamma_K\}$ for $i > K$. Furthermore, (8) can be expressed in the form

$$U'\Gamma'U'^H = U \begin{bmatrix} (1-\epsilon)\Gamma_u + \epsilon \boldsymbol{\xi}_u \boldsymbol{\xi}_u^T & \mathbf{0} \\ \mathbf{0} & (1-\epsilon)\Gamma_l \end{bmatrix} U^H. \tag{17}$$

According to (17), the original rank-one EVD update problem (8), (9) over $\mathbb{C}^L$ has been simplified to the rank-one EVD update of a diagonal matrix over $\mathbb{R}^K$, i.e.,

$$V\Gamma_u' V^T = (1-\epsilon)\Gamma_u + \epsilon \boldsymbol{\xi}_u \boldsymbol{\xi}_u^T \tag{18}$$
$$V^T V = I_K \tag{19}$$

where Γ_u' is diagonal. Once the solutions V and Γ_u' of the *normalized problem* (18), (19) have been found, Γ' and U' can be obtained from

$$U' = U \begin{bmatrix} V & \mathbf{0} \\ \mathbf{0} & I_{L-K} \end{bmatrix}, \quad \Gamma' = \begin{bmatrix} \Gamma_u' & \mathbf{0} \\ \mathbf{0} & (1-\epsilon)\Gamma_l \end{bmatrix}. \tag{20}$$

III. NEW EVD UPDATING SCHEMES BASED ON PLANE ROTATIONS

As explained in Section II-A, in subspace tracking applications, (18) needs not be satisfied exactly, but it is generally desirable to enforce the constraint (19). In this section, a new kind of improved *parametric perturbation* approach is first used to derive an approximate solution to (18), which satisfies (19) exactly. Two approximate decompositions of the resulting matrix V into a product of plane rotations are then obtained. When used in connection with (20), these decompositions lead to new algorithms for the rank-one EVD update. The algorithms are particularly attractive for subspace tracking applications: The eigenvalues are obtained in close form (no iterative search), the eigenvector update (20) is achieved efficiently via sequences of plane rotations (complexity ranging from $O(LK)$ to $O(LK^2)$), and, as a result, the eigenvectors remain orthonormal during the update.

The parametric perturbation approach is presented in Section III-A. The two decompositions of the matrix V into products of plane rotations along with the corresponding EVD updating algorithms are derived individually in Sections III-B and III-C. We refer to these algorithms as PROTEUS-1 and -2, where PROTEUS stands for "plane rotation-based EVD updating scheme."

A. Parametric Perturbation of the Rank-One EVD Update

In essence, the parametric perturbation approach consists of applying perturbation methods in a lower dimensional parameter space associated with the constraint of interest, i.e., (19). Here, only first-order perturbation series are used so that it can also be viewed as a form of constrained linearization [7]. Its application to subspace tracking is based on the assumption that the memory parameter ϵ in (18) is small compared with 1, which is verified in most situations of interest. For example, to obtain an effective exponential window length greater than 20 samples, we need $\epsilon < 0.05$.

We begin by observing that for ϵ in the neighborhood of 0, the modified EVD components Γ_u' and V in (18) and (19) can be analytically connected to Γ_u and I_K, respectively, so that $\Gamma_u' \to \Gamma_u$ and $V \to I_K$ in the limit $\epsilon \to 0$ [38]. Thus, for ϵ sufficiently small, the EVD modifications resulting from the update (18) are small, that is, $\|\Gamma_u' - \Gamma_u\|_2 \ll \gamma_1$ and $\|V - I_K\|_2 \ll 1$. To emphasize this point, let us first write Γ_u' in the form

$$\Gamma_u' = \Gamma_u + \Delta\Gamma_u \tag{21}$$
$$\Delta\Gamma_u = \mathrm{diag}(\delta_1, \ldots, \delta_K) \tag{22}$$

where the unknown parameter δ_i, $i \in \{1, \ldots, K\}$, represents the modification in the ith eigenvalue. According to the above discussion, $|\delta_i| \ll \gamma_1$, provided ϵ is sufficiently small.

The introduction of a similar representation for V in terms of small parameters requires additional care because of the orthogonality constraint (19). To derive such a representation, we first note that $\det(V) = \pm 1$ as a consequence of (19). Without loss of generality, we shall assume that $\det(V) = +1$; this amounts to multiplying one of the modified eigenvectors (i.e., any column of V) by -1. With this additional restriction, V now belongs to the special orthogonal group $SO(K)$ [42], i.e., the group of all $K \times K$ unimodular orthogonal matrices, which is also known as proper rotations. As a member of $SO(K)$, V automatically admits an exponential representation of the form

$$V = \exp(\Theta) \tag{23}$$

where $\Theta = [\theta_{ij}]$ is a skew-symmetric matrix in $\mathbb{R}^{K \times K}$ (i.e., $\Theta^T = -\Theta$, or equivalently, $\theta_{ji} = -\theta_{ij}$), and $\exp(\cdot)$ is the matrix exponential function, which is defined as

$$\exp(\Theta) = \sum_{k=0}^{\infty} \frac{\Theta^k}{k!}. \tag{24}$$

The above representation of V in terms of the $K(K-1)/2$ real parameters θ_{ij} for $1 \leq i < j \leq K$ is of particular interest to us. Indeed, we have seen that for ϵ sufficiently small, $\|V - I_K\|_2 \ll 1$; in light of (23), this in turns implies $|\theta_{ij}| \ll 1$.

The remaining steps of the proposed parametric perturbation approach can be summarized as follows:

i) Substitute (21) and (23) into (18), and use (24) to expand the exponential function (i.e., $V = I_K + \Theta + \cdots$).
ii) Assuming ϵ small, retain only linear terms (i.e., of degree zero or one) in $\Delta\Gamma_u$ and Θ.
iii) Solve the resulting equation for $\Delta\Gamma_u$ and Θ.
iv) Substitute the solutions back into (21) and (23).

The matrices Γ'_u and V so obtained are the desired solutions to the rank-one EVD update problem (18). Since the approximation in step ii) is made at the Θ level, it has no effect on the orthogonality of V, which is kept in $SO(K)$ via (23); the orthonormality constraint (19) is thus automatically satisfied. These steps are carried out below.

Steps i) and ii) lead to

$$\Delta\Gamma_u + \Theta\Gamma_u + \Gamma_u\Theta^T = \epsilon\big(\boldsymbol{\xi}_u\boldsymbol{\xi}_u^T - \Gamma_u\big) \tag{25}$$

which is linear in $\Delta\Gamma_u$ and Θ. To implement step iii), we must consider independently the diagonal and off-diagonal entries in (25). Doing so, we arrive at[2]

$$\delta_i = \epsilon\big(\xi_i^2 - \gamma_i\big), \quad i = 1, \ldots, K \tag{26}$$

$$\theta_{ij} = \epsilon\xi_i\xi_j/(\gamma_j - \gamma_i), \quad 1 \le i < j \le K. \tag{27}$$

The remaining entries of Θ are obtained from its skew-symmetry property (i.e., $\theta_{ii} = 0$ and $\theta_{ij} = -\theta_{ji}$ for $i > j$). Note that in the normalized EVD update problem (18), the eigenvalues $\gamma_i,\ i = 1, \ldots, K$ are distinct so that division by $\gamma_j - \gamma_i$ in (27) is legitimate. For the final step iv), we first substitute (26) into (21) and (22), yielding the updated eigenvalue matrix

$$\Gamma'_u = (1-\epsilon)\Gamma_u + \epsilon\,\mathrm{diag}\big(\xi_1^2, \ldots, \xi_K^2\big). \tag{28}$$

The updated eigenvector matrix V is obtained in the same way upon substitution of (27) in (23).

It turns out that the exponential function in (23) needs not be evaluated explicitly. Indeed, two different approximate decompositions of (23) as a product of plane rotations are derived in the following subsections. When combined with (20), each decomposition leads to a new, computationally efficient algorithm for EVD updating (namely, PROTEUS 1 or 2) in which a specific sequence of plane rotations is used to update the eigenvectors, thereby preserving orthonormality.

B. PROTEUS-1

The only assumption involved in the following derivation of PROTEUS-1 is that $\epsilon \ll 1$. As pointed out in Section III-A, this implies that $\|\Theta\|_2 \ll 1$, which in turn is exploited to obtain the desired decomposition of V (23) into a product of plane rotations.

