Correspondence

On the Asymptotic Convergence and Numerical Stability of the Proteus EVD Trackers

Benoît Champagne

Abstract—**In this correspondence, the asymptotic convergence and numerical stability of the recently introduced subspace tracking algorithms PROTEUS-1 and - 2 are investigated by means of the ODE method. It is shown that 1) under weak conditions, both algorithms globally converge with probability one to the desired EVD components of the data covariance matrix, and 2) they have a built-in mechanism that prevents deviation from orthonormality in the eigenvector estimates over long periods of operation, i.e., numerical stability.**

Index Terms—**Confergence analysis, EVO tracking, ODE method, subspace tracking.**

I. Introduction

Efficient, sequential estimation of various quantities pertaining to the eigenvalue decomposition (EVD) of a time-varying data covariance matrix is of paramount importance for adaptive implementation of subspace methods in real time. In a recent paper [2], new algorithms with low computational complexity are presented for tracking the signal-subspace (i.e., dominant) eigenvalues and orthonormal eigenvectors. Their derivation, which is based on the interpretation of the rank-one modification term in the sample covariance matrix update as a small perturbation, relies on a new type of constrained perturbation approach in which plane rotations are used to implement the first-order perturbation; therefore, eigenvector orthonormality is maintained during the update. This approach directly leads to a first algorithm (Proteus-1) with a complexity of $O(Nr^2)$ operations per iteration, where N and r, respectively, denote the dimensions of the data vector and the dominant subspace. A further approximation based on the assumption of widely separated eigenvalues leads to a second algorithm (Proteus-2) with complexity $O(Nr)$. Extensive computer simulations show that these algorithms converge to the desired EVD components under a wide range of experimental conditions with an estimation accuracy comparable to an exact approach. This is so for Proteus-2 even when the underlying assumption on the eigenvalues is not satisfied. Furthermore, the algorithms appear to be numerically robust, maintaining a level of orthogonality in the eigenvector estimates close to machine precision at all times.

In this correspondence, we investigate the asymptotic convergence and numerical stability of Proteus-1 and 2 using the ODE method. Specifically, we show that 1) both algorithms globally converge with probability one to the desired EVD components of the data covariance matrix, and 2) they have a built-in mechanism that prevents deviation from orthonormality in the eigenvector estimates over long periods of operation, i.e., numerical stability. In [3], DeGroat *et al.* use the ODE method to study the convergence of a related multilevel subspace tracker; however, eigenvector convergence is proved under the assumption of fixed, decreasingly ordered eigenvalue levels so that the results are not generally applicable to Proteus-1 and –2. In addition, potential deviation from orthonormality resulting from finite precision effects, which is a critical issue when considering practical implementations, is left out as an open problem in [3]. Our analysis, which revolves around an alternative ODE system and different Lyapunov functions, overcomes these restrictions and, thus, offers deeper insight into the operation of Proteus-1 and –2 and other related subspace trackers.

Manuscript received July 15, 1998; revised April 28, 1999. This work was supported 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 Prof. Victor A. N. Barroso.

The author is with the Department of Electrical and Computer Engineering, McGill University, Montreal, P.Q., Canada H3A 2A7 (e-mail: champagne@mcgill.ca).

Publisher Item Identifier S 1053-587X(00)00107-0.

II. Problem Formulation

A. Overview of the Proteus Algorithms

Let $\boldsymbol{x}_k \in \mathbb{C}^N$, $k \in \{1, 2, \cdots\}$ denote the time samples of a stochastic vector process with zero-mean and covariance matrix $R_k^o = E[\boldsymbol{x}_k \boldsymbol{x}_k^H]$. Assume that signal-related information is contained in a lower dimensional eigensubspace of R_k^o associated with the r ($1 \leq r < N$) largest (dominant) eigenvalues, which is known as the signal subspace; the orthogonal eigensubspace associated with the $N-r$ smallest (subdominant) eigenvalues is called the noise subspace. The specific versions of the Proteus algorithms under consideration here provide a computationally efficient means for recursively updating estimates of the r largest eigenvalues and corresponding orthonormalized eigenvectors of R_k^o, along with the arithmetic mean of the $N-r$ smallest eigenvalues, each time a new sample vector $\boldsymbol{x}_k$ becomes available.

Specifically, let $\lambda_{i,k}$ and $\boldsymbol{u}_{i,k}$ ($i = 1, \cdots, r$) denote the estimates of the signal-subspace eigenvalues and corresponding eigenvectors, and let $\lambda_{r+1,k}$ denote the representative noise eigenvalue estimate. In addition, let $U_{\mathcal{S},k} = [\boldsymbol{u}_{1,k}, \cdots, \boldsymbol{u}_{r,k}]$. On observation of $\boldsymbol{x}_k$, both Proteus-1 and –2 initially compute

$$\begin{aligned} \boldsymbol{\xi}_{\mathcal{S},k} &= U_{\mathcal{S},k-1}^H \boldsymbol{x}_k \\ \boldsymbol{x}_{\mathcal{N},k} &= \boldsymbol{x}_k - U_{\mathcal{S},k-1}\boldsymbol{\xi}_{\mathcal{S},k} \\ \boldsymbol{\xi}_k &= \left[\boldsymbol{\xi}_{\mathcal{S},k}^T, \|\boldsymbol{x}_{\mathcal{N},k}\|_2\right]^T \\ \boldsymbol{u}_{r+1,k-1} &= \boldsymbol{x}_{\mathcal{N},k}/\|\boldsymbol{x}_{\mathcal{N},k}\|_2 \end{aligned} \tag{1}$$

