Adaptive Estimation of Eigensubspace

G. Mathew, Vellenki U. Reddy, *Senior Member, IEEE*, and Soura Dasgupta, *Senior Member, IEEE*

Abstract—In a recent work we recast the problem of estimating the minimum eigenvector (eigenvector corresponding to the minimum eigenvalue) of a symmetric positive definite matrix into a neural network framework. We now extend this work using an inflation technique to estimate all or some of the orthogonal eigenvectors of the given matrix. Based on these results, we form a cost function for the finite data case and derive a Newton-based adaptive algorithm. The inflation technique leads to a highly modular and parallel structure for implementation. The computational requirement of the algorithm is $\mathcal{O}(N^2)$, N being the size of the covariance matrix.

We also present a rigorous convergence analysis of this adaptive algorithm. The algorithm is locally convergent and the undesired stationary points are unstable. Computer simulation results are provided to compare its performance with that of two adaptive subspace estimation methods proposed by Yang and Kaveh and an improved version of one of them, for stationary and nonstationary signal scenarios. The results show that the proposed approach performs identically to one of them and is significantly superior to the remaining two.

I. INTRODUCTION

THE PROBLEM of estimating the parameters of narrow-band signals in additive white noise has been a subject of active research recently. The various methods which can be applied to this problem can be broadly classified into two categories: i) the eigenstructure-based methods which exploit the eigenstructure (eigenvectors and eigenvalues) of the covariance matrix of the underlying signal; and ii) the noneigenstructure-based methods. The eigenstructure-based methods are preferred, since they yield high resolution and asymptotically exact results. Implementation of these methods calls for the estimation of eigenvectors of the covariance matrix. Even though these eigenvectors can be estimated using any of the known block approaches (if we are given the covariance matrix), for real time applications we need methods to adaptively estimate them; this is the objective of this paper.

In real time applications, the data covariance matrix is recursively updated as

$$R(n) = aR(n-1) + E$$

where $R(n)$ denotes the data covariance matrix at nth data instant, a is a parameter in the range $(0, 1]$, and E is a symmetric matrix of rank much less than that of $R(n-1)$.

G. Mathew and V. U. Reddy are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore-560 012, India.

S. Dasgupta is with the Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, IA 52242 USA.

While working with stationary signal scenarios, we use rank-1 update with $a = (n-1)/n$ and $E = (1/n)x(n)x^T(n)$ where $x(n)$ is the data vector at nth instant. On the other hand, in the nonstationary case, rank-1 updating is carried out by choosing a in the range $(0, 1)$ and E as $x(n)x^T(n)$. Another commonly used updating scheme in the nonstationary case is the rank-2 update with $a = 1$ and $E = x(n)x^T(n) - x(n-L)x^T(n-L)$, where L is the window length over which the covariance matrix is computed.

Let R be the true (or asymptotic) covariance matrix of x. Then, the aim of any adaptive subspace estimation scheme is to estimate the eigenstructure of R using $R(n)$ as its estimate. This is especially true in applications such as the estimation of directions-of-arrival or frequencies of stationary narrowband signals. But, for the case of nonstationary signals, one would like to obtain the eigenstructure of $R(n)$ itself.

Development of adaptive techniques to calculate the eigenstructure of $R(n)$ can be classified under the topic "modified eigenvalue problem." In this, the objective is to develop an algorithm which will compute the eigenstructure of $R(n)$ given the prior knowledge of the eigenstructure of $R(n-1)$. A fundamental work in this area is that of Bunch *et al.* [11]. Some of the very recent contributions are those of DeGroat and Roberts [12], Yu [13], Bischof and Shroff [14], and DeGroat [15]. In [12] is developed a parallel and numerically stable algorithm for the rank-1 recursive updating of the eigenstructure. A parallel algorithm for rank-k recursive updating of the eigenstructure was proposed by Yu [13]. Bischof and Shroff [14] reported an approach for updating the noise subspace using a rank-revealing QR-factorization. A noniterative and computationally inexpensive subspace updating algorithm was proposed by [15]. Most recently, based on the rank-revealing URV decomposition, Stewart [16] proposed a subspace updating algorithm.

In this paper, we are interested in estimating the eigenstructure of R. Many researchers used the fact that eigenvectors can be estimated by minimizing a specific cost function subject to certain nonlinear constraints [2], [5], [7], [9] for developing adaptive algorithms. An adaptive approach for estimating the orthogonal eigenvectors corresponding to the signal subspace of the covariance matrix was first developed by Owsley [1]. Thompson [2] exploited the constrained minimization formulation to develop a constrained stochastic gradient algorithm for seeking the eigenvector corresponding to the minimum eigenvalue (which we refer to as minimum eigenvector). Later, Larimore [3] studied the convergence properties of Thompson's [2] approach. Reddy *et al.* [4] restated this constrained minimization problem into an unconstrained framework and developed a least-squares-type recursive algorithm for seeking

the minimum eigenvector. The development of adaptive algorithms for the single eigenvector case was further explored by Durrani and Sharman [5], Vaccaro [6], and Fuhrmann and Liu [7]. Sharman [8] developed an adaptive algorithm, based on the QR-recursions, to estimate the complete eigenstructure. Recently, Yang and Kaveh [9] proposed an adaptive approach for estimating the complete noise subspace or the signal subspace.

We present an adaptive approach which combines a Newton-type algorithm and an inflation method for estimating the eigensubspace of the covariance matrix. The resulting algorithm is highly modular and well suited for parallel implementation. We develop this in an unconstrained minimization framework. The basic principle of this approach can be used to compute the complete eigenstructure of any symmetric nonindefinite matrix. Our objective here is the same as that of Yang and Kaveh [9], but our approach is different from theirs, both in the nature of the algorithm and the inflation technique. We present simulation results comparing the performance of our approach with the maximization gradient search approach and the inflation-based approach of [9]. Since their inflation-based algorithm uses instantaneous gradient while we use exact gradient, we modified their algorithm by incorporating exact gradient, and compared our algorithm with the modified version as well.

In Section II, we briefly review the unconstrained formulation reported in [10] for a single eigenvector case and extend it to the case of orthogonal multiple eigenvectors. The finite data case is considered in Section III. Convergence analysis of this algorithm is presented in Section IV. Simulation results are discussed in Section V. Computational requirements of the algorithm are given in Section VI and Section VII concludes the paper.

II. FORMULATION FOR SEEKING THE EIGENSUBSPACE

In this section, our aim is to develop a method to obtain the first D orthogonal eigenvectors and the corresponding eigenvalues of a positive definite matrix R of size $N \times N$.

Suppose

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{N-1} \leq \lambda_N \qquad (2.1)$$

represent the eigenvalues of R in the ascending order of magnitude and q_i, $i = 1, \cdots, N$, the corresponding orthonormal eigenvectors. Then, we can write

$$R = \sum_{i=1}^{N} \lambda_i q_i q_i^T = Q \Lambda Q^T \qquad (2.2)$$

where

$$Q = [q_1, q_2, \cdots, q_N] \quad \text{and} \quad \Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N). \qquad (2.3)$$

If R is the true covariance matrix of a time series consisting of P ($2P < N$) real sinusoids in additive white noise of variance σ^2, then its minimum eigenvalue λ_{min} is equal to σ^2 with multiplicity $N - 2P$ (i.e., $\lambda_1 = \lambda_2 = \cdots = \lambda_{N-2P} = \lambda_{min} = \sigma^2$). The minimum eigenvalues and the corresponding

eigenvectors are also known as noise eigenvalues and noise eigenvectors, respectively. Hence, we call the span of $q_1, \cdots, q_{N-2P}$ the *noise subspace* and its orthogonal complement the *signal subspace*. Estimation of a basis for the noise or signal subspace is a primary requirement in many of the "super-resolution" spectral estimation techniques.

It is well known that a minimum eigenvector of R is the solution of the following constrained minimization problem

$$\min_{w} w^T R w \text{ subject to } w^T w = 1 \qquad (2.4)$$

where $w = [w_1, w_2, \cdots, w_N]^T$ is an $N \times 1$ vector. It has been shown in [10] that this problem can be recast into an unconstrained form as

$$\min_{w} \{ J(w, \mu) = w^T R w + \mu (w^T w - 1)^2 \} \qquad (2.5)$$

where μ is a positive parameter. This cost function has a special structure that "a minimizer of $J(w, \mu)$ is a minimum eigenvector of R for any $\mu > \lambda_{min}/2$." We state the following main results (see [10] for proof) pertaining to this cost function.