Consider the $K \times K$ matrix Θ with entries θ_{ij} (27). Let Ψ_{ij} denote the matrix obtained from Θ by setting all its entries to zero, except for the ij and ji entries (i.e., θ_{ij} and $\theta_{ji} = -\theta_{ij}$), which are left unchanged. It is then possible to express Θ in terms of the matrices Ψ_{ij} as

$$\Theta = \sum_{j>i} \Psi_{ij}. \tag{29}$$

Substituting (29) in (23) and using the definition (24), it can be verified that

$$V = \prod_{j>i} \exp(\Psi_{ij}) + O\big(\|\Theta\|_2^2\big). \tag{30}$$

Thus, V can be expressed as a finite product of simpler orthogonal matrices, namely, $\exp(\Psi_{ij})$, plus an error term of the order of $\|\Theta\|_2^2$. Here, the entries of Θ are given by (27) so that $\|\Theta\|_2 = O(\epsilon)$, and the error term in (30) goes to zero as ϵ^2 in the limit $\epsilon \to 0$. Thus, for small values of ϵ, which is the situation of interest in this work, it is reasonable to neglect this term. Note that this has no effect on the orthogonality of the matrix V. Furthermore, in light of the linearization approach used in the previous section, this does not represent an additional approximation.[3]

Now, consider the matrix $\exp(\Psi_{ij})$, which is the basic building block in (30). Using the definition (24) once more, we can verify that (see Appendix A for details)

$$\exp(\Psi_{ij}) = G_{ij}(\theta_{ij}) \tag{31}$$

where $G_{ij}(\theta) \in \mathbb{R}^{K\times K}$ is the well-known plane (or Givens) rotation matrix defined as [22]

$$G_{ij}(\theta) = \begin{bmatrix} I_{i-1} & & & & \\ & \cos(\theta) & \cdots & \sin(\theta) & \\ & \vdots & I_{j-i-1} & \vdots & \\ & -\sin(\theta) & \cdots & \cos(\theta) & \\ & & & & I_{K-j} \end{bmatrix} \tag{32}$$

with all the unspecified entries equal to zero. Insight into the signification of the parameters θ_{ij} in (27) can be gained from this result. Indeed, when considered individually, each θ_{ij} can be interpreted as a small rotation angle (in radians) in the ij-coordinate plane.

Substituting (31) in (30) and neglecting the second-degree error term, a first decomposition of V is obtained, namely

$$V_1 = \prod_{j>i} G_{ij}(\theta_{ij}) \tag{33}$$

with θ_{ij} given by (27). This result simply states that for ϵ small, the matrix V (23) can be expressed as the product of $K(K-1)/2$ plane rotation matrices with small rotation angles θ_{ij}. Furthermore, to the first degree of approximation in Θ, the order in which these rotations appear is immaterial. This is analogous to the well-known fact that small rotations in three-dimensional (3-D) Euclidean space almost commute.

[2] As pointed out by a reviewer, similar equations also occur in the area of continuous-time matrix differential equations. See [15] for additional detail and references.

[3] Note that in Section III-A, the parametrization $V = \prod_{j>i} \exp(\Psi_{ij})$ might have been used directly instead of (23). However, in light of the derivation of PROTEUS-2 in Section III-C, we have found it preferable to use (23).

TABLE II
PROTEUS-1 ALGORITHM (TOTAL COMPLEXITY: $6LK^2 + 1.5K^2 + 2.5K$ RFLOP's)

Step	Operation	Flop count (rflop)
1	$\boldsymbol{\xi} \leftarrow \sqrt{\epsilon}\,\boldsymbol{\xi}$	K
2	for $i = 1 : K-1$	$K(K-1)/2\ \times$
	for $j = i+1 : K$	
	$\theta \leftarrow \xi_i\xi_j/(\gamma_j - \gamma_i)$	3
	$U \leftarrow UG_{ij}(\theta)$	$12L$
	end	
	end	
3	$\Gamma \leftarrow (1-\epsilon)\Gamma + \mathrm{diag}(\xi_i^2)$	$3K$

Based on the decomposition (33), a complete EVD updating algorithm is now obtained. For the eigenvector update, substitute V_1 (33) in the place of V in (20), which can then be expressed as

$$U' = U \prod_{j>i} G_{ij}(\theta_{ij}) \tag{34}$$

where $G_{ij}(\theta_{ij})$ now represents a plane rotation matrix in $\mathbb{R}^{L\times L}$ obtained from (32) by using I_{L-j} instead of I_{K-j} as the lower right block diagonal element. According to (34), the updated eigenvector matrix U' is obtained from postmultiplication of U by a sequence of plane rotations with angles θ_{ij} given by (27). For the eigenvalue update, simply substitute (28) into (20). Observing that the last $L-K$ entries of $\boldsymbol{\xi}$ (16) are zero, it follows that

$$\Gamma' = (1-\epsilon)\Gamma + \epsilon\,\mathrm{diag}(\xi_1^2, \ldots, \xi_L^2). \tag{35}$$

The complete algorithm is presented in Table II under the name PROTEUS-1.

The initial values of $\boldsymbol{\xi}$, Γ, and U used in this algorithm are those obtained after preprocessing (Section II-B). In Step 1 (Table II), the parameter ϵ is incorporated in the data vector in order to reduce the overall number of computations. Step 2 is a double loop over the rows $(i = 1, \ldots, K)$ and the columns $(j > i)$ of the matrix Θ that implements the sequence of $K(K-1)/2$ plane rotations composing (34). As indicated earlier, the order in which the rotations are performed might be modified to better suit a specific processing architecture. Step 3 performs the eigenvalue update via a finite computation, that is, no iterative search is involved.

In the case of complex data, each plane rotation requires $12L$ real floating-point operations (rflops), plus some overhead to compute the trigonometric functions. The total complexity of the algorithm in Table II is thus $6LK^2 + 1.5K^2 + 2.5K$ rflops.[4]

C. PROTEUS-2

The $O(LK^2)$ operation count of the PROTEUS-1 algorithm may be prohibitive for certain real-time subspace tracking applications in which large values of L and/or K are used. In this section, we present an alternative algorithm PROTEUS-2 with an $O(LK)$ operation count. Its derivation is based on the following assumptions: $\epsilon \ll 1$, as in the case of PROTEUS-1, and $\gamma_1 \gg \gamma_2 \gg \cdots \gg \gamma_K$. The latter is equivalent to requiring that the eigenvalues are well separated. However, we point out immediately that the use of PROTEUS-2 is not restricted to this situation. Further related observations are provided at the end of the section.

Recall that after preprocessing (Table I), $\gamma_1 > \gamma_2 > \cdots > \gamma_K$. If we further assume that the eigenvalues are well separated, i.e., $\gamma_k \gg \gamma_l$ for $k < l$, then the rotation angles θ_{ij} (27) can be approximated as

$$\theta_{ij} = -\epsilon\xi_i\xi_j/\gamma_i, \quad 1 \le i < j \le K. \tag{36}$$

In order to preserve the skew symmetry of Θ, we still let $\theta_{ij} = -\theta_{ji}$ for $j \ge i$.

Next, we introduce a particular block representation for the matrix Θ, with entries θ_{ij} now given by (36). Let $\Theta_k(\xi_1, \ldots, \xi_k)$, where the dependence on the ξ_i's is indicated explicitly, denote the principal submatrix of Θ corresponding to its first k rows and first k columns. Then, we have

$$\begin{aligned}\Theta &\equiv \Theta_K(\xi_1, \ldots, \xi_K) \\ &= \begin{bmatrix} \Theta_{K-2}(\xi_1, \ldots, \xi_{K-2}) & -\epsilon\xi_{K-1}\boldsymbol{\phi} & -\epsilon\xi_K\boldsymbol{\phi} \\ \epsilon\xi_{K-1}\boldsymbol{\phi}^T & 0 & \theta_{K-1,K} \\ \epsilon\xi_K\boldsymbol{\phi}^T & -\theta_{K-1,K} & 0 \end{bmatrix}\end{aligned} \tag{37}$$

where

$$\boldsymbol{\phi} = (\phi_1, \ldots, \phi_{K-2})^T, \quad \phi_i = \xi_i/\gamma_i. \tag{38}$$

We can see from (37) that as a direct consequence of our assumption of well-separated eigenvalues, the two $K-2$ dimensional column vectors occupying the upper-right corner of Θ are now linearly related. Below, this additional structural property is exploited to derive a decomposition of $\exp(\Theta)$ as a product of only $2K-3$ plane rotations.

Let

$$\alpha_{K-1} = -\arctan(\xi_K/\xi_{K-1}) \tag{39}$$

and

$$A_{K-1} = G_{K-1,K}(\alpha_{K-1}). \tag{40}$$

The plane rotation matrix A_{K-1} so defined can be used to zero the last entry of the data vector $\boldsymbol{\xi}_u = [\xi_1, \ldots, \xi_K]^T$. More specifically, premultiplication of $\boldsymbol{\xi}_u$ by A_{K-1}^T performs a clockwise rotation by $-\alpha_{K-1}$ rad in the $(K-1, K)$ coordinate plane so that

$$A_{K-1}^T\boldsymbol{\xi}_u = [\xi_1, \ldots, \xi_{K-2}, \xi'_{K-1}, 0]^T \tag{41}$$

where

$$\xi'_{K-1} = \sqrt{\xi_{K-1}^2 + \xi_K^2}. \tag{42}$$

Thus, premultiplication of Θ (37) by A_{K-1}^T followed by postmultiplication by A_{K-1} will zero the first $K-2$ entries in the last row and last column of Θ without affecting the other

[4] In the literature, complexity is sometimes measured in terms of complex multiply–add operations (cops). In this paper, we prefer to use the rflops; to obtain the approximate complexity in cops, simply divide the rflops count by 8.

entries of the matrix. Equivalently, since A_{K-1} is orthogonal, i.e., $A_{K-1}^T = A_{K-1}^{-1}$, we have

$$\Theta = A_{K-1}\begin{bmatrix}\Theta_{K-2}(\xi_1,\ldots,\xi_{K-2}) & -\epsilon\xi'_{K-1}\phi & \mathbf{0}\\ \epsilon\xi'_{K-1}\phi^T & 0 & \theta_{K-1,K}\\ \mathbf{0} & -\theta_{K-1,K} & 0\end{bmatrix} \times A_{K-1}^T. \tag{43}$$

Using a well-known property of the matrix exponential function, namely, $\exp(ABA^{-1}) = A\exp(B)A^{-1}$, it follows that

$$\begin{aligned}\exp(\Theta) &= A_{K-1}\exp\\ &\quad\times\left(\begin{bmatrix}\Theta_{K-2}(\xi_1,\ldots,\xi_{K-2}) & -\epsilon\xi'_{K-1}\phi & \mathbf{0}\\ \epsilon\xi'_{K-1}\phi^T & 0 & \theta_{K-1,K}\\ \mathbf{0} & -\theta_{K-1,K} & 0\end{bmatrix}\right)\\ &\quad\times A_{K-1}^T\\ &= A_{K-1}\exp\\ &\quad\times\left(\Psi_{K-1,K} + \begin{bmatrix}\Theta_{K-1}(\xi_1,\ldots,\xi_{K-2},\xi'_{K-1}) & \mathbf{0}\\ \mathbf{0} & 0\end{bmatrix}\right)A_{K-1}^T\end{aligned} \tag{44}$$

with $\Psi_{K-1,K}$ as defined at the beginning of Section III-B.

