Adaptive Eigendecomposition of Data Covariance Matrices Based on First-Order Perturbations

Benoit Champagne, *Member, IEEE*

Abstract—**In this paper, new algorithms for adaptive eigendecomposition of time-varying data covariance matrices are presented. The algorithms are based on a first-order perturbation analysis of the rank-one update for covariance matrix estimates with exponential windows. Different assumptions on the eigenvalue structure lead to three distinct algorithms with varying degrees of complexity. A stabilization technique is presented and both issues of initialization and computational complexity are discussed. Computer simulations indicate that the new algorithms can achieve the same performance as a direct approach in which the exact eigendecomposition of the updated sample covariance matrix is obtained at each iteration. Previous algorithms with similar performance require $O(LM^2)$ complex operations per iteration, where L and M respectively denote the data vector and signal-subspace dimensions, and involve either some form of Gram-Schmidt orthogonalization or a nonlinear eigenvalue search. The new algorithms have parallel structures, sequential operation counts of order $O(LM)^2$ or less, and do not involve any of the above steps. One particular algorithm can be used to update the complete signal-subspace eigenstructure in $5LM$ complex operations. This represents an order of magnitude improvement in computational complexity over existing algorithms with similar performance. Finally, a simplified local convergence analysis of one of the algorithms shows that it is stable and converges in the mean to the true eigendecomposition. The convergence is geometrical and is characterized by a single time constant.**

I. Introduction

During the last 15 years, signal-subspace algorithms based on the eigendecomposition of the covariance matrix of a random data vector have been applied successfully to both temporal and spatial-domain high-resolution spectral analysis [1]–[2]. In practice, these algorithms are often implemented in a batch mode, using a sample covariance matrix obtained by collecting data vectors over a sufficiently long observation interval. This approach, which relies on the assumption of statistical stationarity of the data, cannot be used in situations where the characteristics of the received signals change with time. In this case, the direct application of signal-subspace methods requires repeated eigendecompositions of a continually updated sample covariance matrix, a task which is generally prohibitive.

Recently, several adaptive eigendecomposition algorithms have been proposed for the efficient application of signal-subspace methods in nonstationary environments. Instead of recomputing the eigendecomposition estimate from scratch with every update of the sample covariance matrix, these algorithms attempt to recursively update the eigendecomposition so as to minimize the amount of computations involved. At least two major families of adaptive eigendecomposition algorithms can be identified, depending on the type of approach used to obtain the desired recursion.

In the first family (e.g., [3]–[6]), the determination of the signal subspace is formulated as a constrained optimization problem. The latter is solved adaptively via a stochastic gradient search over time, using either a recursive or instantaneous estimate of the data covariance matrix to evaluate the gradient vector. These algorithms generally involve some form of Gram-Schmidt orthogonalization at each iteration. The most computationally efficient algorithms in this family require on the order of $O(LM)$ operations per iteration, where L is the dimension of the data vector and M is the dimension of the eigensubspace being tracked. However, only those algorithms with complexity $O(LM^2)$ can achieve a level of performance comparable to direct eigendecomposition of the updated sample covariance matrix.

Algorithms in the second family (e.g., [7]–[10]) are based on variations and extensions of a rank-one updating algorithm for the symmetric eigenvalue problem originally proposed by Golub [11] and later improved by Bunch *et al.* [12]. These algorithms exploit the low-rank nature of the additive modification terms which are used to update the sample covariance matrix in nonstationary applications. Apart from a possible buildup of roundoff errors, these algorithms permit calculation of the exact eigendecomposition of the updated sample covariance matrix much more efficiently than a direct approach. They involve the solution of a nonlinear equation for the calculation of the eigenvalues and they require on the order of $O(LM^2)$ operations per iteration.