Results:

i) w^* is a global minimizer of J if and only if w^* is an eigenvector of R corresponding to the eigenvalue λ_{min} with $\|w^*\|^2 = \beta^2 = 1 - \lambda_{min}/2\mu$.

ii) For a given μ, every local minimizer of J is also a global minimizer.

We now present an inflation approach to extend the above results to obtain the first D orthogonal eigenvectors.

Let w_i^* be the eigenvector (with norm $\beta_i = \sqrt{1 - \lambda_i/2\mu}$) of R corresponding to the eigenvalue λ_i for $i = 1, \cdots, k-1$, (with $2 \leq k \leq D$) and assume that $w_1^*, \cdots, w_{k-1}^*$ are orthogonal. Our objective is to obtain the next orthogonal eigenvector w_k^*, assuming that $w_i^*, i = 1, \cdots, k-1$, are available.

Define a function J_k as

$$J_k(w_k, \mu, \alpha) = w_k^T R w_k + \mu (w_k^T w_k - 1)^2$$
$$+ \alpha \sum_{i=1}^{k-1} (w_k^T w_i^*)^2$$
$$= w_k^T R_k w_k + \mu (w_k^T w_k - 1)^2 \qquad (2.6)$$

where $w_k \in \mathbb{R}^N, \alpha > 0$ and

$$R_k = R + \alpha \sum_{i=1}^{k-1} w_i^* w_i^{*T}$$
$$= R_{k-1} + \alpha w_{k-1}^* w_{k-1}^{*T} \qquad k = 2, \cdots, D \qquad (2.7)$$

with $R_1 = R$. Equation (2.7) defines the inflation step and its implication is discussed below. Defining $q_i = w_i^*/\beta_i$, $i = 1, \cdots, k-1$, we get the spectral representation of R_k as (cf. (2.2), (2.7))

$$R_k = \sum_{i=1}^{N} \left[\lambda_i + \alpha \sum_{j=1}^{k-1} \beta_j^2 \delta_{ij} \right] q_i q_i^T. \qquad (2.8)$$

Now, if α is chosen such that

$$\lambda_k < \lambda_i + \alpha\beta_i^2 \qquad i = 1, \cdots, k-1 \tag{2.9}$$

then the minimum eigenvalue of R_k is λ_k with q_k as the corresponding eigenvector. Thus, if w_k^* is a minimizer of J_k, then from the results (i) and (ii), it follows that w_k^* is a minimum eigenvector of R_k. But, observe from (2.8) that this eigenvector corresponds to the kth eigenvector of R. That is,

$$Rw_k^* = \lambda_k w_k^* \quad \text{with} \quad \|w_k^*\|^2 = 1 - \frac{\lambda_k}{2\mu}. \tag{2.10}$$

Further, the strict inequality in (2.9) guarantees that

$$w_k^{*T} w_i^* = 0 \qquad \forall i = 1, \cdots, k-1. \tag{2.11}$$

Thus, by constructing D cost functions $J_k, k = 1, \cdots, D$, as in (2.6) and finding their minimizers, we get the D orthogonal eigenvectors of R. Norm of these eigenvectors depends on the corresponding eigenvalues and μ. If we wish to obtain all the orthogonal eigenvectors (i.e., $D = N$), the parameters μ and α must satisfy the following conditions

$$\mu > \frac{\lambda_N}{2} \quad \text{and} \quad \alpha > 2\mu. \tag{2.12}$$

The condition on μ follows from the result (i) and that on α can be obtained from (2.9) with $k = N$. Since the eigenvalues of R are not known *a priori*, we suggest the following practical lower bounds for μ and α:

$$\mu > \frac{\text{Trace}(R)}{2} \quad \text{and} \quad \alpha > \text{Trace}(R). \tag{2.13}$$

III. ADAPTIVE APPROACH TO EIGENSUBSPACE ESTIMATION

Consider $y(n) = a^T x(n)$ where $a = [a_1, \cdots, a_N]^T$ is the coefficient vector and $x(n) = [x(n), \cdots, x(n-N+1)]^T$ is the most recent data vector. Following the unconstrained minimization (2.5), we define the criterion function for the data case as

$$J(a, \mu, n) = \frac{1}{n} \sum_{i=1}^{n} y^2(i) + \mu(a^T a - 1)^2$$
$$= a^T R(n) a + \mu(a^T a - 1)^2 \tag{3.1}$$

where n is the number of samples and $R(n)$ is the data covariance matrix defined as

$$R(n) = \frac{1}{n} \sum_{i=1}^{n} x(i) x^T(i). \tag{3.2}$$

In the following subsection, we derive an adaptive algorithm for estimating the minimum eigenvector of the asymptotic covariance matrix of $x(n)$, R, by minimizing J and in the next section we show that it converges to a minimum eigenvector of R.

A. Single Eigenvector Case

Let $a(n)$ be the coefficient vector at the nth instant. Then, the Newton-based adaptive algorithm for minimizing J is of the form

$$a(n) = a(n-1) - \{H^{-1}g\}|_{a=a(n-1)}$$
$$= a(n-1) - [H(n-1)]^{-1} g(n-1). \tag{3.3}$$

Here, $H(n-1)$ and $g(n-1)$ are the Hessian matrix and gradient vector, respectively, of J with respect to a, evaluated at $a = a(n-1)$, and are given by

$$g(n-1) = 2R(n)a(n-1)$$
$$+ 4\mu[a^T(n-1)a(n-1) - 1]a(n-1) \tag{3.4}$$

$$H(n-1) = 2R(n) + 8\mu a(n-1)a^T(n-1)$$
$$+ 4\mu[a^T(n-1)a(n-1) - 1]I_N. \tag{3.5}$$

Now, to develop the Newton-based algorithm (cf. (3.3)), $H(n)$ must be positive definite. Further, to make it computationally efficient, we should be able to obtain a recursion for the inverse of $H(n)$ directly. These two requirements are met if we approximate the Hessian by dropping the last term in (3.5). That is,

$$H(n-1) \approx \tilde{H}(n-1) = 2R(n) + 8\mu a(n-1)a^T(n-1). \tag{3.6}$$

Obtaining the expression for the inverse of $\tilde{H}(n-1)$ (using the matrix inversion lemma) and substituting the result along with (3.4) in (3.3), we obtain the Newton-type adaptive algorithm (after some manipulations) as

$$a(n) = l(n-1)R^{-1}(n)a(n-1) \tag{3.7}$$

$$l(n-1) = \frac{1 + a^T(n-1)a(n-1)}{\dfrac{1}{2\mu} + 2a^T(n-1)R^{-1}(n)a(n-1)} \tag{3.8}$$

$$R^{-1}(n) = \frac{n}{n-1}\left[R^{-1}(n-1) - \frac{R^{-1}(n-1)x(n)x^T(n)R^{-1}(n-1)}{n-1 + x^T(n)R^{-1}(n-1)x(n)} \right]$$
$$n \geq 2. \tag{3.9}$$

B. Extension to the Case of Multiple Eigenvectors

Extension of the above adaptive algorithm to the case of multiple eigenvectors directly follows from the principle described in Section II. The method can be explained using the block schematic given in Fig. 1. Consider the kth unit. Here, $a_k(n) \in \mathbb{R}^N$ is the estimate of the kth eigenvector at nth data instant. The adaptation criterion for kth unit is the minimization of the cost function J_k defined as

$$J_k(a_k, \mu, \alpha, n) = a_k^T R_k(n) a_k + \mu(a_k^T a_k - 1)^2$$
$$k = 1, \cdots, D \tag{3.10}$$

where

$$R_k(n) = R_{k-1}(n) + \alpha a_{k-1} a_{k-1}^T \qquad k = 2, \cdots, D \tag{3.11}$$
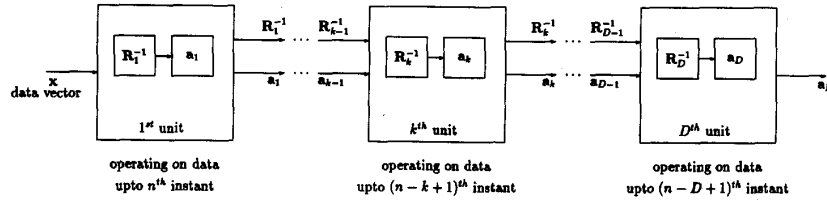
Fig. 1. Modular implementation of the proposed adaptive subspace estimation method.