Now, observing that the two matrices in the argument of the exponential function in the last line of (44) are $O(\epsilon)$, we have

$$\begin{aligned}\exp(\Theta) &= A_{K-1}\exp(\Psi_{K-1,K})\\ &\quad\times\exp\left(\begin{bmatrix}\Theta_{K-1}(\xi_1,\ldots,\xi_{K-2},\xi'_{K-1}) & \mathbf{0}\\ \mathbf{0} & 0\end{bmatrix}\right)A_{K-1}^T\\ &\quad+O(\epsilon^2).\end{aligned} \tag{45}$$

From our development in Section III-B, we recall that $\exp(\Psi_{K-1,K}) = G_{K-1,K}(\theta_{K-1,K})$. Furthermore, we note that for any coordinate pair (i,j) and for any rotation angle α and β, $G_{ij}(\alpha)G_{ij}(\beta) = G_{ij}(\alpha+\beta)$. Thus, (45) can be expressed in the form

$$\begin{aligned}\exp(\Theta) &= B_{K-1}\\ &\quad\times\exp\left(\begin{bmatrix}\Theta_{K-1}(\xi_1,\ldots,\xi_{K-2},\xi'_{K-1}) & \mathbf{0}\\ \mathbf{0} & 0\end{bmatrix}\right)\\ &\quad\times A_{K-1}^T + O(\epsilon^2)\end{aligned} \tag{46}$$

where

$$\begin{aligned}B_{K-1} &= A_{K-1}\exp(\Psi_{K-1,K})\\ &= G_{K-1,K}(\alpha_{K-1}+\theta_{K-1,K}).\end{aligned} \tag{47}$$

Equation (46) indicates that up to an error term of order $O(\epsilon^2)$, the matrix $\exp(\Theta)$ can be factorized as a product of two plane rotations and an orthogonal matrix with only $(K-1)(K-2)/2$ nonzero rotation parameters.

Clearly (i.e., invoking mathematical induction), this procedure can be repeated until $V = \exp(\Theta)$ has been entirely factorized as a product of plane rotations, plus an error term in $O(\epsilon^2)$. Dropping this term, we finally obtain our second approximate decomposition of V (23), namely

$$V_2 = B_{K-1}\cdots B_2 C A_2^T\cdots A_{K-1}^T \tag{48}$$

TABLE III
PROTEUS-2 ALGORITHM (TOTAL COMPLEXITY: $24LK + (2\nu - 14)K - 2(2\nu+7)$ RFLOP's)

Step	Operation	Flop count (rflop)
1	$\boldsymbol{\xi} \leftarrow \sqrt{\epsilon}\,\boldsymbol{\xi}$	K
2	$\xi'_K \leftarrow \xi_K$	
	for $i = K-1:-1:2$	$(K-2)\times$
	$\alpha_i \leftarrow -\arctan(\xi'_{i+1}/\xi_i)$	$\nu+1$
	$\theta \leftarrow -\xi_i\xi'_{i+1}/\gamma_i$	2
	$\xi'_i \leftarrow \sqrt{\xi_i^2 + \xi'^2_{i+1}}$	$\nu+3$
	$U \leftarrow UG_{i,i+1}(\alpha_i+\theta)$	$12L+1$
	end	
	$\theta \leftarrow -\xi_1\xi'_2/\gamma_1$	2
	$U \leftarrow UG_{1,2}(\theta)$	$12L$
	for $i = 2:K-1$	$(K-2)\times$
	$U \leftarrow UG_{i,i+1}^T(\alpha_i)$	$12L$
	end	
3	$\Gamma \leftarrow (1-\epsilon)\Gamma + \mathrm{diag}(\xi_i^2)$	$3K$

where

$$A_i = G_{i,i+1}(\alpha_i),\quad B_i = G_{i,i+1}(\beta_i),\quad C = G_{1,2}(\theta_1). \tag{49}$$

The rotation angles α_i, β_i, and θ_1 in (49) are given by

$$\begin{aligned}\alpha_i &= -\arctan(\xi'_{i+1}/\xi_i),\\ \theta_i &= -\epsilon\xi_i\xi'_{i+1}/\gamma_i,\\ \beta_i &= \alpha_i + \theta_i\end{aligned} \tag{50}$$

where ξ'_i is defined recursively as

$$\xi'_i = \begin{cases}\xi_i, & i = K\\ \sqrt{\xi_i^2 + \xi'^2_{i+1}}, & i = K-1,\ldots,1.\end{cases} \tag{51}$$

According to (48), the matrix V (23) can be approximated by a product of $2K-3$ plane rotation matrices. In contrast to the decomposition V_1 in (33), the order in which the plane rotations appear in (48) is now important because the rotation angles α_i and β_i are not small in general. The above procedure is summarized in the form of the PROTEUS-2 algorithm in Table III.[5] In the complex data case, the total operation count of this algorithm is $24LK + (2\nu-14)K - (4\nu+14)$ rflops, where ν is used here as a common flop count for the operations $\sqrt{\ }$ and $\arctan(\cdot)$.

Some comments are in order about the practical validity of PROTEUS-2. The latter is based on a sequence of approximations between the last two rows and columns of the matrices $\Theta_k(\xi_1,\ldots,\xi_{k-1},\xi'_k)$, starting with $k = K$ down to $k = 3$. As seen, a particular situation where this can be justified theoretically is when $\gamma_1 \gg \gamma_2 \gg \cdots \gg \gamma_K$. Such a spacing of eigenvalues is sometimes encountered in frequency and DOA estimation when the dimension of the signal subspace is small and the SNR is high. Other eigenvalue configurations may also exist for which such a theoretical justification of PROTEUS-2 is possible. More generally, we have found through various experiments that the latter is robust and can be used essentially without regard for the eigenvalue

[5] This version of PROTEUS-2 differs slightly from the one originally presented in [8], where the variable θ was computed as $\xi_i\xi'_{i+1}/(\gamma_{i+1}-\gamma_i)$. We have found that the new way of computing θ in Table III is more robust to rapid changes in the statistics of the observed data.

TABLE IV
PROTEUS-BASED ALGORITHM FOR TRACKING COMPLETE EVD

Step	Operation	Flop count (rflop)
1	$U \leftarrow U(0)$ $\Gamma \leftarrow \Gamma(0)$	
2	for $k = 1, 2, ...$	
(a)	$\mathbf{x} \leftarrow \mathbf{x}(k)$	
(b)	Preprocessing (Table 1)	$14L^2 + (\nu + 5)L$
(c)	Normalized EVD update with any one of PROTEUS algorithms from Table 2 or 3	$6L^3 + 1.5L^2 + 2.5L$ **or** $24L^2 + (2\nu - 14)L - (4\nu + 14)$
(d)	Postprocessing (optional, see text)	
	end	

configuration at the expense of possibly minor deteriorations in tracking performance. Additional theoretical and experimental support for this claim are provided in Sections IV and V.

IV. Applications to Subspace Tracking

In this section, we discuss the application of the PROTEUS algorithms to subspace tracking. Related issues of convergence and numerical stability are also addressed briefly.

A. Complete Subspace Tracking

In this particular form of the subspace tracking problem, we are interested in tracking the complete EVD of the data covariance matrix $R(k)$ (4), i.e., updating the full matrix estimates $\Gamma(k)$ and $U(k)$ in (6) and (7). This can be achieved easily by repeated application of PROTEUS 1 or 2 at each time iteration. The procedure, which must also take into account preprocessing as discussed in Section II-B, is detailed in Table IV, whose description follows.

In the absence of *a priori* knowledge, the algorithm can be initialized by using $U(0) = I_L$ and $\Gamma(0) = \text{diag}(\lambda_1, \dots, \lambda_L)$ in Step 1, with meaningful choices of eigenvalues λ_i (e.g., distinct, positive eigenvalues covering the power range of interest). Other initialization procedures are possible, such as using the EVD of an initial low-rank estimate of the covariance matrix $R(k)$. Step 2, i.e., the loop over the time index k, defines the sequential portion of the algorithm. It consists of four groups of operations:

a) data acquisition;
b) preprocessing as defined in Table I;
c) normalized EVD update with PROTEUS 1 or 2, as described in Tables II and III, respectively;
d) postprocessing.