In this paper, a novel approach to the adaptive eigendecomposition problem is proposed, and then several new algorithms are derived and evaluated. Starting from a rank-one recursive update for sample covariance matrices with exponential windows, the proposed approach consists of treating the additive modification term in this recursion as a perturbation, with the (small) forgetting factor playing the role of a perturbation parameter. Following this interpretation, a first-order perturbation analysis is made to obtain an approximate recursion expressing the eigendecomposition estimates at time k in terms of those at time $k-1$ and the new data vector observed at time k. By making different assumptions on the eigenvalue structure, three algorithms with varying degrees

Manuscript received March 1, 1993, revised February 16, 1994. Support for this work was provided by NSERC operating Grant OGP0105533. The associate editor coordinating the review of this paper and approving it for publication was Dr. R. D. Preuss.

The author is with INRS-Télécommunications, Université du Québec, Verdun, Québec, Canada H3E 1H6.

IEEE Log Number 9403761.

of complexity (denoted PA, PB, and PC, respectively) are obtained.

For the particular case of tracking a signal-subspace of dimension M, the resulting algorithms can be realized by means of M linear combiners with nonlinear weight-vector adaptation equations. Moreover, they do not involve an explicit Gram-Schmidt orthogonalization step nor the solution of a nonlinear equation for the eigenvalues. Simulation results for narrow-band array data indicate improved performance of the proposed algorithms when compared to the family of adaptive algorithms proposed in [5]. In particular, the algorithm PC can achieve the same level of performance as direct eigendecomposition of the sample covariance matrix with only $5LM$ operations per iteration. This represents a major improvement over existing algorithms which require on the order of $O(LM^2)$ operations per iteration to achieve the same performance. A simplified local convergence analysis of the algorithm PA shows that it is stable and converges in the mean to the true eigendecomposition. The convergence is geometrical, with all adaptive modes characterized by the same time constant.

The paper is organized as follows. The data model and the formulation of the adaptive eigendecomposition problem as a perturbation problem are discussed in Section II. The first-order perturbation analysis and the derivation of the new algorithms are presented in Section III, where a stabilization technique and the issues of initialization and computational complexity are also considered. Section IV presents the results of computer simulations. Finally, Section V provides some conclusions. The convergence analysis of the algorithm PA is outlined in Appendix I.

II. Formulation of the Problem

Let $x(k)$ denote the complex L-dimensional data vector observed at time k. In the context of time series analysis, $x(k)$ corresponds to the kth signal frame, which is made up of L consecutive signal samples. In array-processing, $x(k)$ may be identified with the narrow-band output of an array of L sensors over the kth snapshot. The data vector $x(k)$ is modeled as a linear superposition of $M < L$ signal components in additive background noise. More precisely

$$x(k) = A(k)s(k) + n(k) \tag{1}$$

where $A(k)$ is a complex $L \times M$ matrix whose columns represent the signal waveforms (typically, but not necessarily, complex exponentials), $s(k)$ is a complex M-dimensional vector containing the amplitudes of the signals, and $n(k)$ is a complex L-dimensional vector representing background noise. It is assumed that $s(k)$ and $n(k)$ are zero-mean, uncorrelated, vector random processes with covariance matrices at time k respectively given by

$$R_s(k) = E[s(k)s^H(k)] \tag{2}$$

$$R_n(k) = \sigma_n^2(k)I_L \tag{3}$$

where the superscript ^{H} denotes complex conjugate transposition, $\sigma_n^2(k)$ is the noise variance at time k, and I_L is the $L \times L$ identity matrix. From (1)–(3), the data covariance matrix is obtained as

$$R_x(k) = A(k)R_s(k)A^H(k) + \sigma_n^2(k)I_L. \tag{4}$$

Let $\lambda_i(k)$ and $q_i(k)$, $i = 1, \ldots, L$, denote the eigenvalues and corresponding orthonormalized eigenvectors of the data covariance matrix $R_x(k)$, with the eigenvalues arranged in decreasing order. Assuming that $A(k)$ is of full rank and that $R_s(k)$ is positive-definite, it can be shown that

$$\lambda_1(k) \geq \cdots \geq \lambda_M(k) > \lambda_{M+1}(k) = \cdots = \lambda_L(k) = \sigma_n^2(k) \tag{5}$$

and

$$A^H(k)Q_n(k) = 0 \tag{6}$$

where 0 denotes the $M \times (L - M)$ zero matrix and

$$Q_n(k) = [q_{M+1}(k), \ldots, q_L(k)]. \tag{7}$$

Hence, the number of source signals, M, is given indirectly by the multiplicity of the smallest eigenvalue, while the column span of the matrix $A(k)$ is identical to that of

$$Q_s(k) = [q_1(k), \ldots, q_M(k)]. \tag{8}$$

The column spans of $Q_s(k)$ and $Q_n(k)$ are appropriately referred to as the signal- and noise-subspaces, respectively.