with $R_1(n) = R(n)$. Then, the Newton-type algorithm for recursively estimating the D eigenvectors follows from (3.7) to (3.9) and is given below:

$$a_k(n) = l_k(n-1)R_k^{-1}(n)a_k(n-1)$$
$$k = 1, \cdots, D \qquad (3.12)$$

$$l_k(n-1) = \frac{1 + a_k^T(n-1)a_k(n-1)}{\dfrac{1}{2\mu} + 2a_k^T(n-1)R_k^{-1}(n)a_k(n-1)}$$
$$k = 1, \cdots, D \qquad (3.13)$$

$$R_k^{-1}(n) = R_{k-1}^{-1}(n)$$
$$- \frac{R_{k-1}^{-1}(n)a_{k-1}(n)a_{k-1}^T(n)R_{k-1}^{-1}(n)}{\dfrac{1}{\alpha} + a_{k-1}^T(n)R_{k-1}^{-1}(n)a_{k-1}(n)}$$
$$k = 2, \cdots, D. \qquad (3.14)$$

The recursion (3.14) is obtained by applying the matrix inversion lemma to (3.11). Note that $R_1^{-1}(n) = R^{-1}(n)$ and it is updated as in (3.9).

The above algorithm can be implemented using a pipeline architecture as described below. During one sampling interval, kth unit goes through the following steps (see Fig. 1):

1) The current weight vector a_k and the matrix R_k^{-1} are passed on to the $(k + 1)$th unit.
2) The weight vector a_{k-1} and the matrix R_{k-1}^{-1} are accepted from the $(k - 1)$th unit.
3) R_k^{-1}, l_k, and a_k are updated.

As a result of steps 1 to 3, the data $x(n)$ which enters the first unit reaches the kth unit after a delay of $(k - 1)$ sampling intervals. Consequently, there will be a similar delay in getting the final eigenvector estimates from each of these units. Because of the nature of the inflation technique used (cf. (3.11)), the computations required for updating the eigenvector estimates are identical for all the units, thus making the adaptive algorithm both modular and parallel in nature.

We may point out here that we did not assume any particular signal scenario during the development of either the asymptotic formulation in Section II or the adaptive algorithm in Section III. Hence, the adaptive algorithm developed above (cf. (3.12)–(3.14)) can be used to estimate the eigenvectors of the covariance matrix of any signal scenario and is not limited to the case of sinusoids in noise.

IV. CONVERGENCE ANALYSIS

In this section, we provide a proof for the local convergence of the recursive algorithm given in (3.12) to (3.14). That is, assuming that the algorithm is *initialized appropriately*, we show that it *converges asymptotically* to the desired set of orthogonal eigenvectors of the asymptotic covariance matrix R. We do this in two steps. First, we prove the convergence of $\{a_1(n)\}$ to a minimum eigenvector of R and then extend this to the case of multiple eigenvectors.

For the case of a single eigenvector, we shall refer to the algorithm given in (3.7) to (3.9). The convergence analysis is based on the following two assumptions:

1) *Ergodicity assumption:* The underlying process $\{x(n)\}$ is ergodic. That is, $\exists$ R such that $R^{-1}(n)$ of (3.9) obeys

$$\lim_{n \to \infty} R^{-1}(n) = R^{-1}. \qquad (4.1)$$

2) *Richness assumption:* The data is rich, i.e., it satisfies the persistent excitation condition: $\exists$ finite $\alpha_1 > 0$ and $\alpha_2 > 0$ such that for all $n \geq 0$

$$\alpha_1 I_N \leq R^{-1}(n) \leq \alpha_2 I_N. \qquad (4.2)$$

In the analysis of convergence, it is important to specify a precise quantification of the distance between $a(n)$ and the subspace of minimum eigenvectors of R. Let us define

$$b(n) = Q^T a(n)$$

and

$$f(n) = \frac{\|[b_{M+1}(n), \cdots, b_N(n)]\|_\infty}{\|[b_1(n), \cdots, b_M(n)]\|_\infty} \qquad (4.3)$$

where M is assumed to be the multiplicity of the minimum eigenvalue, i.e.

$$\lambda_1 = \lambda_2 = \cdots = \lambda_M < \lambda_{M+1} \leq \lambda_{M+2} \cdots \leq \lambda_N \qquad (4.4)$$

and Q is as defined in (2.3). Clearly, $f(n) = 0$ implies that $a(n)$ is in the desired subspace.

Premultiplying (3.7) with Q^T, we obtain

$$b(n) = l(n-1)[\Lambda^{-1} + \Psi(n)]b(n-1) \qquad (4.5)$$

where

$$\Psi(n) = Q^T[R^{-1}(n) - R^{-1}]Q. \qquad (4.6)$$

Now, consider the following Lemma.

Lemma 1: Under (4.1) and (4.2), the quantities in (4.5) have the following properties:

i) There exists finite positive η_1 and η_2 such that

$$2\mu\eta_2 \leq l(n) \leq 2\mu\eta_1 \qquad \forall n. \tag{4.7}$$

ii) $\boldsymbol{\Psi}(n)$ in (4.6) obeys

$$\lim_{n\to\infty} \boldsymbol{\Psi}(n) = 0. \tag{4.8}$$

iii) There exist n_0 and $\delta_4, \delta_5, \delta_6$ (all are positive scalars) such that $\forall n \geq n_0$

$$(a) \quad \frac{\dfrac{1}{\lambda_{M+1}} + \psi_{M+1,M+1}(n)}{\dfrac{1}{\lambda_M} + \psi_{MM}(n)} < 1 - \delta_4 \tag{4.9}$$

$$(b) \quad \delta_5 < l(n-1)\left[\frac{1}{\lambda_N} + \psi_{NN}(n)\right]$$
$$\leq \cdots \leq l(n-1)\left[\frac{1}{\lambda_{M+1}} + \psi_{M+1,M+1}(n)\right] \tag{4.10a}$$

$$l(n-1)\left[\frac{1}{\lambda_i} + \psi_{ii}(n)\right] < \delta_6$$
$$\forall i \in \{1, \cdots, N\} \tag{4.10b}$$

$$\left[\frac{1}{\lambda_{M+1}} + \psi_{M+1,M+1}(n)\right] < \left[\frac{1}{\lambda_i} + \psi_{ii}(n)\right]$$
$$\forall i \leq M \tag{4.10c}$$

$$\lim_{n\to\infty} |\psi_{ii}(n) - \psi_{jj}(n)| = 0$$
$$\forall i, j \in \{1, \cdots, M\} \tag{4.10d}$$

$$(c) \quad |\psi_{ij}(n)| < \frac{\delta_5\delta_4}{8N} \qquad \forall i \neq j \tag{4.11}$$

where $\psi_{ij}(n)$ denotes the (i,j)th element of $\boldsymbol{\Psi}(n)$.

Proof:

i) From (3.8), we get

$$l(n-1) = 2\mu\frac{1 + \|\boldsymbol{a}(n-1)\|^2}{1 + 4\mu\boldsymbol{a}^T(n-1)\boldsymbol{R}^{-1}(n)\boldsymbol{a}(n-1)}. \tag{4.12}$$

Substituting (4.2) in (4.12), we get

$$\gamma_1(n-1) \leq l(n-1) \leq \gamma_2(n-1) \tag{4.13}$$

where

$$\gamma_1(n-1) = 2\mu\frac{1 + \|\boldsymbol{a}(n-1)\|^2}{1 + 4\mu\alpha_2\|\boldsymbol{a}(n-1)\|^2}$$

$$\gamma_2(n-1) = 2\mu\frac{1 + \|\boldsymbol{a}(n-1)\|^2}{1 + 4\mu\alpha_1\|\boldsymbol{a}(n-1)\|^2}.$$

Since μ, α_1 and α_2 are finite positive quantities, there exists finite $\eta_1 > 0$ and $\eta_2 > 0$ such that

$$\frac{1 + \|\boldsymbol{a}(n-1)\|^2}{1 + 4\mu\alpha_2\|\boldsymbol{a}(n-1)\|^2} \geq \eta_2$$

and

$$\frac{1 + \|\boldsymbol{a}(n-1)\|^2}{1 + 4\mu\alpha_1\|\boldsymbol{a}(n-1)\}^2} \leq \eta_1. \tag{4.14}$$

Combining (4.13) and (4.14), we get (i).

ii) This follows from (4.6) and (4.1).

iii) All the results under this item can be established using (4.4), (4.1), (4.7), and (4.8).