Postprocessing is optional and contains additional operations that may be needed in specific applications. For example, it may be necessary to permute the EVD components so that the eigenvalues appear in nonincreasing order after the update. In other situations involving closely spaced eigenvalues, it may be advantageous to force these eigenvalues to be equal by replacing them with their average. This will result in computational savings and may improve the accuracy of the estimated EVD. The decision as to whether certain eigenvalues should be considered identical can either be based on *a priori* knowledge, numerical considerations, or on statistical hypothesis testing.

In the case of distinct eigenvalues, the operation count of the preprocessing step (Table I) is $14L^2 + (\nu+5)L$ flops. The total operation count of the subspace tracker in Table IV is obtained by adding this figure to the operation count of the selected PROTEUS algorithm.

B. Partial Subspace Tracking

The eigenvalues of the data covariance matrix $R(k)$ in (1) are often known to satisfy

$$\lambda_1(k) > \cdots > \lambda_M(k) > \lambda_{M+1}(k) = \cdots = \lambda_L(k) \tag{52}$$

with $L - M > 1$. This occurs, for instance, when $\mathbf{x}(k) = \mathbf{s}(k) + \mathbf{n}(k)$, where $\mathbf{s}(k)$ is a signal component whose covariance matrix has rank $M < L - 1$, and $\mathbf{n}(k)$ is an uncorrelated background noise component with covariance matrix $\lambda_{M+1}(k)I_L$. In this case, the column span of the matrix

$$Q_s(k) = [\mathbf{q}_1(k), \dots, \mathbf{q}_M(k)] \tag{53}$$

and its orthogonal complement are, respectively, called the signal and the noise subspace. In these types of applications, we are often interested in tracking only the signal-subspace eigenvectors $\mathbf{q}_i(k)$, $i = 1, \dots, M$ and the eigenvalues $\lambda_i(k)$, $i = 1, \dots, M+1$. Clearly, this information can be obtained as a by product of the subspace tracking algorithm presented in Table V. However, it is possible to develop a more efficient version of this algorithm that does not maintain and update the noise subspace eigenvectors, thus requiring less memory and computations.

To this end, suppose that the following information is available at time $k-1$: $\gamma_1 > \cdots > \gamma_M > \gamma_M + 1 = \cdots = \gamma_L$,[6] and $U_s = [\mathbf{u}_1, \dots, \mathbf{u}_M]$. Accordingly, $U = [U_s, U_n]$, where U_n contains the unknown noise-subspace eigenvectors. Beginning with preprocessing and referring to Table I, we note that $\boldsymbol{\xi}$ in Step 1 can be partitionned as $\boldsymbol{\xi}^T = [\boldsymbol{\xi}_s^T, \boldsymbol{\xi}_n^T]$, with $\boldsymbol{\xi}_s = U_s^H \mathbf{x}$ and $\boldsymbol{\xi}_n = U_n^H \mathbf{x}$, where $\boldsymbol{\xi}_s$ can be computed explicitly. The transformations in Step 2 can then be applied to $\boldsymbol{\xi}_s$ and U_s. Important specializations occur in Step 3. Indeed, we note that there are only $K = M + 1$ distinct eigenvalues. The problem can thus be deflated by using the

[6] To simplify the discussion, we assume that the signal subspace eigenvalues are distinct, but this is not necessary.

TABLE V
PROTEUS-BASED ALGORITHM FOR TRACKING RANK-M SIGNAL SUBSPACE AND NOISE EIGENVALUE

Step	Operation	Flop count (rflop)
1	$K \leftarrow M+1$	
	$U_s \leftarrow U_s(0), \quad L \times M$	
	$\Gamma \leftarrow \Gamma(0), \quad K \times K$	
2	for $k = 1, 2, \ldots$	
(a)	$\mathbf{x} \leftarrow \mathbf{x}(k)$	
(b1)	$\boldsymbol{\xi}_s \leftarrow U_s^H \mathbf{x}$	$8LM$
(b2)	$D \leftarrow \mathrm{diag}(\xi_i/\lvert\xi_i\rvert; i = 1, \ldots, M)$	$(\nu+5)M$
	$\boldsymbol{\xi}_s \leftarrow D^H \boldsymbol{\xi}_s$	
	$U_s \leftarrow U_s D$	$6LM$
(b3)	$\mathbf{x}_n \leftarrow \mathbf{x} - U_s \boldsymbol{\xi}_s$	$4LM$
	$\xi_K \leftarrow \lVert\mathbf{x}_n\rVert_2$	$4L + (\nu - 1)$
	$\mathbf{u}_K \leftarrow \mathbf{x}_n / \xi_K$	$2L$
	$U \leftarrow [U_s, \mathbf{u}_K]$	
	$\boldsymbol{\xi}^T \leftarrow [\boldsymbol{\xi}_s^T, \xi_K]$	
(c)	Normalized EVD update with PROTEUS-1 or 2	$6LK^2 + 1.5K^2 + 2.5K$ **or** $24LK + (2\nu - 14)K - (4\nu + 14)$
(d)	$\gamma_K \leftarrow \gamma_K - \frac{L-K}{L-M}\xi_K^2$	
(e)	Reordering	
	end	

block Householder matrix

$$H = \mathrm{diag}(I_M, \tilde{H}) \tag{54}$$

where $\tilde{H} = \tilde{H}^T = \tilde{H}^{-1}$ (size $L - M$) is such that

$$\tilde{H}^T \boldsymbol{\xi}_n = \lVert\boldsymbol{\xi}_n\rVert_2 \mathbf{e}_1 \tag{55}$$

and $\mathbf{e}_1$ denotes the left-most column of I_{L-M}. From (54), it follows that

$$UH = [U_s, U_n\tilde{H}]. \tag{56}$$

Finally, Step 4 is not necessary here since the eigenvalues are already in the desired order.

Next, we note that to update the signal subspace eigenvectors with PROTEUS-1 or 2, only the first $K = M + 1$ columns of UH (56) are needed (see Tables II or III). Since U_s is already known, it is thus only necessary to compute the Kth column, which is given by $\mathbf{u}_K = U_n\tilde{H}\mathbf{e}_1$. From (55), it follows immediately that

$$\mathbf{u}_K = U_n\tilde{H}\mathbf{e}_1 = U_n\boldsymbol{\xi}_n / \lVert\boldsymbol{\xi}_n\rVert_2. \tag{57}$$

To compute $\mathbf{u}_K$ (57), let $\mathbf{x}_n = \mathbf{x} - U_s\boldsymbol{\xi}_s$ be the orthogonal projection of the data vector $\mathbf{x}$ on the noise subspace, and note that $U_n\boldsymbol{\xi}_n = \mathbf{x}_n$ and $\lVert\boldsymbol{\xi}_n\rVert_2 = \lVert\mathbf{x}_n\rVert_2$.

Following an EVD update with PROTEUS-1 or 2, we generally find that $\gamma_{M+1} > \gamma_{M+2} = \cdots = \gamma_L$. Some postprocessing is therefore needed to maintain a strict equality constraint on these eigenvalue estimates, in agreement with (52). To this end, we use the noise eigenvalue smoothing technique of [23], which amounts to

$$\gamma_{M+1} \leftarrow \gamma_{M+1} - \frac{L-K}{L-M}\xi_{M+1}^2 \tag{58}$$

where here, $\xi_{M+1} = \lVert\boldsymbol{\xi}_n\rVert_2$ [see (55)]. This technique, which admits a least-square interpretation, often results in slightly improved robustness to background noise.

The complete algorithm is presented in Table V. We note that the size of the matrix U in this table is now $L \times K$, instead of $L \times L$, as was assumed previously. Thus, when using PROTEUS-1 or 2 to implement Step 2(c), the size of the plane rotations matrices $G_{ij}(\cdot)$ in Tables II or III, respectively, must be adjusted accordingly, i.e., $K \times K$. The total operation count of the algorithm in Table V is $18LM + 6L + (\nu+5)M + (\nu+2)$ rflops plus that of the selected PROTEUS algorithm. In particular, when PROTEUS-2 is used, a tracking algorithm with total complexity $O(LM)$ is obtained.

In certain applications, M is unknown or may change over time. In this case, the algorithm can easily be complemented with a postprocessing function that performs incremental rank estimation based on the available EVD information and then makes the necessary adjustments to the algorithm parameters. One such technique presented in [50] can be applied without modification to the algorithm of Table V (simply set $r(t) \equiv M$), but alternative solutions do exist. Finally, note that other partial problems of interest have been considered in the literature, such as spherical [13] and four-level [14] subspace tracking. Clearly, efficient subspace tracking algorithms based on PROTEUS-1 and 2 could be derived as well for these specific situations.

C. Convergence

Computer simulations have shown that the new subspace trackers in Tables IV and V converge to and track the desired EVD components under a wide range of operating conditions (see Section V). Some theoretical justifications for this behavior are briefly reported below.