where the eigenvectors $\boldsymbol{u}_{i,k-1}$ ($i = 1, \cdots, r$) have been premultipled by complex scalars of unit magnitude so that the entries of $\boldsymbol{\xi}_{\mathcal{S},k}$ are real. After these computations, the eigenvectors are updated as

$$U_{\mathcal{S},k} = [U_{\mathcal{S},k-1}, \boldsymbol{u}_{r+1,k-1}]V_k[I_r, \mathbf{0}]^T \tag{2}$$

where the matrix V_k, which is different for the two algorithms, represents a product of plane rotations aimed at approximating an exponential matrix function as

$$V_k = \exp(\gamma\Theta_k) + O(\gamma^2) \tag{3}$$

where $\gamma > 0$ is a small gain parameter that controls the memory of the estimation, and $\Theta = [\theta_{ij}] \in \mathbb{R}^{(r+1)\times(r+1)}$ is a skew-symmetric matrix (i.e., $\Theta^T = -\Theta$) with upper diagonal entries

$$\theta_{ij,k} = \xi_{i,k}\xi_{j,k}/(\lambda_{j,k-1} - \lambda_{i,k-1}), \qquad 1 \leq i < j \leq r+1. \tag{4}$$

In Proteus-1, V_k is expressed as a product of $r(r+1)/2$ plane rotations so that the resulting algorithm complexity is $O(Nr^2)$. In the derivation of Proteus-2, it is assumed that the eigenvalues are widely separated, i.e., $\lambda_{i,k-1} \gg \lambda_{i+1,k-1}$ for $i = 1, \cdots, r$ so that (4) reduces to

$$\theta_{ij,k} = -\xi_{i,k}\xi_{j,k}/\lambda_{i,k-1}, \qquad 1 \le i < j \le r+1. \tag{5}$$

By exploiting (5), a much simplified approximation V_k in terms of only $2r-1$ plane rotations can be obtained, leading to an algorithm complexity of only $O(Nr)$. Note that since (2) is implemented with plane rotations, the columns of $U_{S,k}$ remain orthonormal at all time (assuming proper initialization and infinite precision arithmetic). In both algorithms, the eigenvalues are finally updated as

$$\lambda_{i,k} = \begin{cases} (1-\gamma)\lambda_{i,k-1} + \gamma|\xi_{i,k}|^2, & i = 1, \cdots, r \\ (1-\gamma)\lambda_{i,k-1} + \gamma|\xi_{i,k}|^2/(N-r), & i = r+1. \end{cases} \tag{6}$$

B. Algorithm Linearization

A central assumption in the *asymptotic* convergence analysis of a stochastic recursive algorithm with the ODE method is that of small gain so that the algorithm may be expressed in the linearized form

$$\phi_k = \phi_{k-1} + \gamma_k H(\phi_{k-1}, \boldsymbol{x}_k) + O(\gamma_k^2) \tag{7}$$

where

- ϕ_k sequence of parameter estimates recursively produced by the algorithm;
- $\gamma_k > 0$ small, possibly time-varying scalar gain;
- $H(\phi_{k-1}, \boldsymbol{x}_k)$ function (the so-called vector field) that defines, up to first-order terms in γ_k, how the parameter estimates ϕ_{k-1} are updated by the algorithms;
- $O(\gamma_k^2)$ residual second and higher order terms [1].

The parameter estimates produced by the Proteus algorithms are the matrix of eigenvector estimates $U_{S,k} = [\boldsymbol{u}_{1,k}, \cdots, \boldsymbol{u}_{r,k}]$ and the eigenvalue estimates $\lambda_{1,k}, \cdots, \lambda_{r+1,k}$; accordingly, we define $\phi_k = \{U_{S,k}, \lambda_{1,k}, \cdots, \lambda_{r+1,k}\}$. Using a similar notation, an arbitrary point in the parameter space $\mathcal{P}$ is represented by $\phi = \{U_S, \lambda_1, \cdots, \lambda_{r+1}\}$, where $U_S = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r] \in \mathbb{C}^{N\times r}$ and $\lambda_i \in \mathbb{R}$, and the vector field $H(\phi, \boldsymbol{x}_k)$ is conveniently structured as $H(.) = \{H_{U_S}(.), H_{\lambda_1}(.), \cdots, H_{\lambda_{r+1}}(.)\}$.

Making use of this notation and setting $\gamma \equiv \gamma_k$ in (3)–(6), both Proteus-1 and –2 can be expressed in the form (7), with $H_{U_S}(.)$ and $H_{\lambda_i}(.)$ $(i = 1, \cdots, r+1)$ given by

$$H_{U_s}(\phi, \boldsymbol{x}_k) = \left[U_S\mathcal{U}_S,\ I - U_S U_S^H\right]\left(L \otimes \boldsymbol{x}_k\boldsymbol{x}_k^H\right)[\mathcal{U}_S,\ \mathbf{0}]^H \tag{8}$$

$$H_{\lambda_i}(\phi, \boldsymbol{x}_k) = \boldsymbol{u}_i^H \boldsymbol{x}_k\boldsymbol{x}_k^H\boldsymbol{u}_i - \lambda_i, \qquad i = 1, \cdots, r \tag{9}$$

and

$$H_{\lambda_{r+1}}(\phi, \boldsymbol{x}_k) = \frac{1}{N-r}\,\mathrm{tr}\left[\left(I - U_S U_S^H\right)^2 \boldsymbol{x}_k\boldsymbol{x}_k^H\right] - \lambda_{r+1}. \tag{10}$$