Now, observe that the underlying process governing the behavior of $f(n)$ in (4.3) is such that (see (4.1), (4.2), (4.7)) for every n_0, there exists $\epsilon_6 > 0$ such that

$$|f(n_0)| < 1 \qquad \text{whenever} \quad |f(0)| < \epsilon_6. \tag{4.15}$$

Then, the following theorem (proved in Appendix) establishes the local convergence of $\boldsymbol{a}(n)$ to a minimum eigenvector of $\boldsymbol{R}$:

Theorem 1: There exists $\epsilon_6 > 0$ such that with $|f(0)| < \epsilon_6$

$$\lim_{n\to\infty} f(n) = 0. \tag{4.16}$$

That is, "if $\boldsymbol{a}(n)$ is initialized sufficiently close to the space of minimum eigenvectors of $\boldsymbol{R}$, then $\boldsymbol{a}(n)$ will converge asymptotically to a minimum eigenvector."

Having shown the local convergence of $\{\boldsymbol{a}(n)\}$ (or $\{\boldsymbol{a}_1(n)\}$), we want to examine how the algorithm behaves in the neighborhood of any of the undesired eigenvectors (i.e., eigenvectors other than the minimum eigenvectors).

Consider the ideal case of $\boldsymbol{R}(n) = \boldsymbol{R}$ for all n. Then, (3.7) and (3.8) can be rewritten as

$$\boldsymbol{b}(n) = l(n-1)\boldsymbol{\Lambda}^{-1}\boldsymbol{b}(n-1) \tag{4.17}$$

$$l(n) = \frac{1 + \boldsymbol{b}^T(n)\boldsymbol{b}(n)}{\dfrac{1}{2\mu} + 2\boldsymbol{b}^T(n)\boldsymbol{\Lambda}^{-1}\boldsymbol{b}(n)} \tag{4.18}$$

where $\boldsymbol{b}(n) = \boldsymbol{Q}^T\boldsymbol{a}(n)$ or $\boldsymbol{a}(n) = \Sigma_{i=1}^{N}b_i(n)\boldsymbol{q}_i$. Now, (4.17) implies that

$$b_i(n) = \frac{l(n-1)}{\lambda_i}b_i(n-1) \qquad i = 1, \cdots, N. \tag{4.19}$$

Suppose $\boldsymbol{a}(n)$ is close to the kth eigenvector for $k \in \{M + 1, \cdots, N\}$. Then, $|b_k(n)| \gg |b_j(n)|$ for all $j \neq k$. Suppose also that at least one of $b_1(n-1), \cdots, b_M(n-1)$ is nonzero but possibly very small. Then, since $l(n-1)/\lambda_j > l(n-1)/\lambda_k$ for $j \in \{1, \cdots, M\}$, $b_j(n)$ will increase exponentially at a rate faster than that experienced by $b_i(n)$ for $i > M$. Thus, the algorithm forces $\boldsymbol{a}(n)$ to move towards a minimum eigenvector. But, if $\boldsymbol{a}(n)$ is exactly on one of the undesired eigenvectors, say the $(M + 1)$st, then $b_i(n) = 0$ for $i \neq M + 1$, and it follows from (4.19) that it will remain at that eigenvector, implying that the global convergence of the algorithm cannot be guaranteed. In practice, however, $\boldsymbol{R}(n) \neq \boldsymbol{R}$ and hence, the eigenvectors and eigenvalues of $\boldsymbol{R}(n)$ keep changing from instant to instant since $\boldsymbol{R}(n)$ itself is changing. Hence, even if one starts exactly on one of the nonminimum eigenvectors of $\boldsymbol{R}$, the varying nature of $\boldsymbol{R}(n)$ will cause $\boldsymbol{a}(n)$ to move in a random fashion. However, as $\boldsymbol{R}(n)$ converges to $\boldsymbol{R}$, the vector $\boldsymbol{a}(n)$ will move towards the minimum eigenvector. This is because, (4.5), (4.8) and $\lambda_i \geq \lambda_{\min} \forall i$ imply that $\boldsymbol{a}(n)$ will be richer in the minimum eigenvector compared to $\boldsymbol{a}(n-1)$ for sufficiently large n. Thus, all the nonminimum eigenvectors of $\boldsymbol{R}$ are unstable, and hence, the algorithm will eventually converge to a minimum eigenvector of $\boldsymbol{R}$.

Thus, combining the local convergence theorem (which shows that "minimum eigenvectors of R are stable stationary points") with the above arguments (which prove the fact that "nonminimum eigenvectors of R are unstable stationary points"), we conclude that $\{a_1(n)\}$ converges with probability one to a minimum eigenvector of R.

Combining this result with (3.11) and (4.1), we get

$$\lim_{n \to \infty} R_2^{-1}(n) = R_2^{-1} \qquad (4.20)$$

where R_2 is as defined in (2.7). Further, from (3.7), (3.14), (4.2), and (4.7), and noting that $a_1(n) = a(n)$ and $R_1(n) = R(n)$, we obtain

$$\alpha_3 I_N \le R_2^{-1}(n) \le \alpha_4 I_N \qquad \forall n \qquad (4.21)$$

where $\alpha_3 > 0$ and $\alpha_4 > 0$. As a result, $l_2(n)$ (cf. (3.13)) is bounded. Then, it follows from the above-stated theorem that $\{a_2(n)\}$ converges (locally) asymptotically to the minimum eigenvector of R_2 which is nothing but the second eigenvector of R (see Section II).

Using similar arguments, one can easily see that $\{a_k(n)\}$ converges to the kth eigenvector of R, $k = 2, \cdots, D$.

V. SIMULATION RESULTS

In this section, we present some computer simulation results. Performance of the proposed approach was compared with the methods proposed by Yang and Kaveh [9], viz. the inflation based approach and its modified version, and the maximization gradient search approach. The performance measures used are the projection error measure and the orthogonality measure.

A. Adaptive Subspace Estimation Methods of [9]

Since one of the main features of the proposed approach is the inflation method (cf. (3.11)), we have chosen the inflation-based noise subspace estimation method of [9] as one of the methods for performance comparison purpose. The updating steps of their algorithm for the kth eigenvector are given below:

$$a_k'(n) = \overline{a}_k'(n-1) - 2\eta_A x_k(n) y_k(n)$$
$$k = 1, \cdots, D \qquad (5.1)$$

$$\overline{a}_k'(n) = \frac{a_k'(n)}{\|a_k'(n)\|} \qquad k = 1, \cdots, D \qquad (5.2)$$

$$y_k(n) = \overline{a}_k'^T(n-1) x_k(n) \qquad k = 1, \cdots, D \qquad (5.3)$$

$$x_k(n) = x_{k-1}(n) + \overline{a}_{k-1}'(n)(\overline{a}_{k-1}'^T(n) x_{k-1}(n))$$
$$k = 2, \cdots, D \qquad (5.4)$$

with $x_1(n) = x(n)$. Here, η_A is the step-size parameter. Equation (5.4) is the inflation step. If we assume a deterministic framework (as in Section III), the above algorithm can be considered as a constrained gradient search to minimize the least-squares criterion

$$\frac{a_k'^T R_k'(n) a_k'}{2} \qquad (5.5)$$

with the true gradient (evaluated at $\overline{a}_k'(n-1)$, i.e., $R_k'(n)\overline{a}_k'(n-1)$) replaced by its instantaneous estimate

$x_k(n) y_k(n)$. Here, $R_k'(n) = (1/n)\Sigma_{i=1}^n x_k(i) x_k^T(i)$ with $R_1'(n) = R(n)$. Since the Newton algorithm derived in Section III uses exact gradient, we modified the above algorithm correspondingly. The resulting algorithm (for $k = 1, \cdots, D$) is as follows:

$$a_k'(n) = \overline{a}_k'(n-1) - 2\eta_E R_k'(n) \overline{a}_k'(n-1) \qquad (5.6)$$

$$\overline{a}_k'(n) = \frac{a_k'(n)}{\|a_k'(n)\|} \qquad (5.7)$$

$$R_k'(n) = \frac{1}{n} \sum_{i=1}^n x_k(i) x_k^T(i)$$
$$= \frac{n-1}{n} R_k'(n-1) + \frac{1}{n} x_k(n) x_k^T(n) \qquad (5.8)$$

with $x_k(n)$ as defined in (5.4) and η_E is the step-size parameter.