To begin, consider the complete trackers of Table IV, with distinct eigenvalues. Coarse convergence to the vicinity of the true EVD can be linked to the ability of the recursive estimate $\hat{R}(k)$ (10) to converge to the true covariance matrix $R(k)$ (1). Indeed, let $E(k)$ denote the approximation error that results in (8) at time k from dropping second-order terms in the PROTEUS algorithms, and assume that $\|E(k)\|_2 \leq \epsilon^2\delta$ for some upper bound $\delta > 0$ independent of ϵ. It can be verified easily that the difference $\Delta_1(k) = U(k)\Gamma(k)U(k)^H - \hat{R}(k)$ is bounded by

$$\|\Delta_1(k)\|_2 \leq (1-\epsilon)^k \|\Delta_1(0)\|_2 + \epsilon\delta. \tag{59}$$

This shows that $\Delta_1(k)$ can be made arbitrarily small for k large by choosing $\epsilon > 0$ accordingly. Thus, if $\hat{R}(k)$ is a good tracker of $R(k)$ for ϵ small, so must $U(k)\Gamma(k)U(k)^H$ be after some time.

Under the assumption of stationarity, i.e., $R(k) = R = Q\Lambda Q^H$ in (4), local convergence to the true EVD can be investigated via a mean-value analysis of the error matrices $\Delta_2(k)$ and $\Delta_3(k)$, which is defined as[7]

$$\Delta_2(k) = \Gamma(k) - \Lambda \tag{60}$$

$$\exp(\Delta_3(k)) = Q^H U(k). \tag{61}$$

Assuming small errors, i.e., $\|\Delta_2(k)\|_2 \ll \gamma_1$ and $\|\Delta_3(k)\|_2 \ll 1$, and proceeding as in the analysis of stochastic gradient algorithms, it can be shown that when PROTEUS-1 is used in Table IV

$$E[\Delta_i(k)] \approx (1-\epsilon)E[\Delta_i(k-1)], \quad i = 2, 3. \tag{62}$$

Hence, for $0 < \epsilon < 1$, convergence in the mean is geometrical. When Proteus-2 is used, (62) still holds for $i = 2$ but must be modified as

$$E[\delta_{3,ij}(k)] \approx (1-\epsilon_{ij})E[\delta_{3,ij}(k-1)] \tag{63}$$

$$\epsilon_{ij} = \left(1 - \frac{\lambda_j}{\lambda_i}\right)\epsilon \leq \epsilon, \quad i < j \tag{64}$$

for $i = 3$. Note that although specific assumptions on the eigenvalues were made in the derivation of PROTEUS-2, the resulting subspace tracker converges geometrically, regardless of the particular eigenvalue configuration; only the individual rates of convergence of the matrix entries $\delta_{3,ij}$ are affected.

A more general convergence analysis of the PROTEUS-based signal subspace trackers in Table V using the ODE method [2] shows that these algorithms are locally asymptotically stable [9] and, under weak additional assumptions, globally stable.

D. Numerical Stability

In a finite-precision implementation of any recursive subspace tracker, buildups of numerical errors may occur if the algorithm is operated over long periods of time without reinitialization. Such buildups always represent a potential treat since they may lead to numerical instability, which in effects renders the algorithm impractical. In the new subspace trackers proposed here, the eigenvalue update is numerically stable due to the presence of the factor $1 - \epsilon$ in Step 3 of the PROTEUS algorithms (see Tables II and III). For $0 < \epsilon < 1$, this factor limits the buildup of numerical errors in the eigenvalue computation.

The only other possible source of error buildup is the eigenvector update via sequences of plane rotations in Step 2 of Tables II and III. It is well known that numerical errors generated through repeated application of orthogonal matrices may keep on accumulating beyond limit. In certain subspace trackers that have been proposed recently, partial orthogonalization mechanisms [12], [30], [31] are indeed necessary to avoid such an error buildup. However, we have found experimentally that the new subspace trackers based on PROTEUS-1 and 2 are not sensitive to this problem. In all the simulations that we have done, some involving as many as 10^6 iterations, error buildup never occurred (see Section VII). Thus, partial orthogonalization appears to be unnecessary with these new algorithms.[8] In effect, they have the capability of correcting small deviations from orthonormality in the eigenvector estimates, as we have been able to verify theoretically by studying their internal dynamics with the ODE method.

V. Computer Experiments

In this section, the performance of the PROTEUS-based subspace trackers is investigated via computer experiments. The data vector $\mathbf{x}(k)$ is modeled as

$$x_\mu(k) = \sum_{\nu=1}^{M} e^{j(\mu-1)\omega_\nu(k)} s_\nu(k) + n_\mu(k), \quad \mu = 1, \ldots, L \tag{65}$$

where

- $x_\mu(k)$ μth entry of $\mathbf{x}(k)$;
- $\omega_\nu(k)$ normalized angular frequencies (possibly time-varying) of the νth exponential signal component;
- $s_\nu(k)$ complex amplitude of this component;
- $n_\mu(k)$ additive noise term.

The temporal sequences $s_\nu(k)$ $(\nu = 1, \ldots, M)$ and $n_\mu(k)$ $(\mu = 1, \ldots, L)$ are mutually independent, strictly white noise processes, each time sample having a complex, zero-mean circular Gaussian probability density function. The common variance of the $n_\mu(k)$ is set to 1, whereas the variances of the $s_\nu(k)$ are specified by the signal-to-noise ratio (SNR) parameters, SNR_ν, $\nu = 1, \ldots, M$, respectively. The above model is fairly general and can be used for both spatial and temporal spectral analysis applications.

In the experiments, data generated according to this model are processed with two different versions of the generic signal-subspace tracker presented in Table V, respectively, based on (and simply referred to as) PROTEUS-1 and 2. The data are also process with Karasalo's [23] and the NASVD [25] algorithms for comparison. The former, which is referred to

[7] A matrix $\Delta_3(k) = [\delta_{3,ij}(k)]$ satisfying (61) can always be found by forcing $\det(U(k)) = 1$; furthermore, $\Delta_3(k)$ is skew-Hermitian, i.e., $\delta_{3,ij}(k) = -\delta^*_{3,ji}(k)$ [42].

[8] As pointed out by a reviewer, parametrizations of orthogonal matrices apparented to (33) have been used in [37] and [54] to counteract error accumulations by updating only the associated rotation parameters. In these works, however, the eigenvectors are not directly available and must be computed off line [in $O(LM^2)$ operations].

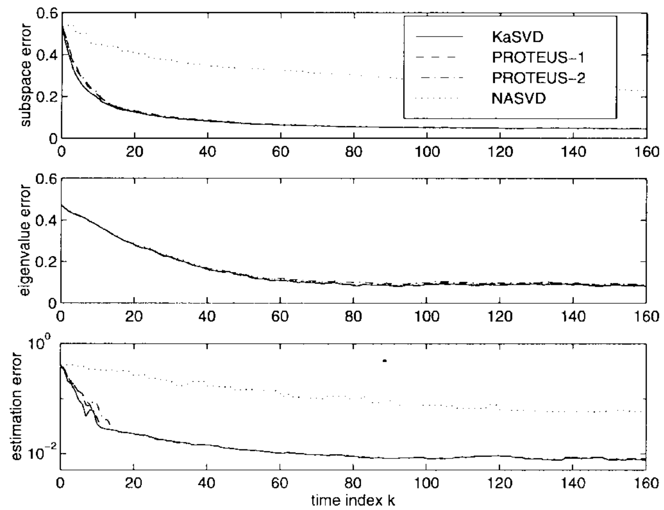

Fig. 1. Initial convergence of KaSVD, PROTEUS-1 and 2, and NASVD algorithms. Subspace error (top), eigenvalue error (middle), and frequency estimation error (bottom) versus discrete-time index k for SNR $= 15$ dB, and $\epsilon = 0.025$.

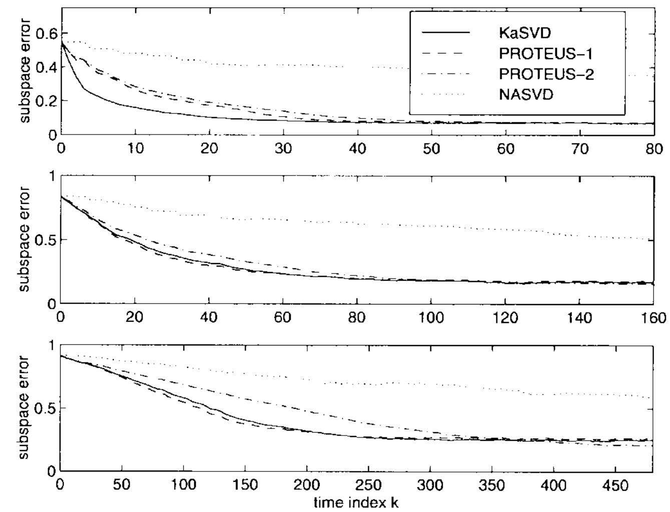

Fig. 2. Initial convergence of the trackers for various SNR and ϵ. Subspace error versus discrete-time k for SNR $= 15$ dB, $\epsilon = 0.05$ (top), SNR $= 5$ dB, $\epsilon = 0.025$ (middle) and SNR $= 0$ dB, $\epsilon = 0.0125$ (bottom).

as KaSVD for convenience, is an $O(LM^2)$ approach but uses an exact (iterative) SVD algorithm of complexity $O(M^3)$ for the deflated problem and is thus of limited practical value for large M. Nevertheless, it provides a useful benchmark since its performance is almost identical to an exact EVD of the sample covariance matrix (10). NASVD is also of interest since, similarly to PROTEUS-2, it is designed to track the complete signal-subspace EVD (i.e., eigenvalues and eigenvectors) in $O(LM)$ operations per iteration. In our implementation of NASVD, we use a single diagonal sweep made up of $K-2$ outer and one inner rotations, as suggested in [25], so that both NASVD and PROTEUS-2 use a total of $2K+O(1)$ Givens rotations.[9] In all the algorithms, the correct number of sources M is assumed.