In (8), $\mathcal{U}_S$ is an $r \times Nr$ block diagonal matrix obtained from the columns of U_S, i.e., $\mathcal{U}_S = \mathrm{diag}(\boldsymbol{u}_1^H, \cdots, \boldsymbol{u}_r^H)$, $\otimes$ is the Kronecker matrix product, and $L = [l_{ij}] \in \mathbb{R}^{(r+1)\times(r+1)}$ is a skew-symmetric matrix (i.e., $l_{ij} = -l_{ji}$) with algorithm dependent entries, as given below for $i < j$

$$l_{ij} = \begin{cases} 1/(\lambda_j - \lambda_i), & \text{for Proteus-1} \\ -1/\lambda_i, & \text{for Proteus-2.} \end{cases} \tag{11}$$

III. Asymptotic Convergence Analysis

We will model the sequence of input vectors $\{\boldsymbol{x}_k\}_{k=1}^{\infty}$ driving (7) as a stationary, temporally white, complex vector random process with zero-mean and positive-definite (true) covariance matrix R^o, i.e., $R^o = E[\boldsymbol{x}_k\boldsymbol{x}_k^H] > \mathbf{0}$. This model has the advantage of simplifying the discussion while preserving the essential aspects of the analysis; a more general class of models allowing for temporal correlation is described in [1]. We denote by $\lambda_1^o \ge \cdots \ge \lambda_N^o > 0$ the eigenvalues of R^o and by $\boldsymbol{u}_i^o$ $(i = 1, \cdots, N)$ a corresponding set of orthonormal eigenvectors; accordingly, the EVD of R^o is expressed as

$$R^o = U^o\Lambda^o U^{oH}, \qquad U^{oH}U^o = I \tag{12}$$

where $\Lambda^o = \mathrm{diag}(\lambda_1^o, \cdots, \lambda_N^o)$, and $U^o = [\boldsymbol{u}_1^o, \cdots, \boldsymbol{u}_N^o]$. To simplify the analysis, we assume that the true signal subspace eigenvalues are distinct, i.e., $\lambda_1^o > \cdots > \lambda_r^o > \lambda_{r+1}^o$; again, generalizations are possible.

A. The Algorithm's ODE

The ODE of a stochastic recursive algorithm expressed in the linearized form (7) is given by

$$\dot{\phi}(\tau) = h(\phi(\tau)), \qquad \phi(0) = \phi_0 \tag{13}$$

where

- $\tau \ge 0$ continuous-time variable (the fictitious time);
- $\phi(\tau)$ differentiable function of τ whose behavior is related to that of the recursive algorithm;
- ϕ_0 initial condition;

where the dot operator denotes time derivative, i.e., $d/d\tau$, and the function $h(\phi)$ (mean vector field) is given here by

$$h(\phi) = E[H(\phi, \boldsymbol{x}_k)]. \tag{14}$$

The ODE (13) is purely deterministic in nature; its solution $\phi(\tau)$ provides information about the ensemble of stochastic trajectories ϕ_k generated by the recursive algorithm (7). The connection between the discrete-time k and the fictitious time τ is achieved via the relation $\tau_k = \sum_{l=1}^{k}\gamma_l$.

In the present application of the ODE method, we let $\phi(\tau) = \{U_S(\tau), \lambda_1(\tau), \cdots, \lambda_{r+1}(\tau)\}$, and $h(.) = \{h_{U_S}(.), h_{\lambda_1}(.), \cdots, h_{\lambda_{r+1}}(.)\}$ so that (13) may be expressed as

$$\dot{U}_S(\tau) = h_{U_S}(\phi(\tau)) \tag{15}$$

$$\dot{\lambda}_i(\tau) = h_{\lambda_i}(\phi(\tau)), \qquad i = 1, \cdots, r+1. \tag{16}$$

The determination of the ODE now amounts to a computation of the expected values of the components of the vector field $H(\phi, \boldsymbol{x}_k)$ in (8)–(10). Observing that $E[\boldsymbol{x}_k\boldsymbol{x}_k^H] = R^o$, this yields the following nonlinear system of first-order time-invariant ODE's (i.e., autonomous dynamic system):

$$\dot{U}_S = \left[U_S\mathcal{U}_S,\ I_N - U_S U_S^H\right](L \otimes R^o)[\mathcal{U}_S,\ \mathbf{0}]^H \tag{17}$$

$$\dot{\lambda}_i = \boldsymbol{u}_i^H R^o\boldsymbol{u}_i - \lambda_i, \qquad i = 1, \cdots, r \tag{18}$$

and

$$\dot{\lambda}_{r+1} = \frac{1}{N-r}\,\mathrm{tr}\left[\left(I_N - U_S U_S^H\right)^2 R^o\right] - \lambda_{r+1} \tag{19}$$

where the dependence on the fictitious time τ has been omitted to simplify the notation.

B. Equilibrium and Invariant Sets of the ODE

We begin with the definition of the desired parameter set $\mathcal{D}^o$. Recall that the purpose of the Proteus algorithms is to track the r dominant eigenvalues and corresponding eigenvectors, along with the arithmetic mean of the $N-r$ subdominant eigenvalues of an underlying data covariance matrix. Here, a stationary signal model with true covariance matrix R^o (12) is assumed; accordingly, we define $\mathcal{D}^o$ as the set of all points $\phi = \{U_S, \lambda_1, \cdots, \lambda_{r+1}\} \in \mathcal{P}$, which is the parameter space, such that

$$\begin{aligned} R^o U_S &= U_S \Lambda_S \\ U_S^H U_S &= I_r \\ \lambda_i &= \lambda_{\pi_i}^o \qquad (i=1, \cdots, r) \end{aligned} \tag{20}$$

$$\lambda_{r+1} = \frac{1}{(N-r)} \sum_{i=r+1}^{N} \lambda_i^o \tag{21}$$

where $\Lambda_S = \text{diag}(\lambda_1, \cdots, \lambda_r)$, and π_i is an arbitrary permutation of the integers $\{1, \cdots, r\}$.