Another method that we used in performance comparison is the maximization gradient search method of [9] combined with the inflation approach proposed in Section III–B (cf. (3.11)). The resulting algorithm is given below:

$$a_k'(n) = \overline{a}_k'(n-1) + 2\eta_M \breve{R}_k^{-1}(n) \overline{a}_k'(n-1)$$
$$k = 1, \cdots, D \qquad (5.9)$$

$$\overline{a}_k'(n) = \frac{a_k'(n)}{\|a_k'(n)\|} \qquad k = 1, \cdots, D \qquad (5.10)$$

$$\breve{R}_k^{-1}(n) = \breve{R}_{k-1}^{-1}(n)$$
$$- \frac{\breve{R}_{k-1}^{-1}(n) \overline{a}_{k-1}'(n) \overline{a}_{k-1}'^T(n) \breve{R}_{k-1}^{-1}(n)}{\frac{1}{\alpha} + \overline{a}_{k-1}'^T(n) \breve{R}_{k-1}^{-1}(n) \overline{a}_{k-1}'(n)}$$
$$k = 2, \cdots, D \qquad (5.11)$$

with $\breve{R}_1^{-1}(n) = R^{-1}(n)$, which is updated as in (3.9). Again, η_M is the step-size parameter.

B. Performance Measures

Our objective is to estimate the first D orthogonal eigenvectors, q_i, $i = 1, \cdots, D$, of R. Let S_D be the span of these vectors. The measures we used for evaluating the quality of the estimated eigensubspace are as follows. To see the closeness of the estimated eigenvectors to the true subspace, we use the projection error measure, $E(n)$, defined as

$$E(n) = \frac{1}{D} \sum_{i=1}^D \left\| \overline{a}_i(n) - \sum_{j=1}^D q_j q_j^T \overline{a}_i(n) \right\| \qquad (5.12)$$

where $\overline{a}_i(n) = a_i(n)/\|a_i(n)\|$. That is, $E(n)$ is the sum of the norms of the projection error vectors obtained by projecting each $\overline{a}_i(n)$ onto S_D. Thus, the smaller the value of $E(n)$, the closer the estimated subspace is to S_D, and vice-versa. Vectors q_i, $i = 1, \cdots, D$, are obtained using the eigenvalue decomposition routine in MATLAB.

Note that $E(n)$ does not reveal the extent of orthogonality among the estimated eigenvectors. Since the proposed approach results in orthogonal eigenvectors because of the inflation approach, we define an orthogonality measure,

$Orth_{\max}(n)$, as

$$Orth_{\max}(n) = \max_{i,j \in \{1, \cdots, D\}} \{|\overline{a}_i^T(n)\overline{a}_j(n)|\}_{i \neq j}. \quad (5.13)$$

Observe that $Orth_{\max}(n)$ is the magnitude cosine of the angle between those two vectors which have the smallest angle between them. Thus, the larger the value of $Orth_{\max}(n)$, the less orthogonal the estimated eigenvectors are, and vice-versa.

For the case of Yang and Kaveh's [9] algorithms, $\overline{a}_i(n)$ is replaced by $\overline{a}_i'(n)$ in (5.12) and (5.13).

C. Performance Evaluation

To study the convergence and tracking performance of the above algorithms, we used both stationary and nonstationary signal scenarios.

Stationary Signal Scenario: The data $x(n)$ was generated as

$$x(n) = A\sin(0.4\pi n) + A\sin(0.48\pi n + \theta) + v(n)$$
$$n = 1, 2, \cdots. \quad (5.14)$$

The initial phase θ (uniformly distributed in $[-\pi, \pi]$) and $v(n)$, a zero-mean white noise of unit variance, were varied from trial to trial of the Monte Carlo simulations. The amplitude of the sinusoids, A, was chosen to give the desired signal to noise ratio (SNR), defined as $10\log(A^2/2)$. The size of the covariance matrix N was fixed at 6 and D at 2.

The four algorithms were initialized as below: *Newton type algorithm (cf. (3.12)–(3.14)) (denoted by A_N):*

$$R^{-1}(0) = 100I_N, \quad a_k(0) = i_k \quad \text{for } k = 1, \cdots, D.$$

Approximate gradient algorithm (cf. (5.1)–(5.4)) (denoted by A_A):

$$a_k'(0) = i_k \quad \text{for } k = 1, \cdots, D.$$

*Exact gradient algorithm (cf. (5.6)–(5.8)) (denoted by A_E):*

$$a_k'(0) = i_k, \qquad R_k'(0) = 0 \qquad \text{for } k = 1, \cdots, D.$$

*Maximization gradient algorithm (cf. (5.9)–(5.11)) (denoted by A_M):*

$$R^{-1}(0) = 100I_N, \qquad a_k'(0) = i_k \qquad \text{for } k = 1, \cdots, D.$$

Here, i_j represents the jth column of the $N \times N$ identity matrix, I_N. Since the updating of the inverse of the data covariance matrix (cf. (3.9)) can be started from $n = 2$ only, $R^{-1}(1)$ was also set to $100I_N$. 100 Monte Carlo runs were performed in each case. The projection error measure for A_N, A_A, A_E, and A_M was evaluated at each iteration (data sample) and averaged over 100 trials. The results are plotted in Fig. 2 for SNR values of 0 dB and 10 dB.

Observe that the performance of A_N is far superior to that of A_A for both the SNR's. This is, however, expected from the nature of the two algorithms. The plots corresponding to A_N and A_E show that the initial convergence of A_N is better, and as the data length increases, both perform almost alike at high SNR while the A_N performs better at low SNR. One obvious reason for the superior performance of A_N is
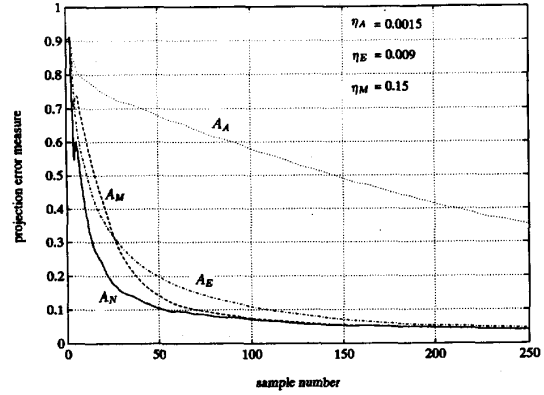
(a)
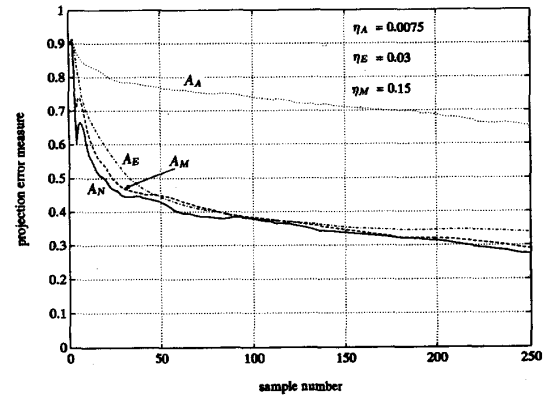
(b)

Fig. 2. Convergence performance of the four algorithms (A_N, A_A, A_E, and A_M) with the subspace of R as the reference (stationary signal scenario): (a) SNR = 10 dB; (b) SNR = 0 dB.

that it is a Newton-type algorithm, whereas the A_E and A_A are gradient-only-based algorithms. Further, the type of the inflation technique used in each case has an influence on the performance. In A_E and A_A, the inflation is done at the data level, while in A_N it is done at the covariance matrix level. The data level inflation introduces extra terms in the expressions for the covariance matrices of higher units (*i.e.*, $R_i'(k), i = 2, \cdots, D$) compared to matrix level inflation. For example, the covariance matrix corresponding to the kth unit in A_E is given by (cf. (5.8))

$$R_k'(n) = R_{k-1}'(n) + R_{k-1}'(n)\overline{a}_{k-1}'(n)\overline{a}_{k-1}'^T(n)$$
$$+ \overline{a}_{k-1}'(n)\overline{a}_{k-1}'^T(n)R_{k-1}'(n)$$
$$+ (\overline{a}_{k-1}'^T(n)R_{k-1}'(n)\overline{a}_{k-1}'(n))\overline{a}_{k-1}'(n)\overline{a}_{k-1}'^T(n)$$
$$= R(n) + \sum_{i=1}^{k-1}[(\overline{a}_i'^T(n)R_i'(n)\overline{a}_i'(n))\overline{a}_i'(n)\overline{a}_i'^T(n)$$
$$+ R_i'(n)\overline{a}_i'(n)\overline{a}_i'^T(n) + \overline{a}_i'(n)\overline{a}_i'^T(n)R_i'(n)].$$
$$(5.15)$$