Typical signal-subspace algorithms such as MUSIC (see paper by Schmidt in [1]) exploit the information embedded in the eigendecomposition of the data covariance matrix to achieve high-resolution spectral analysis of the observed data. In stationary environments, these algorithms are often implemented in a batch mode (i.e., off-line). This involves the following steps: collecting data vectors $x(k)$ for $k = 1, \ldots, K$, where the number of observations K is sufficiently large to minimize the effects of noise; forming an estimate $\hat{R}_x$ of the data covariance matrix as in

$$\hat{R}_x = \frac{1}{K}\frac{1}{K}\sum_{k=1}^{K} x(k)x^H(k); \tag{9}$$

performing the eigendecomposition of $\hat{R}_x$; and, finally, using the eigendecomposition to extract the relevant spectral information. Under standard assumptions of stationarity, the performance of signal-subspace algorithms based on (9) is known to improve as K increases [2].

In several applications of interest, however, the assumption of stationarity is valid only over a limited time interval due to changes in the characteristics of the signal and noise. In array processing, for instance, the direction of arrivals of the signals of interest may change with time as the sources move. In this case, it is convenient to estimate the data covariance matrix recursively, in a manner that de-emphasizes past observations. The following recursion is often used in applications:

$$\hat{R}_x(k) = \alpha\hat{R}_x(k-1) + (1-\alpha)x(k)x^H(k) \tag{10}$$

where α is a real smoothing factor such that $0 \leq \alpha \leq 1$. Equation (10), referred to as a rank-one update of the sample covariance matrix, is equivalent to exponential time averaging

of the sequence $x(k)x^H(k)$. In this respect, the factor $1/(1-\alpha)$ provides a rough measure of the effective length of the exponential window.

The real-time (i.e., on-line) implementation of signal-subspace algorithms based on a recursive update of the sample covariance matrix, as in (10), requires repeated eigendecomposition of the time-varying matrix $\hat{R}_x(k)$. The computational costs associated with performing this eigendecomposition from scratch at each iteration are very high, and in many instances, prohibitive. As indicated in Section I, two major families of adaptive eigendecomposition algorithms have been proposed recently to overcome this difficulty. Instead of performing the eigendecomposition of the updated matrix $\hat{R}_x(k)$ from scratch, these algorithms recursively update the eigendecomposition over time (either exactly, or approximately) so that the overall computational load is reduced.

We now present an alternative formulation of the adaptive eigendecomposition problem as a first-order perturbation problem. In Section III, this formulation will lead to a new family of adaptive algorithms with varying degrees of complexity. Let (10) be written as

$$\hat{R}_x(k) = \hat{R}_x(k-1) + \varepsilon[x(k)x^H(k) - \hat{R}_x(k-1)] \quad (11)$$

where $\varepsilon = 1 - \alpha$, with $0 \leq \varepsilon \leq 1$. For ε sufficiently small, the modification term $\varepsilon[x(k)x^H(k) - \hat{R}_x(k-1)]$ in (11) can be interpreted as a small perturbation of $\hat{R}_x(k-1)$. Intuitively, then, we expect the eigendecomposition of $\hat{R}_x(k)$ to be related to that of $\hat{R}_x(k-1)$ through small correction terms which are well-behaved functions of ε. This argument can be formalized as follows.