Recall that a subset $\mathcal{D}_E \subset \mathcal{P}$ is an equilibrium set of ODE (13) if $h(\phi) = 0$ for all $\phi \in \mathcal{D}_E$. Invoking the existence theorem for dynamic systems [5], it follows that $\phi(0) \in \mathcal{D}_E$ implies $\phi(\tau) = \phi(0)$ for all $\tau \geq 0$, where $\phi(\tau)$ is a solution of (13). More generally, an *invariant set* $\mathcal{D}_I$ of the ODE (13) is characterized by the property that if a point ϕ_o is in $\mathcal{D}_I$, then so is any solution $\phi(\tau)$ that passes through ϕ_o. The following lemmas identify important equilibrium and invariant sets of the ODE system (17)–(19); they point to desirable behaviors of the underlying recursive algorithms. Throughout this correspondence, $\phi(\tau) = \{U_S(\tau), \lambda_1(\tau), \cdots, \lambda_{r+1}(\tau)\}$ represents an arbitrary solution of (17)–(19).

Lemma 1: The desired parameter set $\mathcal{D}^o$ is an equilibrium set of the ODE system (17)–(19).

Proof: The right-hand sides of (17)–(19) are identically zero when $\phi \in \mathcal{D}^o$. □

Lemma 2: Let $\mathcal{M} = \{\phi \in \mathcal{P}: U_S^H U_S = I_r\}$. If $\phi(0) \in \mathcal{M}$, then $\phi(\tau) \in \mathcal{M}$ for all $\tau \geq 0$.

Proof: Using (17), we immediately obtain

$$\begin{aligned} &\frac{d}{d\tau}\left[U_S^H U_S\right] \\ &= \left[U_S^H U_S \mathcal{U}_S,\, \left(I_r - U_S^H U_S\right) U_S^H\right] (L \otimes R^o) [\mathcal{U}_S,\, \mathbf{0}]^H \\ &\quad + [\mathcal{U}_S,\, \mathbf{0}] \left(L^T \otimes R^o\right) \left[U_S^H U_S \mathcal{U}_S,\, \left(I_r - U_S^H U_S\right) U_S^H\right]^H. \end{aligned} \tag{22}$$

Now, if $U_S^H U_S = I_r$, (22) reduces to $d/d\tau[U_S^H U_S] = [\mathcal{U}_S, \mathbf{0}][(L + L^T) \otimes R^o][\mathcal{U}_S, \mathbf{0}]^H = \mathbf{0}$, where the last equality follows from the skew-symmetric property of L. Thus, the condition $U_S^H U_S = I_r$ defines an equilibrium state of the nonautonomous system (22), and therefore, invoking the existence theorem for the dynamic system, $U_S^H U_S|_{\tau=0} = I_r$ implies $U_S^H U_S = I_r$ for all $\tau \geq 0$. □

Lemma 3: For $i = 1, \cdots, r$, $U_S(0)^H \boldsymbol{u}_i^o \neq \mathbf{0}$ $(=\mathbf{0})$ implies $U_S(\tau)^H \boldsymbol{u}_i^o \neq \mathbf{0}$ $(=\mathbf{0})$ for all $\tau \geq 0$.

Proof: Define $\boldsymbol{v} = U_S^H \boldsymbol{u}_i^o$. Using (17) and (12), straightforward manipulations yield

$$\dot{\boldsymbol{v}} = -[\mathcal{U}_S,\, \mathbf{0}] (L \otimes R^o) [\mathcal{U}_S,\, \mathbf{0}]^H \boldsymbol{v} + \Omega \left(U_S^H R^o U_S - \lambda_i^o I_r\right) \boldsymbol{v} \tag{23}$$

where $\Omega = \text{diag}(l_{1,r+1}, \cdots, l_{r,r+1})$. Clearly, $\boldsymbol{v} = \mathbf{0}$ is an equilibrium point of this nonautonomous system (i.e., $\boldsymbol{v} = \mathbf{0} \Rightarrow \dot{\boldsymbol{v}} = \mathbf{0}$). Thus, $\boldsymbol{v}(0) \neq \mathbf{0}$ $(=\mathbf{0})$ implies $\boldsymbol{v}(\tau) \neq \mathbf{0}$ $(=\mathbf{0})$ for all $\tau \geq 0$. □

Lemma 4: For $i = 1, \cdots, r+1$, the initial condition $\lambda_i(0) > 0$ implies $\lambda_i(\tau) > 0$ for all $\tau \geq 0$.