On the other hand, the corresponding expression in the case
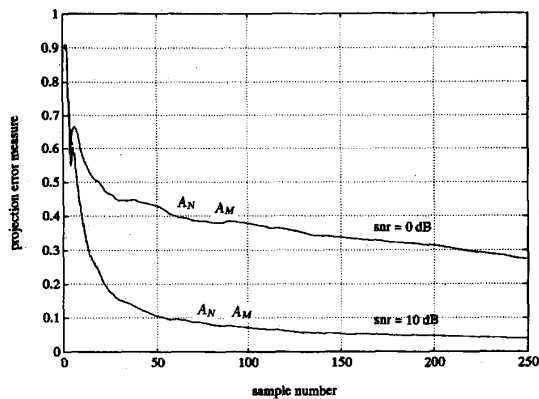
Fig. 3. Comparison of the convergence performance of A_N with A_M for $\eta_M = 20$ (stationary signal scenario).

Fig. 4. Orthogonality measure for the proposed approach (A_N) (stationary signal scenario).

of A_N is given by

$$R_k(n) = R(n) + \alpha \sum_{i=1}^{k-1} a_i(n)a_i^T(n). \qquad (5.16)$$

Note the presence of additional terms on the R.H.S of (5.15) as compared to (5.16). Thus, if $R(n)$ has not converged close enough to R, the additional terms (which contain $R(n)$ implicitly) will cause an increased estimation error in $R'_k(n)$ compared to that in $R_k(n)$. So, the lower the SNR, higher will be the performance degradation of A_E compared to A_N. This phenomenon together with the fact that A_N is a Newton-type algorithm explains its superior performance.

Comparing the performance of A_N and A_M, we observe that A_N is superior. However, extensive simulation studies showed that the superiority of A_N over A_M comes down as the value of η_M goes up. A typical set of results demonstrating this behavior is shown in Fig. 3. Note that for $\eta_M = 20$, the performance of A_N and A_M are identical. The reason for such a behavior can be explained as below.

In A_M, the step-size parameter η_M can be any positive quantity [9]. For small values of η_M, the algorithm behaves essentially as a gradient-only-based algorithm. But, when η_M is chosen large compared to unity, the second term dominates the R.H.S of (5.9), and as a result (and because of (5.11)) the algorithm behaves essentially as the Newton algorithm A_N. Consequently, the best performance of A_M will be identical to that of A_N. We therefore will not compare A_N with A_M in the subsequent simulation studies.

We may point out here that the value of the error measure for large n (sample number) is smaller for 10 dB SNR, since convergence of $R(n)$ to R is faster when the SNR is higher.

Values of the orthogonality measure, $Orth_{\max}(n)$, calculated at each data instant and averaged over 100 Monte Carlo runs, are shown in Fig. 4. It is of the order of 10^{-4} for the Newton case, implying that the implicit orthogonalization built into the algorithm through the inflation technique is very effective. On the other hand, values of $Orth_{\max}(n)$ are very high for A_E and A_A (eg. 0.1152 and 0.1832 for A_A and A_E, respectively, at 250th data instant for 10 dB SNR), implying the inferiority of data level inflation.
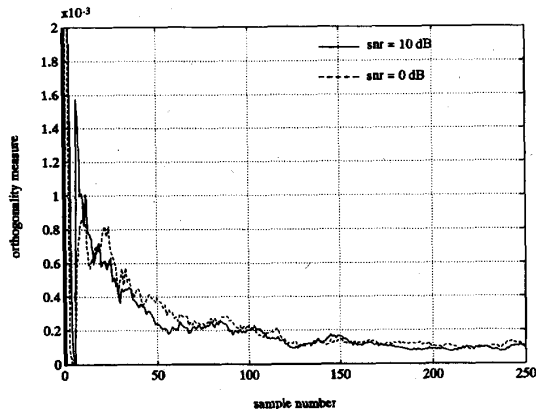
In some applications of adaptive subspace estimation, we would like the estimated subspace at nth data instant to be close to the corresponding subspace of $R(n)$. To study this, we did the following simulations. The projection error measure at nth data instant was evaluated using the corresponding subspace of $R(n)$ as the reference. Since $R(n)$ is deficient in full-rank until $n = N - 1$, we computed the error measure from $n = N$ onwards instead from $n = 1$. The averaged results (over 100 trials) are plotted in Fig. 5. Note from these plots that the initial convergence performance of A_N is much superior to that of A_E and A_A. This suggests that the Newton based algorithm A_N will have an edge over A_A and A_E under nonstationary signal scenarios.

Nonstationary Signal Scenario: The signal scenario used for nonstationary simulations is as follows. As before, we considered sinusoids in white noise, but the frequencies of the sinusoids were now made to vary in a linear fashion. The data $x(n)$ were generated as

$$x(n) = A\sin(2\pi f_1(n)n) + A\sin(2\pi f_2(n)n + \theta) + v(n)$$
$$n = 1, 2, \cdots \qquad (5.17)$$

where $f_1(n)$ and $f_2(n)$ are the frequencies (normalized) of the two sinusoids at nth instant which were varied from 0.05 to 0.25 Hz, and 0.1 to 0.25 Hz, respectively, over a data length of 200 samples. The values of N and D were chosen as 6 and 2, respectively.

The data covariance matrix was computed as the sum of outer products of the L most recent data vectors (i.e., a sliding rectangular window of length L). This results in the following rank-2 updating of the covariance matrix

$$R(n) = \sum_{i=0}^{L-1} x(n-i)x^T(n-i)$$
$$= R(n-1) + x(n)x^T(n)$$
$$- x(n-L)x^T(n-L). \qquad (5.18)$$

We chose L as 40 in our simulations. The change in the frequency of the first sinusoid over this interval of 40 samples

(a)

(b)

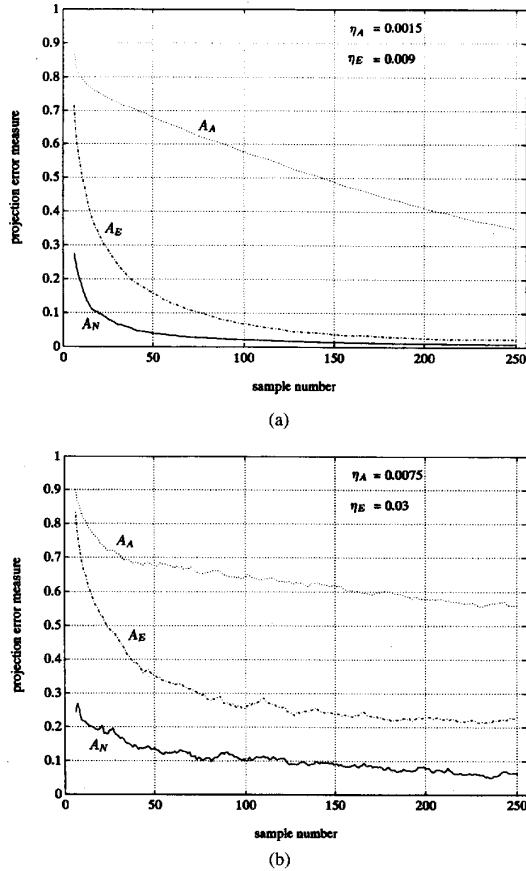Fig. 5.   Convergence performance of the three algorithms (A_N, A_A and A_E) with the subspace of $R(n)$ as the reference (stationary signal scenario): (a) SNR = 10 dB; (b) SNR = 0 dB.

is 0.04 Hz. Applying the matrix inversion lemma twice, we get the inverse of $R(n)$ as

$$R^{-1}(n) = \check{R}^{-1}(n)$$
$$- \frac{\check{R}^{-1}(n)x(n-L)x^T(n-L)\check{R}^{-1}(n)}{-1 + x^T(n-L)\check{R}^{-1}(n)x(n-L)} \quad (5.19)$$

where

$$\check{R}^{-1}(n) = R^{-1}(n-1)$$
$$- \frac{R^{-1}(n-1)x(n)x^T(n)R^{-1}(n-1)}{1 + x^T(n)R^{-1}(n-1)x(n)}. \quad (5.20)$$

Here, $\check{R}(n) = R(n-1) + x(n)x^T(n)$. Then, (3.12)–(3.14) with (5.20) and (5.19) describe the Newton type algorithm (A_N) for the nonstationary case. The corresponding exact gradient algorithm (A_E) is given by (5.6), (5.7), (5.3), and (5.4) with the covariance matrices $R'_k(n), k = 1, \cdots, D$ updated as

$$R'_k(n) = R'_k(n-1) + x_k(n)x^T_k(n)$$
$$- x_k(n-L)x^T_k(n-L) \qquad k = 1, \cdots, D.$$
$$(5.21)$$

We did not apply the approximate algorithm (A_A) for the nonstationary case since the updating of the data covariance

matrix as above does not arise in this algorithm, as it is based on the instantaneous estimate of the covariance matrix.