To begin with, note that $\hat{R}_x(k)$ (11) is expressed as a power series in ε, converging to $\hat{R}_x(k-1)$ in the limit $\varepsilon \to 0$. Moreover, assuming that $\hat{R}_x(0)$ is Hermitian, it follows from (11) that $\hat{R}_x(k)$ is Hermitian for all positive integers k and for all real ε. According to a fundamental theorem from the theory of perturbation of Hermitian matrices [13], it follows that the eigenvalues and normalized eigenvectors of $\hat{R}_x(k)$ can also be expanded in power series in ε, converging to the respective eigenvalues and eigenvectors of $\hat{R}_x(k-1)$ in the limit $\varepsilon \to 0$. More precisely, let $\gamma_i(k)$ and $u_i(k)$ denote the eigenvalues and normalized eigenvectors of the sample covariance matrix $\hat{R}_x(k)$ (11), that is,

$$\hat{R}_x(k)u_i(k) = \gamma_i(k)u_i(k) \quad (12)$$

$$u_i{}^H(k)u_j(k) = \delta_{ij} \quad (13)$$

where δ_{ij} is the Kronecker delta. According to Theorem 1 in Chapter 1 of [13], there exist power series

$$\gamma_i(k) = \gamma_{i0} + \gamma_{i1}\varepsilon + \gamma_{i2}\varepsilon^2 + \cdots, \quad \gamma_{i0} = \gamma_i(k-1) \quad (14)$$

$$u_i(k) = u_{i0} + u_{i1}\varepsilon + u_{i2}\varepsilon^2 + \cdots, \quad u_{i0} = u_i(k-1) \quad (15)$$

all convergent in a neighborhood of $\varepsilon = 0$, which satisfy the eigenvalue problem (12)–(13). In (14) and (15), $\gamma_{i0} = \gamma_i(k-1)$ and $u_{i0} = u_i(k-1)$ are the eigenvalues and normalized eigenvectors of $\hat{R}_x(k-1)$, while γ_{ij} and $u_{ij} (j \geq 1)$ are unknown coefficients that must be determined.

This general result is valid regardless of the multiplicities of the unperturbed eigenvalues γ_{i0}. However, in the case of repeated eigenvalues (i.e., multiplicity > 1), a particular basis of unperturbed eigenvectors, $\{u_{i0}: i = 1, \ldots, L\}$, must be selected for the power series expansions (14) and (15) to exist. The theory of perturbation provides various methods for calculating the jth-order ($j \geq 1$) coefficients γ_{ij} and vector coefficients u_{ij} appearing in the expansions (14) and (15). In the case of repeated eigenvalues, the proper choice for $\{u_{i0}: i = 1, \ldots, L\}$ is also prescribed by the application of these methods.

In practice, perturbation series of the type (14) and (15) are truncated to yield tractable approximations. In particular, for n, a positive integer, an nth-order approximation results from neglecting all terms of order ε^m with $m > n$. This type of approximation is justified on the basis that the perturbation series are convergent in some neighborhood of $\varepsilon = 0$. Hence, provided ε is sufficiently small, a low-order approximation can be used to evaluate $\gamma_i(k)$ and $u_i(k)$ with good accuracy. In this paper, it is proposed to recursively update the eigendecomposition of the sample covariance matrix $\hat{R}_x(k)$ in (11) based on a first-order approximation of (14) and (15).

There are several reasons for choosing this approach. For low-order approximations, perturbation methods are known to provide numerical results comparatively quickly, along with significant physical insights. Another important aspect of a perturbation approach is that it makes an algebraically consistent use of the small parameter ε: all the calculations are done with the same degree of accuracy and no resources are spent on trying to achieve unnecessary precision. In this respect, algorithms of the second family identified in Section I (e.g., [7]–[10]) are somewhat inefficient. Finally, being algebraic in nature, the proposed perturbation approach is statistically robust in the sense that it does not rely on sophisticated modeling assumptions about the signal and noise.