Proof: The solutions to (18) and (19) can be expressed in the form $\lambda_i(\tau) = e^{-\tau}[\int_0^\tau f_i(t) e^t \, dt + \lambda_i(0)]$, where $f_i(\tau) = \boldsymbol{u}_i^H R^o \boldsymbol{u}_i$ $(i = 1, \cdots, r)$, and $f_{r+1}(\tau) = (1/(N-r)) \text{tr}[(I_N - U_S U_S^H)^2 R^o]$. Since $R^o > 0$, the functions $f_i(\tau) > 0$ for all $\tau \geq 0$; accordingly, $\lambda_i(0) > 0$ implies $\lambda_i(\tau) > 0$ for all $\tau \geq 0$. □

C. Stability Analysis by Lyapunov's Method

We note the following.

i) In infinite precision arithmetic, the eigenvector estimates produced by the Proteus algorithms remain orthonormal at all times if properly initialized, that is, $\phi_0 \in \mathcal{M}$ implies $\phi_k \in \mathcal{M}$ for all $k \geq 0$.

ii) According to Lemma 2, the associated ODE system (17)–(19) propagates orthonormal eigenvectors, i.e., $\phi(0) \in \mathcal{M}$ implies $\phi(\tau) \in \mathcal{M}$ for all $\tau \geq 0$.

iii) According to (20), the desired parameter set $\mathcal{D}^o \subset \mathcal{M}$.

Based on these observations, we conclude that in the stability analysis of the equilibrium set $\mathcal{D}^o$ of the ODE system (17)–(19), we may at first limit our attention to trajectories that are entirely confined to the manifold $\mathcal{M}$, i.e., $\phi(\tau) \in \mathcal{M}$ for all $\tau \geq 0$. Restricting the domain of definition of (17)–(19) in this manner yields the useful identities

$$\dot{U}_S^H U_S + U_S^H \dot{U}_S = 0, \quad P_\mathcal{N}^2 = P_\mathcal{N}^H = P_\mathcal{N}, \quad P_\mathcal{N} U_S = 0 \tag{24}$$

where $P_\mathcal{N} \equiv I - U_S U_S^H$ now defines an orthogonal projector on the (estimated) noise subspace.

Define the distance between an arbitrary point ψ and a subset $\mathcal{D} \subset \mathcal{P}$ of the parameter space as $\text{dist}(\psi, \mathcal{D}) = \inf_{\psi_o \in \mathcal{D}} \|\psi - \psi_o\|_F$. Recall that an equilibrium set $\mathcal{D}_E$ of the ODE system (17)–(19) is *stable* if for every choice of $\epsilon > 0$, there exists $\delta > 0$ such that $\text{dist}(\phi(\tau), \mathcal{D}_E) < \epsilon$ for all $\tau \geq 0$ whenever $\text{dist}(\phi(0), \mathcal{D}_E) < \delta$. $\mathcal{D}_E$ is *asymptotically stable* if it is stable and if there exists $\delta' > 0$ such that $\lim_{\tau\to\infty} \text{dist}(\phi(\tau), \mathcal{D}_E) = 0$ whenever $\text{dist}(\phi(0), \mathcal{D}_E) < \delta'$. The *domain of attraction* of $\mathcal{D}_E$ is then the set $\mathcal{D}_A$, such that $\phi(0) \in \mathcal{D}_A$ implies $\lim_{\tau\to\infty} \text{dist}(\phi(\tau), \mathcal{D}_E) = 0$.

Theorem 5: Let $l_{ij} = 1/(\lambda_j - \lambda_i)$ for $i < j$ in (11) (i.e., Proteus-1), and assume that $\lambda_{r+1}^o = \cdots = \lambda_N^o$. Then, $\mathcal{D}^o$ is an asymptotically stable equilibrium set of the ODE system (17)–(19) over the manifold $\mathcal{M}$. Its domain of attraction is $\mathcal{D}_{A_1} = \{\phi \in \mathcal{M}: U_S^H \boldsymbol{u}_i^o \neq \mathbf{0}$ for $1 \leq i \leq r\}$.

Proof: As a candidate Lyapunov function, consider

$$\begin{aligned} V_1(\phi) &= \tfrac{1}{2}\|R_1(\phi) - R^o\|_F^2, \\ R_1(\phi) &= U_S \Lambda_S U_S^H + \lambda_{r+1}\left(I_N - U_S U_S^H\right). \end{aligned} \tag{25}$$

First, invoking [4, Th. 7.4.51] in [4], it can be shown that the global minimum of $V(\phi)$ over $\mathcal{D}_{A_1}$ is attained on $\mathcal{D}^o$: $V(\phi) = 0$ for $\phi \in \mathcal{D}^o$ and $V(\phi) > 0$ for $\phi \in \overline{\mathcal{D}}^o \cap \mathcal{D}_{A_1}$. Next, we must study the sign of the time derivative of $V(\tau) \equiv V(\phi(\tau))$ along an arbitrary trajectory $\phi(\tau)$ of the ODE system (17)–(19) subject here to the restriction that $\phi(\tau) \in \mathcal{M}$. A lengthy derivation making use of (24) yields

$$\begin{aligned} \dot{V}_1(\tau) = &-\sum_{i=1}^{r} \dot{\lambda}_i^2 - (N-r)\dot{\lambda}_{r+1}^2 \\ &- 2\sum_{1\leq i<j\leq r} \left|\boldsymbol{u}_i^H R^o \boldsymbol{u}_j\right|^2 - 2\|P_\mathcal{N} R^o U_S\|_F^2. \end{aligned} \tag{26}$$