The following performance measures are computed:

i) the subspace error $||\mathcal{P}[U_s(k)] - \mathcal{P}[Q_s(k)]||_2$, where $U_s(k)$ and $Q_s(k)$ are the $L \times M$ matrices of estimated and true signal-subspace eigenvectors, respectively, and $\mathcal{P}[\cdot]$ denotes the projector on the column span of its matrix argument;

ii) the eigenvalue error $\sum_{i=1}^{K} |\gamma_i(k) - \lambda_i(k)| / \sum_{i=1}^{K} \lambda_i(k)$;

iii) the orthonormality error $||U_s(k)^H U_s(k) - I_M||_F/\sqrt{M}$, where $||\cdot||_F$ denotes the Frobenius norm;

iv) the frequency estimation error $\frac{1}{M}\sum_{\nu=1}^{M} |\hat{\omega}_\nu(k) - \omega_\nu(k)|$, where $\hat{\omega}_\nu(k)$ denotes the root-MUSIC [1] estimate of $\omega_\nu(k)$.

Results for initial convergence, numerical stability, and tracking behavior are presented below.

A. Initial Convergence

We consider a stationary scenario with $L = 10$, $M = 4$, $\omega_\nu(k) = 0.0, 0.25, 1.0$, and 1.25, and SNR$_\nu$ = SNR, for $\nu = 1, \ldots, 4$. In the first series of results, the algorithms are initialized with the EVD of a low-rank sample covariance matrix obtained by averaging the outer products of the first M data vectors. Fig. 1 shows the subspace, eigenvalue, and frequency estimation errors (all averaged over 40 independent runs) for SNR $= 15$ dB and $\epsilon = 0.025$. The convergence rates and residual errors of the PROTEUS algorithms are almost identical to that of KaSVD. Note that in the case of PROTEUS-2, this is so even though the first K true eigenvalues are not particularly well separated (true eigenvalues: 648, 471, 113, 37.9, 1.0). Regarding NASVD, its initial convergence rate is significantly lower than the other algorithms, which is consistent with the results reported in [34] and [35]. Although it is possible to improve the convergence of NASVD by making multiple sweeps, this generally results in a significant increase in its computational complexity [in some of our experiments, $O(M)$ sweeps were needed to obtain a performance comparable to KaSVD, thus making NASVD an $O(LM^2)$ algorithm].

Fig. 2 demonstrates the effects of using different SNR and ϵ (only the subspace error is shown). The top curve was obtained with SNR $= 15$ dB and $\epsilon = 0.05$, which corresponds to a very short exponential window of effective length 20 samples. Due to second-order effects, the convergence of the PROTEUS algorithms is now slowed down slightly as compared with KaSVD. The middle curve corresponds to SNR $= 5$ dB and $\epsilon = 0.025$ (true eigenvalues: 65.7, 48.0, 12.2, 4.7, 1.0), and finally, the bottom curve corresponds to SNR $= 0$ dB and $\epsilon = 0.0125$ (true eigenvalues: 21.5, 15.9, 4.5, 2.2, 1.0). At lower SNR, it is necessary to use a longer exponential window to average out the effects of the noise; thus, we have the smaller value of ϵ used in the last case.

To investigate convergence in the absence of *a priori* knowledge, another series of experiments was conducted with the same simulation parameters but this time using random initial conditions obtained by computing the EVD of the sample covariance matrix of (scaled) noise-only vectors. Typical results for the subspace error are shown in Fig. 3 for

[9] We have generally found that PROTEUS-2 requires less overhead and memory space than NASVD due to the need to maintain and update an upper triangular matrix in the latter.

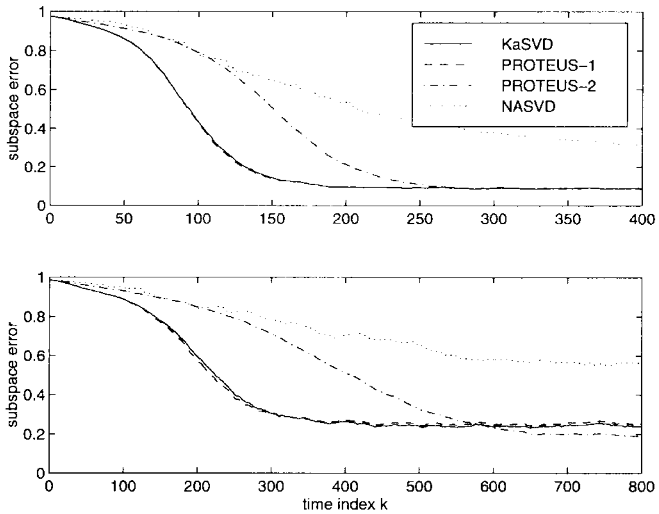

Fig. 3. Initial convergence of the trackers using random initial conditions. Subspace error versus discrete-time k for SNR $= 10$ dB, $\epsilon = 0.025$ (top) and SNR $= 0$ dB, $\epsilon = 0.0125$ (bottom).

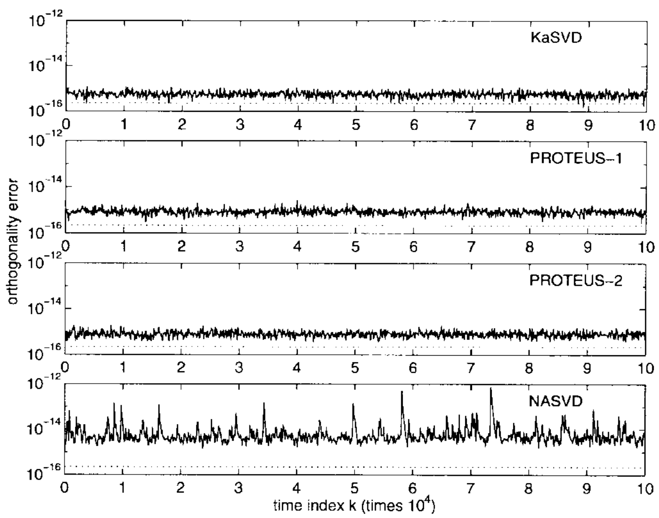

Fig. 4. Investigating numerical stability of the various trackers. Orthogonality error versus discrete-time k for SNR $= 15$ dB and $\epsilon = 0.025$.

SNR $= 10$ dB and $\epsilon = 0.025$ (top) and for SNR $= 0$ dB and $\epsilon = 0.0125$ (bottom). Generally, we have found that whenever some form of convergence is possible with KaSVD, the PROTEUS algorithms also converge to the same error levels but possibly with an extra delay that depends on the specific simulation scneario.

B. Numerical Stability

To test the numerical stability of the various subspace trackers, we used the same scenario as above but let the algorithms run over very long periods of time. Note that *no* additional reorthonormalization mechanisms were used with the PROTEUS and NASVD algorithms in order to characterize their intrinsic error buildup properties. In the case of KaSVD, the situation is different since an exact, costly SVD routine is used that produces orthonormal eigenvectors at each iteration.

Fig. 4 shows the orthonormality error of the various trackers versus discrete-time k for SNR $= 15$ dB and $\epsilon = 0.025$ (single run). The doted horizontal line in each subplot represents the relative accuracy of numbers on our computer system, i.e. 2^{-52}. The sample mean and standard deviation of the orthonormality error curves in Fig. 5 appear in Table VI. The PROTEUS-based trackers show no apparent buildup of numerical errors with time and maintain a similar level of orthogonality as the exact KaSVD approach, which uses the Matlab `svd()` routine [29]. The other performance measures (which are not shown) are also stable and remain at the levels attained after initial convergence. Numerous experiments of this type generally suggest that in the computer environment used here (Matlab, 32-bit UNIX workstation), no additional reorthonormalization mechanism is necessary to ensure the numerical stability of the new algorithms. NASVD also appears to be numerically stable, although its error level is higher by an order of magnitude due to some initial buildup.

TABLE VI
SAMPLE MEAN AND STANDARD DEVIATION OF ORTHONORMALITY ERROR FOR KaSVD, PROTEUS-1, PROTEUS 2, AND NASVD TRACKERS

Algorithm	Mean	Standard. dev.
KaSVD	5.73×10^{-16}	1.71×10^{-16}
PROTEUS-1	8.71×10^{-16}	2.60×10^{-16}
PROTEUS-2	8.40×10^{-16}	2.62×10^{-16}
NASVD	1.01×10^{-14}	3.50×10^{-14}

C. Tracking Capability

To illustrate the tracking capability of the PROTEUS algorithms, we first consider the following scenario: $L = 10$, $M = 4$, and $\omega_\nu(k)$ time varying according to some simple trigonometric functions. In Fig. 6, we show the true angular frequencies $\omega_\nu(k)$ and their root-MUSIC estimates (single run) obtained from the KaSVD, PROTEUS, and NASVD trackers for $\epsilon = 0.025$ and SNR $= 15$ dB. The tracking performance of the PROTEUS algorithms is seen to be comparable with that of KaSVD; NASVD is not as good. The subplot in the lower right corner of Fig. 6 compares the signal subspace errors of PROTEUS-2 and NASVD.