The two algorithms, A_N and A_E, were initialized as follows. Keeping the frequencies $f_1(n)$ and $f_2(n)$ fixed at 0.05 and 0.1 Hz, respectively, L data samples, $x(n), n = 1, \cdots, L$, were generated (cf. (5.17)). Then, $R(n)$ and $R^{-1}(n)$ were initialized as $R(L) = \Sigma^L_{i=1}x(i)x^T(i)$ and $R^{-1}(L)$, respectively. The coefficient vectors, $a_k(n)$ and $a'_k(n)$ for $k = 1, \cdots, D$, were initialized as $a_k(L) = a'_k(L) = q_k(L)$ where $q_k(L)$ is the kth eigenvector of $R(L)$. Then, the modified data vectors $x_k(n), n = 1, \cdots, L$, (in the case of A_E) were generated using (5.4) with $\bar{a}'_k(n) = q_k(L), k = 1, \cdots, D$, for all $n = 1, \cdots, L$, and the matrices $R'_k(n)$ were initialized as $R'_k(L) = \Sigma^L_{i=1}x_k(i)x^T_k(i), k = 1, \cdots, D$. Since $q_k(L)$, $k = 1, \cdots, D$, are the orthonormal eigenvectors of $R(L)$, we get

$$R'_k(L) = R'_{k-1}(L) + 3\lambda_{k-1}(L)q_{k-1}(L)q^T_{k-1}(L)$$
$$k = 2, \cdots, D$$

where $R'_1(L) = R(L)$ and $\lambda_k(L)$ is the kth eigenvalue of $R(L)$.

From the $(L+1)$st sample onwards, the frequencies of the two sinusoids were varied in steps of 0.001 and 0.00075 Hz per data sample, respectively. The algorithms (A_N and A_E) were run on this data and the projection error measure was computed using the subspace of $R(n)$ as the reference (cf. (5.18)). The results, averaged over 100 Monte Carlo runs, are shown in Fig. 6 for a SNR of 10 dB. Note that the error measure is zero at $n = 40$ since the algorithms were initialized to the eigenvectors of $R(40)$. The sudden increase in the error for $41 \leq n \leq 50$ is due to the fact that the reference subspace of $R(n)$ has changed significantly from that of $R(40)$ and the algorithms could not track this sudden change. From $n = 50$ onwards, the algorithms begin to track the subspace with the A_N performing much superior compared to A_E. The fluctuations seen in these plots are due to the fact that the change in the reference eigenvectors from instant to instant is quite significant because of the nonstationarity of the signal. The plots show that the Newton type algorithm tracks the subspace quite satisfactorily. The inferior performance of A_E can be attributed to its slow convergence behaviour.

Based on these results, we conclude that the proposed Newton-cum-inflation-based approach (A_N) for adaptively estimating the eigensubspace is much superior to the inflation-based gradient algorithms A_A and A_E, and is identical to the maximization gradient algorithm A_M.

VI. COMPUTATIONAL REQUIREMENTS

In this section, we give the computational requirements (in terms of the number of square roots, multiplications, divisions, and additions required per iteration) of the four algorithms considered above (for the stationary case). Because of the parallelism that is present in all these approaches, the effective computational load is equal to that required for updating one eigenvector estimate. Table I lists this computational load. Note that the proposed approach needs more multiplications compared to the approximate and exact gradient algorithms.
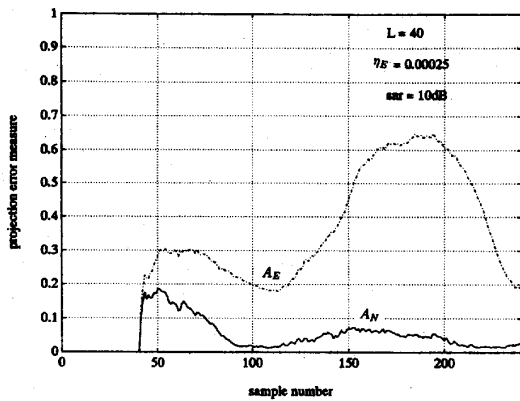
Fig. 6. Tracking performance of the two algorithms (A_N and A_E) with the subspace of $R(n)$ as the reference (nonstationary signal scenario with sliding rectangular window of length 40).

TABLE I
COMPUTATIONAL REQUIREMENTS OF THE FOUR ALGORITHMS

Operation	Proposed Newton (A_N)	Exact gradient (A_E)	Approximate gradient (A_A)	Maximization gradient (A_M)
Square-roots	0	1	1	1
Multiplications	$3.5N^2 + 5.5N + 1$	$2.5N^2 + 6.5N$	$5N + 1$	$3.5N^2 + 5.5N$
Divisions	3	2	1	3
Additions	$2.5N^2 + 2.5N + 3$	$1.5N^2 + 4.5N$	$4N$	$2.5N^2 + 3.5N + 1$

However, in view of the decreasing trend in computing costs, the excess computations required are not significant.

VII. CONCLUSIONS

The problem of obtaining the orthogonal eigenvectors of a symmetric positive definite matrix has been formulated as an unconstrained minimization problem and a Newton-type adaptive algorithm has been developed for estimating these eigenvectors in the data case. This algorithm makes use of an implicit orthogonalization procedure which is built into it through an inflation technique.

The algorithm is highly modular in nature and suitable for parallel implementation. Convergence analysis of this algorithm is also presented. Its performance (evaluated under stationary and nonstationary signal scenarios) has been found to be superior to that of the inflation-based algorithm proposed in [9] and an improved version of it, and is identical to the maximization gradient search procedure of [9]. Projection error measure and orthogonality measure are used in evaluating the performance.

APPENDIX

In this appendix, we prove Theorem 1. But first, we develop the following preliminary result.

Lemma A.1: Consider a matrix D of size $N \times N$. Suppose $\forall i$, there exists $|\epsilon_i| < \epsilon$ such that the elements of D are given by

$$d_{ii} = \tilde{d} + \epsilon_i \quad \text{and} \quad |d_{ij}| < \epsilon \qquad \forall i \neq j. \qquad (A.1)$$

Then, $\forall y \in \mathbb{R}^N$

$$\|Dy\|_\infty \geq [|\tilde{d}| - N\epsilon]\|y\|_\infty. \qquad (A.2)$$

Proof: Expressing $D = \tilde{d}I_N + \tilde{D}$, we get from (A.1), $\|\tilde{D}\|_\infty < N\epsilon$. It then follows that

$$\|Dy\|_\infty \geq |\tilde{d}|\,\|y\|_\infty - \|\tilde{D}y\|_\infty$$
$$\geq |\tilde{d}|\,\|y\|_\infty - N\epsilon\|y\|_\infty.$$

Proof of Theorem 1: Let $\Phi(n) = l(n-1)[\Lambda^{-1} + \Psi(n)]$. Partition $b(n)$ as

$$b(n) = [b_1^T(n), b_2^T(n)]^T \qquad (A.3)$$

where $b_1(n) = [b_1(n), \cdots, b_M(n)]^T$ and $b_2(n) = [b_{M+1}(n), \cdots, b_N(n)]^T$. Similarly, partition $\Phi(n)$ into $\Phi_{11}(n)$, $\Phi_{12}(n)$, $\Phi_{22}(n)$ and $\Phi_{21}(n)$ so that (4.5) becomes

$$\begin{bmatrix} b_1(n) \\ b_2(n) \end{bmatrix} = \begin{bmatrix} \Phi_{11}(n) & \Phi_{12}(n) \\ \Phi_{21}(n) & \Phi_{22}(n) \end{bmatrix} \begin{bmatrix} b_1(n-1) \\ b_2(n-1) \end{bmatrix}. \qquad (A.4)$$