Clearly, $\dot{V}_1(\tau) \leq 0$. Let $\phi \in \mathcal{D}_{A_1}$ be such that $\dot{V}_1 = 0$. Setting $\|P_\mathcal{N} R^o U_S\|_F = 0$ yields $R^o U_S = U_S T$ for some matrix T, i.e., the column span of U_S is an invariant subspace of R^o; in $\mathcal{D}_{A_1}$, this implies that the columns of U_S form an orthonormal basis of the signal subspace. Under this condition, setting $|\boldsymbol{u}_i^H R^o \boldsymbol{u}_j| = 0$ for all $i < j$ implies that the columns of U_S are the individual signal subspace eigenvectors. Then, setting $\dot{\lambda}_i = 0$ $(i = 1, \cdots, r+1)$ and using (18) and (19), we conclude that $\phi \in \mathcal{D}^o$. In summary, $\dot{V}_1(\tau) = 0$ for $\phi \in \mathcal{D}^o$, and $\dot{V}_1(\tau) < 0$ for $\phi \in \overline{\mathcal{D}}^o \cap \mathcal{D}_{A_1}$. Finally, we note that

$V(\phi) \to \infty$ as $\phi \to \infty$. Invoking [5, Th. 13.IX] and Lemma 3, we conclude that $\mathcal{D}^o$ is asymptotically stable with its domain of attraction given by $\mathcal{D}_{A_1}$. □

Invoking basic theoretical results from [1], the convergence of Proteus-1 to the desired parameter set $\mathcal{D}^o$ is implied by Theorem 5. Specifically, for the case of decreasing gain, i.e., $\sum_{k=1}^{\infty} \gamma_k^\alpha < \infty$ for some $\alpha > 1$ and $\sum_{k=1}^{\infty} \gamma_k = +\infty$, the theory guarantees that ϕ_k converges to $\mathcal{D}^o$ with probability one under the weak condition that $U_{S,k}$ enters a compact subset of $\mathcal{D}_{A_1}$ infinitely often with probability one. Note that a similar condition is required for the convergence of other subspace trackers, e.g., [6]. Here, this condition may be justified on the basis that for Gaussian input vectors $\boldsymbol{x}_k$ and $\lambda_{r+1}^o > 0$, $U_{S,k-1}^H \boldsymbol{u}_i^o = \mathbf{0}$ for some $i \in \{1, \cdots, r\}$ implies $E[\|U_{S,k}^H \boldsymbol{u}_i^o\|^2] \geq \gamma_k^2 r(\lambda_{r+1}^o/\lambda_1^o)^2 > 0$. For the case of fixed gain, i.e., $\gamma_k = \gamma$, the theory asserts that ϕ_k remains within a certain distance of $\mathcal{D}^o$, for all $\delta > 0$, $\limsup_{k\to\infty} \Pr[\mathrm{dist}(\phi_k, \mathcal{D}^o) > \delta] \leq C(\gamma, \delta)$, with $\lim_{\gamma \to 0} C(\gamma, \delta) = 0$.

Insight can be gained into the convergence of Proteus-1 from Theorem 5. According to (25) and (26), the algorithm seeks the global minimum of an "error surface," which is defined as the squared error between the true covariance matrix R^o and a sphericalized estimate of the latter, i.e., $R_1(\phi)$. Another interesting aspect is the fact that no specific ordering of the eigenvalues $\lambda_i(\tau)$ $(i = 1, \cdots, r+1)$ is assumed. Accordingly, Proteus-1 converges to the desired parameter set $\mathcal{D}^o$, regardless of the initial eigenvalue ordering; this has been confirmed experimentally. Note that in the case $N - r \geq 2$, Theorem 5 assumes that the noise-subspace eigenvalues are identical; however, at the expense of more elaborate developments, this constraint may be relaxed to accommodate the case of a colored background noise.[1]

Theorem 6: Let $l_{ij} = -1/\lambda_i$ for $i < j$ in (11) (i.e., Proteus-2). Then, $\mathcal{D}^o$ is an asymptotically stable equilibrium set of the ODE system (17)–(19) over the manifold $\mathcal{M}$. Its domain of attraction is $\mathcal{D}_{A_2} = \{\phi \in \mathcal{M}\colon U_S^H \boldsymbol{u}_i^o \neq \mathbf{0}$ for $1 \leq i \leq r$ and $\lambda_i > 0$ for $i = 1, \cdots, r+1\}$.

Proof: Stability of the desired eigenvectors is shown as in Theorem 5 using the Lyapunov function $V_2(\phi) = 1/2\|R_2(\phi) - R^o\|_F^2$, where $R_2(\phi) = U_S \,\mathrm{diag}(\lambda_1^o, \cdots, \lambda_r^o) U_S^H + \lambda_{r+1}^o (I - U_S U_S^H)$, for which

$$\dot{V}_2(\tau) = -2 \sum_{1 \leq i < j \leq r} \frac{(\lambda_i^o - \lambda_j^o)}{\lambda_i} \left| \boldsymbol{u}_i^H R^o \boldsymbol{u}_j \right|^2 - 2 \sum_{i=1}^{r} \frac{(\lambda_i^o - \lambda_{r+1}^o)}{\lambda_i} \boldsymbol{u}_i^H R^o P_N R^o \boldsymbol{u}_i. \tag{27}$$

Stability of the eigenvalues then follows immediately from (18) and (19). That $\mathcal{D}_{A_2}$ is a domain of attraction is implied by Lemmas 3 and 4 and the fact that $V_2(\phi) \to \infty$ as $\phi \to \infty$. □

According to the ODE method, convergence of Proteus-2 toward the desired parameter set $\mathcal{D}^o$ follows immediately from Theorem 6. Note that in the latter, no specific assumption is made on the eigenvalue structure, even though the derivation of Proteus-2 is based on the assumption that $\lambda_{i,k} \gg \lambda_{i+1,k}$ $(i = 1, \cdots, r)$. This provides a theoretical justification for the robustness of this algorithm to the eigenvalue separation, as experimentally evidenced in [2].