To evaluate the tracking ability of the PROTEUS algorithms under severe conditions, we consider a subspace rotation test similar to that proposed in [11], in which the true signal subspace undergoes a sudden large change corresponding to a 90° rotation. In our implementation of this test, we use $L = 10$, $M = 4$, and $\omega_\nu(k) = \mathrm{sgn}(k)(2\pi\nu/L)$ for $\nu = 1, \ldots, 4$, where $\mathrm{sgn}(k) = 1$ for $k > 0$ and -1 for $k \leq 0$. We also allow sufficient time for the algorithms to converge prior to the change at time $k = 1$. Fig. 6 shows the subspace errors of the various algorithms (10 run average) for SNR $= 15$ dB and $\epsilon = 0.05$ (top) and SNR $= 5$ dB and $\epsilon = 0.025$ (bottom).

VI. SUMMARY AND CONCLUSIONS

A parametric perturbation approach was used to derive new EVD updating schemes for the rank-one modification problem. Called PROTEUS-1 and 2, these schemes use sequences of plane rotations to update the eigenvectors so that the latter remain orthonormal. New subspace trackers based on the PROTEUS schemes were derived, and their properties investigated

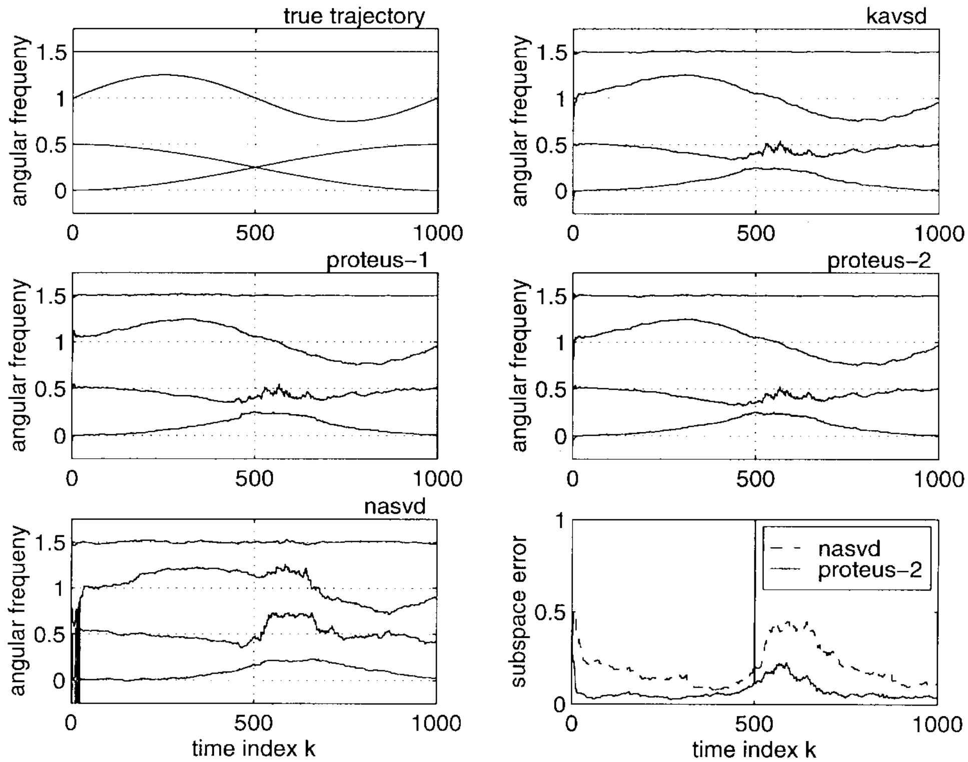

Fig. 5. Illustrating tracking capability of various algorithms. True frequencies versus discrete-time k (top left). Root-MUSIC frequency estimates versus k for KaSVD subspace (top right). Same for PROTEUS-1 (middle left). Same for PROTEUS-2 (middle right). Same for NASVD (bottom left). Comparison of subspace errors for PROTEUS-2 and NASVD (bottom right).

via simulations. One particular algorithm based on PROTEUS-2 can track all the signal subspace EVD components in only $O(LM)$ operations while achieving a performance comparable with an exact EVD approach and maintaining perfect orthonormality in the eigenvectors. Some conclusive remarks follow.

Although our formulation of the subspace tracking problem centered around the EVD of the sample covariance matrix in (10), we point out that the PROTEUS schemes and the associated trackers of Section IV can be derived using an SVD formulation based on the data matrix [9]. Indeed, the PROTEUS trackers are truly data matrix algorithms (i.e., squaring operations occur only in the computations of the rotation angles).

It is interesting to compare the conventional perturbation approach used in [6] and the parametric one used here. The former can be viewed as an unconstrained linearization of (18) and (19) with respect to $\Delta V = V - I_K$, where ΔV is a small arbitrary matrix. The latter can be viewed as a constrained linearization of (18) and (19) within the lower dimensional space of skew-symmetric matrices $\Theta = \log V$, which is the tangent space of the group $SO(K)$ at the group identity (i.e., the Lie algebra). In addition to preserving orthonormality, an important advantage of this approach is the existence of efficient realizations of (23) via plane rotations; this was not apparent in [6].

Further simplification of PROTEUS-1 into an effective $O(LK)$ scheme, as was done in Section III-C to derive PROTEUS-2, is not a trivial task. For instance, another

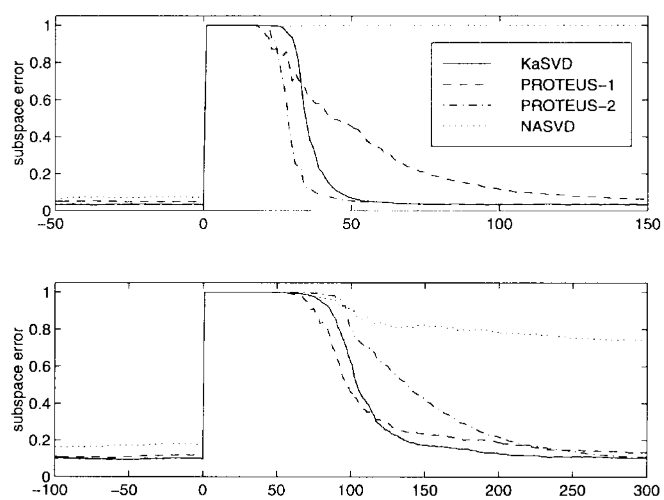

Fig. 6. Ability of the various trackers to recover from a sudden, 90° rotation in the true signal subspace. Subspace error versus discrete-time k for SNR $= 15$ dB, $\epsilon = 0.025$ (top) and for SNR $= 0$ dB, $\epsilon = 0.0125$ (bottom).

$O(LK)$ scheme can be derived from (34) by setting θ_{ij} to zero for $|j - i| > l$, where l is some fixed small integer [8]. However, for small values of l, this scheme does not perform as well as PROTEUS-2.

The structured sequences of plane rotations used in the PROTEUS algorithms involve a large number of local matrix operations that exhibit a high degree of modularity and concurrency. As such, they are potentially well suited for parallel implementation on special-purpose circuitry (e.g., systolic array implementation using CORDIC processors). Further significant reductions in processing time could be achieved in this way.

Appendix A
Proof of (31)

Without loss of generality, consider the case $K = 2$, and let

$$\Psi = \begin{bmatrix} 0 & \theta \\ -\theta & 0 \end{bmatrix}. \tag{66}$$

Observe that for $n = 0, 1, 2, \ldots$

$$\Psi^{2n} = (-1)^n \begin{bmatrix} \theta^{2n} & 0 \\ 0 & \theta^{2n} \end{bmatrix}$$
$$\Psi^{2n+1} = (-1)^n \begin{bmatrix} 0 & \theta^{2n+1} \\ -\theta^{2n+1} & 0 \end{bmatrix}. \tag{67}$$

Therefore, we have

$$\begin{aligned} \exp(\Psi) &= I + \Psi + \frac{1}{2!}\Psi^2 + \frac{1}{3!}\Psi^3 + \cdots \\ &= \begin{bmatrix} 1 - \frac{1}{2!}\theta^2 + \cdots & \theta - \frac{1}{3!}\theta^3 + \cdots \\ -(\theta - \frac{1}{3!}\theta^3 + \cdots) & 1 - \frac{1}{2!}\theta^2 + \cdots \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} = G_{12}(\theta). \end{aligned} \tag{68}$$