Then, we have the following

$$f(n) = \frac{\|b_2(n)\|_\infty}{\|b_1(n)\|_\infty}$$
$$\leq \frac{\|\Phi_{22}(n)\|_\infty \|b_2(n-1)\|_\infty + \|\Phi_{21}(n)\|_\infty \|b_1(n-1)\|_\infty}{\|\Phi_{11}(n)b_1(n-1)\|_\infty - \|\Phi_{12}(n)b_2(n-1)\|_\infty}. \qquad (A.5)$$

Because of (4.8) and (4.10d), we can write $\forall n \geq n_0$

$$|\phi_{ij}(n)| < \epsilon_1(n) \qquad \forall i \neq j \qquad (A.6)$$
$$|\phi_{ii}(n) - \phi_{jj}(n)| < \epsilon_1(n) \qquad \forall i, j \in \{1, \cdots, M\} \qquad (A.7)$$

where $\epsilon_1(n)$ is such that $0 < \epsilon_1(n) < \epsilon_1$ and $\lim_{n \to \infty} \epsilon_1(n) = 0$. Denoting $\phi_{ii}(n) = \phi_i(n)$ and using Lemma A.1, (A.6), (4.10), and (A.7), we can express (A.5) as

$$f(n) \leq \frac{1}{\phi_1(n) - N\epsilon_1(n)(1 + f(n-1))}[\phi_{M+1}(n) \cdot f(n-1) + N\epsilon_1(n)] \qquad \forall n \geq n_0. \qquad (A.8)$$

Now, choose $\epsilon_1 < \delta_5\delta_4/8N$. Then, if $f(n-1) < 1$, using (4.10), and (4.9), we can reduce (A.8) to

$$f(n) \leq \frac{1}{\left(1 - \frac{\delta_4}{4}\right)}\left[(1 - \delta_4)f(n-1) + \frac{N\epsilon_1(n)}{\phi_1(n)}\right] \qquad (A.9)$$
$$\leq \frac{1}{\left(1 - \frac{\delta_4}{4}\right)}\left[(1 - \delta_4)f(n-1) + \frac{\delta_4}{8}\right]. \qquad (A.10)$$

Thus, in view of (4.15), (A.10) implies that $f(n) < 1 \forall n \geq n_0$. Now, consider (A.9). Since $\phi_1(n) > \delta_5$, it can be written as

$$f(n) \leq \frac{(1 - \delta_4)}{\left(1 - \frac{\delta_4}{4}\right)}f(n-1) + \frac{N}{\delta_5\left(1 - \frac{\delta_4}{4}\right)}\epsilon_1(n). \qquad (A.11)$$

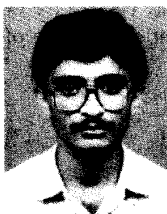Since $(1 - \delta_4) < (1 - \delta_4/4)$ and $\lim_{n \to \infty} \epsilon_1(n) = 0$, we conclude from (A.11) that

$$\lim_{n \to \infty} f(n) = 0.$$

ACKNOWLEDGMENT

REFERENCES

[1] N. L. Owsley, "Adaptive data orthogonalization," *Proc. IEEE Int. Conf. ASSP*, 1978, pp. 109–112.

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

[3] M. G. Larimore, "Adaptive convergence of spectral estimation based on Pisarenko harmonic retrieval," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, no. 4, pp. 955–962, Aug. 1983.

[4] 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, no. 3, pp. 399–405, June 1982.

[5] T. S. Durrani and K. C. Sharman, "Eigenfilter approaches to adaptive array processing," *IEE Proc.*, vol. 130, pts. F and H, no. 1, pp. 22–28, Feb. 1983.

[6] R. J. Vaccaro, "On adaptive implementations of Pisarenko's harmonic retrieval method," *Proc. IEEE Int. Conf. ASSP*, pp. 6.1.1–6.1.4, 1984.

[7] D. R. Fuhrmann and B. Liu, "Rotational search methods for adaptive Pisarenko harmonic retrieval," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 6, pp. 1550–1565, Dec. 1986.

[8] K. C. Sharman, "Adaptive algorithms for estimating the complete covariance eigenstructure," *Proc. IEEE Int. Conf. ASSP*, 1986, pp. 1401–1404.

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

[10] G. Mathew and V. U. Reddy, "Development and analysis of a neural network approach to Pisarenko's harmonic retrieval method," *IEEE Trans. Signal Processing*, vol. 42, no. 3, pp. 663–667, Mar. 1994.

[11] J. R. Bunch, C. P. Neilson, and D. C. Sorensen, "Rank-one modification of the symmetric eigenproblem," *Numerische Mathematik*, vol. 31, pp. 31–48, 1978.

[12] R. D. DeGroat and R. A. Roberts, "Efficient numerically stabilized rank-one eigenstructure updating," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 2, pp. 301–316, Feb. 1990.

[13] K. B. Yu, "Recursive updating the eigenvalue decomposition of a covariance matrix," *IEEE Trans. Signal Processing*, vol. 39, no. 5, pp. 1136–1145, May 1991.

[14] C. H. Bischof and G. M. Shroff, "On updating signal subspaces," *IEEE Trans. Signal Processing*, vol. 40, no. 1, pp. 96–105, Jan. 1992.

[15] R. D. DeGroat, "Noniterative subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, no. 3, pp. 571–577, Mar. 1992.

[16] G. W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Processing*, vol. 40, no. 6, pp. 1535–1541, June 1992.

Vellenki U. Reddy (S'68–M'70–SM'82) received the B.E. and M.Tech. degrees in electronics and communication engineering from Osmania University and the Indian Institute of Technology (IIT), Kharagpur, India, in 1962 and 1963, respectively, and the Ph.D. degree in electrical engineering from the University of Missouri in 1971.

He was an Assistant Professor at IIT, Madras, India, during 1972–1976 and a Professor at IIT, Kharagpur, during 1976–1979. During 1979–1982 and 1986–1987, he was a Visiting Professor at the Department of Electrical Engineering, Stanford University. In April 1982, he joined Osmania University as a Professor, and was the Founder-Director of the Research and Training Unit for Navigational Electronics, funded by the Department of Electronics, Government of India. Since April, 1988, he has been with the Indian Institute of Science, Bangalore, as a Professor of Electrical Communication Engineering, and is presently the Chairman of the Department. He has served as a consultant in the signal processing area to several R&D laboratories in Hyderabad and Bangalore. His research interests are in sensitivity analysis of high-resolution algorithms, adaptive algorithms, adaptive arrays, and wavelet transforms.

Dr. Reddy is a Fellow of the Indian Academy of Sciences, Indian National Academy of Engineering, and Indian National Science Academy and the Institution of Electronics and Telecommunication Engineers (IETE), India. He received the S. K. Mitra Memorial Award (1989) from IETE for the best research paper.
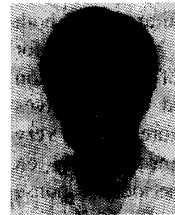
Soura Dasgupta (M'87–SM'93) was born in 1959 in Calcutta, India. He received the B.E. degree in electrical engineering from the University of Queensland, Australia, in 1980, and the Ph.D. degree in systems engineering from the Australian National University in 1985.

He is currently Professor with the Department of Electrical and Computer Engineering at the University of Iowa, Iowa City. In 1981, he was a Junior Research Fellow in the Electronics and Communication Sciences Unit at the Indian Statistical Institute, Calcutta. He has held visiting appointments at the University of Notre Dame, University of Iowa, Université Catholique de Louvain-La-Neuve, Belgium, and the Australian National University. His research interests are in controls, signal processing, and neural networks.

Between 1988 and 1991, he served as an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He is a corecipient of the Gullimen Cauer Award for the best paper published in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS in the calendar years of 1990 and 1991, and is a Presidential Faculty Fellow.

G. Mathew received the B.E. degree in electronics and communication engineering from Karnataka Regional Engineering College, Surathkal, India, in 1987, and the M.Sc. (Engg.) and Ph.D. degrees in electrical communication engineering from the Indian Institute of Science (IISc), Bangalore, in 1989 and 1994, respectively.

He is currently working as a Project Associate in the defense-sponsored signal processing and communications research program at IISc. His research interests are in adaptive algorithms and neural networks.