In a finite-precision implementation of Proteus-1 or -2, the eigenvector estimates will invariably deviate from perfect orthonormality due to quantization effects. For instance, due to a finite representation capability, the plane rotations used in the update are only approximately orthogonal. A potentially more serious problem is the accumulation of round-off/truncation errors in the sequential application of these approximate rotations over long periods of operations. Indeed, some subspace trackers require the use of reorthonormalization mechanisms to avoid such error accumulations. Yet, as experimentally demonstrated in [2], the Proteus algorithms show no sign of error buildup; they maintain a level of orthogonality in the eigenvector estimates close to machine precision at all time. The following theorem provides an explanation for this behavior.

[1]Specifically, the conclusions of Theorem 5 remain valid if, among all possible combinations of $N - r$ eigenvalues, the noise subspace eigenvalues are the most clustered ones in a least squares sense.

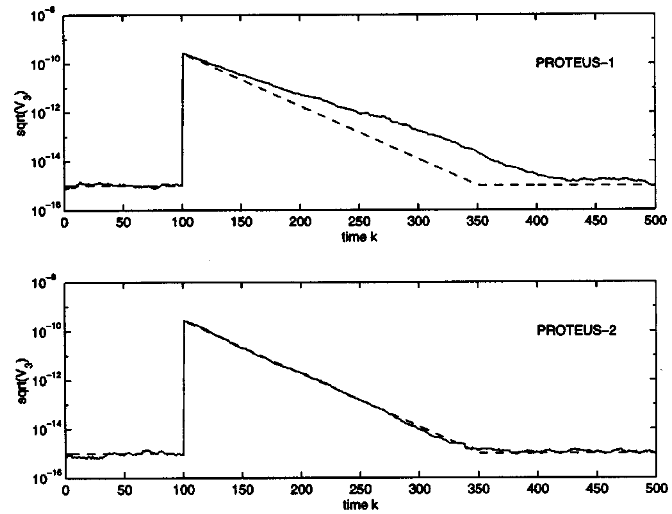

Fig. 1. Orthonormality error $\sqrt{V_3(k)}$ versus time sample k for Proteus 1 and 2 after sudden deviation from orthonormality at $k = 100$. Solid line: average of $\sqrt{V_3(k)}$ over four independent experiments. Dashed line: asymptotic behavior predicted by theory.

Theorem 7: For both Proteus-1 and 2, the manifold $\mathcal{M}$ is an asymptotically stable invariant set of the ODE (17) in the vicinity of the desired solution set $\mathcal{D}^o$ in the parameter space $\mathcal{P}$ [i.e., $\phi(\tau)$ close to $\mathcal{D}^o$ but not restricted to $\mathcal{M}$].

Proof: To investigate stability, consider $V_3(\phi) = 1/2\|U_S^H U_S - I_r\|_F^2$. Note that $V_3(\phi) \geq 0$ with equality only when $U_S^H U_S = I_r$ (i.e., $\phi \in \mathcal{M}$). Using (22), we may obtain

$$\dot{V}_3(\tau) = \mathrm{tr}\left\{ \left(U_S^H U_S - I_r\right) \left(U_S^H R^o U_S \Omega + \Omega U_S^H R^o U_S\right) \cdot \left(U_S^H U_S - I_r\right)\right\} \tag{28}$$

where $\Omega = \mathrm{diag}(l_{1,r+1}, \cdots, l_{r,r+1})$. In the vicinity of $\mathcal{D}^o$, i.e., $\mathrm{dist}(\psi, \mathcal{D}^o) < \delta$ for some small $\delta > 0$, we have $U_S^H R^o U_S = \mathrm{diag}(\lambda_{\pi_1}^o, \cdots, \lambda_{\pi_r}^o) + O(\delta)$. Furthermore, $l_{i,r+1} \approx -1/(\lambda_i^o - \lambda_{r+1}^o)$ for Proteus-1 and $l_{i,r+1} \approx -1/\lambda_i^o$ for Proteus-2. Accordingly, we find that $\dot{V}_3(\tau) \leq -4(1 + O(\delta))V_3$. Thus, assuming that the trajectory $\phi(\tau)$ remains sufficiently close to $\mathcal{D}^o$, $\dot{V}_3(\tau) \leq 0$ with equality only when $\phi \in \mathcal{M}$. □

Thus, both Proteus-1 and -2 are numerically stable in the sense that any small deviation from orthonormality due to finite precision arithmetic is compensated by an internal reorthonormalization mechanism that pushes back the parameter vector estimate ϕ_k toward $\mathcal{M}$, provided that ϕ_k is in the vicinity of the desired parameter set $\mathcal{D}^o$. In practice, this restriction is of no consequence since deviation from orthonormality only occurs after a relatively long period of operation that extends much beyond the initial convergence time.

Fig . 1 illustrates the behavior of $V_3(k) \equiv V_3(\phi_k)$ for Proteus-1 and -2 after a sudden deviation from orthonormality. The simulation scenario is the same as in [2, Fig. 2(b)], but at time $k = 100$, the matrix $U_{S,k}$ is replaced by $U_{S,k}(I_r + \Delta)$, where the entries of Δ are independent Gaussian variables with zero-mean and standard deviation 10^{-10}. It can be seen that after the sudden increase at $k = 100$, $V_3(k)$ exponentionally decays to its original value. Furthermore, the rate of decay is consistent with the above theory, i.e., $V_3(k) \approx V_3(k_o)e^{-4\gamma(k-k_o)}$ for $k \geq k_o$ (which is shown as a dashed line in Fig. 1).