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] A. J. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1983, pp. 336–339.
[2] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. New York: Springer-Verlag, 1990.
[3] G. Bienvenu and L. Kopp, "Adaptivity to background noise spatial coherence for high resolution passive methods," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1980, pp. 307–310.
[4] C. H. Bishof and G. M. Shroff, "On updating signal subspaces," *IEEE Trans. Signal Processing*, vol. 40, pp. 96–105, Jan. 1992.
[5] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, "Rank-one modification of the symmetric eigenproblem," *Numer. Math.*, vol. 31, pp. 31–48, 1978.
[6] B. Champagne, "Adaptive eigendecomposition of data covariance matrices based on first-order perturbations," *IEEE Trans. Signal Processing*, vol. 42, pp. 2758–2770, Oct. 1994.
[7] ——, "Adaptive subspace tracking using a constrained linearization approach," in *Proc. Int. Conf. Neural Networks Signal Process.*, Nanjing, China, Dec. 1995, pp. 1083–1086.
[8] B. Champagne and Q.-G. Liu, "A new family of EVD tracking algorithms using Givens rotations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, 1996, vol. 5, pp. 2539–2542.
[9] B. Champagne, "SVD updating via constrained perturbations with application to subspace tracking," in *Proc. 30th Asilomar Conf.*, Pacific Grove, CA, 1996, pp. 1379–1385.
[10] H. Chen, T. K. Sarkar, S. A. Dianat, and J. D. Brulé, "Adaptive spectral estimation by the conjugate gradient method," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 272–284, Apr. 1986.
[11] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78. pp. 1327–1343, Aug. 1990.
[12] R. D. DeGroat and R. A. Roberts, "Efficient, numerically stabilized rank-one eigenstructure updating," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 301–316, Feb. 1990.
[13] R. D. DeGroat, "Noniterative subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, pp. 571–577, Mar. 1992.
[14] R. D. DeGroat, E. M. Dowling, H. Ye, and D. A. Linebarger, "Spherical subspace tracking for efficient, high performance adaptive signal processing applications," *Signal Process.*, vol. 50, pp. 101–121, 1996.
[15] J. Dehaene, "Continuous-time matrix algorithms, systolic algorithms and adaptive neural networks," Ph.D. dissertation, Faculty Appl. Sci., Katholiecke Univ. Leuven, Leuven, Belgium, 1995.
[16] J. P. Delmas, "A complex adaptive eigensubspace algorithm for DOE or frequency estimation and tracking," in *Proc. Euro. Conf. Signal Process.*, 1992, pp. 657–660.
[17] E. M. Dowling, L. P. Ammann, and R. D. DeGroat, "A TQR-iteration based adaptive SVD for real time angle and frequency tracking," *IEEE Trans. Signal Processing*, vol. 42, pp. 914–926, Apr. 1994.
[18] W. Ferzali and J. G. Proakis, "Adaptive SVD algorithm for covariance matrix eigenstructure computation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1990, pp. 2615–2618.
[19] Z. Fu and E. M. Dowling, "Conjugate gradient eigenstructure tracking for adaptive spectral estimation," *IEEE Trans. Signal Processing*, vol. 43, pp. 1151–1160, May 1995.
[20] D. R. Fuhrmann and B. Liu, "Rotational search methods for adaptive Pisarenko harmonic retrieval," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1550–1565, Dec. 1986.
[21] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, pp. 403–420, 1970.
[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd Ed. Baltimore, MD: John Hopkins Univ. Press, 1989.
[23] I. Karasalo, "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 8–12, Feb. 1986.
[24] J. Karhunen, "Adaptive algorithms for estimating eigenvectors of correlation type matrices," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1984, pp. 14.6.1–14.6.4.
[25] A. Kavčić and B. Yang, "Subspace tracking with adaptive threshold rank estimation," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 14, pp. 75–91, 1996.
[26] C. S. MacInnes and R. J. Vaccaro, "Tracking directions-of-arrival with invariant subspace updating," *Signal Process.*, vol. 50, pp. 137–150, 1996.
[27] G. Mathew, V. U. Reddy, and S. Dasgupta, "Adaptive estimation of eigensubspace," *IEEE Trans. Signal Processing*, vol. 43, pp. 401–411, Feb. 1995.
[28] G. Mathew and V. Reddy, "A quasi-Newton adaptive algorithm for generalized symmetric eigenvalue problem," *IEEE Trans. Signal Processing*, vol. 44, pp. 2413–2422, Oct. 1996.
[29] *Matlab Reference Guide for Unix Workstations*. Natick, MA, 1992.
[30] M. Moonen, P. V. Dooren, and J. Vandewalle, "A note on: Efficient numerically stabilized rank-one eigenstructure updating," *IEEE Trans. Signal Processing*, vol. 39, pp. 1911–1914, Aug. 1991.
[31] M. Moonen, P. V. Dooren, and J. Vandewalle, "A singular value decomposition updating algorithm for subspace tracking," *SIAM J. Matrix Anal. Appl.*, vol. 13, pp. 1015–1038, Oct. 1992.
[32] M. D. Ortigueira and M.-A. Lagunas, "Eigendecomposition versus singular value decomposition in adaptive array signal processing," *Signal Process.*, vol. 25, pp. 35–49, 1991.
[33] N. L. Owsley, "Adaptive data orthogonalization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1978, pp. 109–112.
[34] P. Pango and B. Champagne, "Accurate subspace tracking algorighms based on cross-space properties," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1997, pp. 3833–3836.
[35] D. J. Rabideau, "Fast, rank adaptive subspace tracking and applications," *IEEE Trans. Signal Processing*, vol. 44, pp. 2229–2244, Sept. 1996.
[36] V. U. Reddy, B. Egardt, and T. Kailath, "Least squares type algorithm for adaptive implementation of Pisarenko's harmonic retrieval method," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 399–405, June 1982.
[37] P. A. Regalia, "An adaptive unit norm filter with applications to signal analysis and Karhunen-Loeve transformations," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 646–649, May 1990.
[38] F. Rellich, *Perturbation Theory of Eigenvalue Problems*. New York: Gordon and Breach, 1969.
[39] K. C. Sharman, "Adaptive algorithms for estimating the complete covariance eigenstructure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Tokyo, Japan, 1986, pp. 1401–1404.
[40] R. Schmidt, "Multiple emitter location and signal parameter estimation," in *Proc. RADC Spectrum Estimation Workshop*, 1979, pp. 243–258.
[41] R. Schreiber, "Implementation of adaptive array algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1038–1045, Oct. 1986.

[42] B. Schutz, *Geometrical Methods of Mathematical Physics*, Cambridge, U.K.: Cambridge Univ. Press, 1980.

[43] S. Smith, "Linear and nonlinear conjugate gradient methods for adaptive processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, 1996, vol. 3, pp. 1834–1837.

[44] G. W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, pp. 1535–1541, June 1992.

[45] P. Strobach, "Low rank adaptive filtering," *IEEE Trans. Signal Processing*, vol. 44, pp. 2932–2947, Dec. 1996.

[46] P. A. Thompson, "An adaptive spectral analysis technique for unbiased frequency estimation in the presence of white noise," in *Proc. 13th Asilomar Conf. Circuits, Syst., Comput.*, 1979, pp. 529–533.

[47] R. Vaccaro, "On adaptive implementations of Pisarenko harmonic retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1984, p. 6.1.1.

[48] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 387–392, Apr. 1985.

[49] G. Xu, H. Zha, G. Golub, and T. Kailath, "Fast algorithms for updating signal subspaces," *IEEE Trans. Circuits Syst. II*, vol. 41, pp. 537–549, Aug. 1994.

[50] B. Yang and F. Gersemsky, "Adaptive direction-of-arrival estimation based on rank and subspace tracking," in *SVD and Signal Processing, III*, M. Moonen and B. De Moor Eds. Amsterdam, The Netherlands: Elsevier, 1995, pp. 287–294.

[51] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing*, vol. 43, pp. 95–107, Jan. 1995.

[52] J.-F. Yang and M. Kaveh, "Adaptive eigensubspace algorithms for direction or frequency estimation and tracking," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 241–251, Feb. 1988.

[53] K.-B. Yu, "Recursive updating the eigenvalue decomposition of a covariance matrix," *IEEE Trans. Signal Processing*, vol. 39, pp. 1136–1145, May 1991.

[54] F. Vanpoucke and M. Moonen, "Factored spherical subspace tracking," *Integration, VLSI J.*, vol. 20, pp. 3–21, Dec. 1995.

Benoît Champagne was born in Joliette, P.Q., Canada, on January 13, 1961. He received the B.Ing. degree in engineering physics from the Ecole Polytechnique of Montréal, Montréal, P.Q., Canada, in 1983, the M.Sc. degree in physics from the University of Montréal in 1985, and the Ph.D degree in electrical engineering from the University of Toronto, Toronto, Ont., Canada, in 1990.

In June 1990, he joined INRS-Télécommunications, Université du Québec, Verdun, P.Q., where he is presently an Associate Professor. Since September 1994, he has also been an Adjunct Professor with the Department of Electrical Engineering at McGill University, Montréal. His current research interests are in the areas of statistical signal processing and adaptive filtering, including subspace tracking, array processing and beamforming, subband adaptive filtering, and echo cancellation.

Qing-Guang Liu was born on June 20, 1964, in Henan, China. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Xi'an Jiaotong University, China, in 1985, 1988, and 1991, respectively.

From October 1991 to October 1993, he was a Postdoctoral Researcher and a Lecturer with Tsinghua University, China. He was with INRS-Telecommunications, Montréal, P.Q., Canada, as a Postdoctoral Researcher from March 1994 to September 1995 and as a Research Associate from September 1995 to March 1997. He is now a Member of Scientific Staff with Nortel, Montréal. His main interests include adaptive filtering, adaptive array, echo cancellation, and speech processing.