REFERENCES

[1] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. New York: Springer-Verlag, 1990.

[2] B. Champagne and Q. G. Liu, "Plane rotation-based EVD updating schemes for efficient subspace tracking," *IEEE Trans. Signal Processing*, vol. 46, pp. 1886–1900, July 1998.

[3] 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.

[4] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[5] J. La Salle and S. Lefschetz, *Stability by Lyapunov's Direct Method with Applications*. New York: Academic, 1961.

[6] B. Yang, "Asymptotic convergence analysis of the projection approximation subspace tracking algorithms," *Signal Process.*, vol. 50, pp. 123–136, Mar. 1996.

Experimental Performance of Adaptive Beamforming in a Sonar Environment with a Towed Array and Moving Interfering Sources

Alex B. Gershman, Etienne Németh, and Johann F. Böhme

***Abstract*—The performances of adaptive array algorithms are known to suffer from a strong degradation in scenarios with moving interfering sources. In this correspondence, basic adaptive beamforming techniques are compared using shallow sea sonar data recorded in a towed horizontal array environment with moving interfering sources originated from shipping noise. Our experimental results show the relationship between the practical performances of adaptive and conventional beamforming techniques compared in terms of output SINR or a related measure given by the noncompensated postbeamforming interference power. These results demonstrate noticeable performance improvements that can be achieved using several robust algorithms relative to traditional adaptive beamforming schemes.**

***Index Terms*—Robust adaptive beamforming, towed array.**

I. INTRODUCTION

Adaptive array algorithms [14] are known to degrade in scenarios with moving interfering sources [4], [5]. As a rule, such a degradation occurs either due to rapidly moving interfering sources or because of array motion (e.g., in towed arrays, arrays with moving platforms, etc.) and is caused by the fact that the interferers move away from the sharp notches of the adapted pattern and lose their wavefront coherence if their motion is sufficiently fast. Such a degradation may be especially strong for large arrays because of relatively sharp notches of an adapted pattern. Recently, several robust algorithms have been proposed to overcome this problem via artificial widening of adaptive pattern nulls [4], [5]. Simulation results in [4] and [5] have shown drastic performance improvements achieved relative to traditional (nonrobust) adaptive array techniques. However, no experimental performance study has been done in the presence of multiple moving interferences. It is worth noting that the practical performances of adaptive beamformers in a real sonar environment may be subject to much stronger degradation than that resulting from simulations and theoretical study [6], [15]. This stronger degradation may be caused by wavefront fluctuations and distortions in inhomogeneous media, imperfect wavefront coherence, modal propagation effects, diffuse scattering, and reverberation phenomena, as well as receiving antenna defects [6], [17]. Although the real-data performance evaluation of adaptive beamforming techniques in sonar environments appears to be an important today task [9], very few array processing results have been reported for actual sonar hydrophone array systems [2], [6], [10], [15].

In this correspondence, the real-data performances of the popular sample matrix inversion (SMI) [14], loaded SMI (LSMI) [1], and Hung–Turner (HT) [8] algorithms are compared with that of conventional beamformer [14] and recently developed robust adaptive beamforming methods [4], [5]. To study the experimental performances of these techniques, we employ experimental Baltic Sea array data recorded by a towed horizontal array of 16 hydrophones.

Manuscript received November 20, 1998; revised June 2, 1999. This work was supported in part by the Grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada and the German Research Foundation (DFG). The associate editor coordinating the review of this paper and approving it for publication was Prof. S. M. Jesus.

A. B. Gershman is with the Communications Research Laboratory, Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ont., Canada L8S 4K1 (e-mail: gershman@ieee.org).

E. Németh and J. F. Böhme are with Signal Theory Group, Department of Electrical Engineering, Ruhr University, Bochum, Germany.

Publisher Item Identifier S 1053-587X(00)00088-X.

II. ADAPTIVE BEAMFORMING ALGORITHMS

The complex adaptive beamformer output with the weight vector $\boldsymbol{w}$ at time t can be expressed as

$$z(t) = \boldsymbol{w}(t)^H \boldsymbol{y}(t) \tag{1}$$

where

$$\boldsymbol{y}(t) = (y_1(t), \cdots, y_n(t))^T \tag{2}$$

is the $n \times 1$ vector of array observations (hereafter referred to as the beamforming snapshot), whereas $(\cdot)^T$ and $(\cdot)^H$ stand for the transpose and Hermitian transpose, respectively. In this correspondence, it is assumed that the vector (2) contains signal-free observations, i.e., the interference and noise components only. The optimal weight vector maximizing the signal-to-interference-plus-noise ratio (SINR) is given by [14]

$$\boldsymbol{w}_{opt}(t) = \alpha(t)\, \boldsymbol{R}(t)^{-1}\ \boldsymbol{a}_S(t) \tag{3}$$

where $\boldsymbol{a}_S(t)$ is the time-varying (in the general case) $n \times 1$ desired signal direction vector

$$\boldsymbol{R}(t) = \mathrm{E}\{\boldsymbol{y}(t)\boldsymbol{y}^H(t)\} \tag{4}$$

is the $n \times n$ interference-plus-noise covariance matrix, and $\alpha(t)$ is a normalizing constant [14], which does not affect the output SINR

$$\mathrm{SINR}(t) = \frac{\sigma_S^2 |\boldsymbol{w}(t)^H \boldsymbol{a}_S(t)|^2}{\boldsymbol{w}(t)^H \boldsymbol{R}(t) \boldsymbol{w}(t)} \tag{5}$$

of the beamformer (1). Here, σ_S^2 is the signal power in a single sensor.