There is a major conceptual difference between the approach proposed in this paper and the adaptive algorithms of the second family, in which the eigendecomposition of $\hat{R}_x(k)$ (11) is updated *exactly* at each iteration. Indeed, even if the exact eigendecomposition of $\hat{R}_x(0)$ is known initially, recursive updating based on a first-order approximation of (14) and (15) generally results in an *approximate* eigendecomposition of $\hat{R}_x(k)$ for $k \geq 1$. (This is also true for the algorithms of the first family.) However, whether or not the resulting estimates $\gamma_i(k)$ and $u_i(k)$ form the exact eigendecomposition of $\hat{R}_x(k)$ is not the primary concern. This is particularly true in nonstationary environments where the definition of $\hat{R}_x(k)$ itself is somewhat arbitrary. What is important is that $\gamma_i(k)$ and $u_i(k)$ be good estimators of the eigendecomposition of the true data covariance matrix $R_x(k)$ in (4). For the proposed perturbation approach, this is confirmed by simulation (Section IV) and by mathematical analysis (Appendix I).

In the next section, the first-order approximations to the power series (14)–(15) are obtained explicitly by applying the methods of perturbation theory. This leads to a new family of adaptive eigendecomposition algorithms with varying degrees of complexity.

III. First-Order Perturbation Analysis

The first step consists of deriving a basic set of equations for the first-order coefficients γ_{i1} and u_{i1} in (14) and (15). To simplify the notations, let

$$R_0 = \hat{R}_x(k-1) \tag{16}$$

$$R_1 = x(k)x^H(k) - \hat{R}_x(k-1) \tag{17}$$

so that (11) can be expressed as

$$\hat{R}_x(k) = R_0 + \varepsilon R_1. \tag{18}$$

In the following derivation, it is temporarily assumed that the zeroth-order coefficients $\gamma_{i0} = \gamma_i(k-1)$ and $u_{i0} = u_i(k-1)$ are the *exact* eigenvalues and orthonormalized eigenvectors of $R_0 = \hat{R}_x(k-1)$[1]. Furthermore (in the case of repeated eigenvalues), it is assumed that the set $\{u_{i0}: i = 1, \ldots, L\}$ has been properly selected so that the power series expansions (14) and (15) exist.

Substituting (14), (15) and (18) in (12) and multiplying term by term, one obtains

$$R_0u_{i0} + (R_0u_{i1} + R_1u_{i0})\varepsilon = \gamma_{i0}u_{i0} + (\gamma_{i0}u_{i1} + \gamma_{i1}u_{i0})\varepsilon + O(\varepsilon^2) \tag{19}$$

where $O(\varepsilon^2)$ represents terms of order ε^2. Since (19) is valid for all ε in a neighborhood of $\varepsilon = 0$, corresponding powers of ε must be equal. This yields

$$R_0u_{i0} = \gamma_{i0}u_{i0}, \tag{20}$$

$$R_0u_{i1} + R_1u_{i0} = \gamma_{i0}u_{i1} + \gamma_{i1}u_{i0}. \tag{21}$$

Note that (20) provides no new information; it is already satisfied by assumption.

Equation (21) can be expressed in a more convenient form by projecting it on the orthonormal basis vectors u_{j0} for $j = 1, \ldots, L$. Premultiplying (21) by u_{i0}^H, observing that $u_{i0}^H R_0 = \gamma_{i0}u_{i0}^H$, which follows from (20), and using the orthonormality of the set $\{u_{i0}: i = 1, \ldots, L\}$, one obtains

$$\gamma_{i1} = u_{i0}^H R_1 u_{i0}. \tag{22}$$

Similarly, premultiplying (21) by u_{j0}^H $(j \neq i)$ gives

$$(\gamma_{j0} - \gamma_{i0})u_{j0}^H u_{i1} = -u_{j0}^H R_1 u_{i0}, \quad j \neq i. \tag{23}$$

Another equation is obtained by substituting (15) in (13) and equating corresponding powers of ε:

$$u_{i1}^H u_{j0} + u_{i0}^H u_{j1} = 0. \tag{24}$$

The desired basic set of equations for γ_{i1} and u_{i1} is provided by (22)–(24). Equation (22) can be used to determine γ_{i1} explicitly, while the expansion coefficients of u_{i1} in the basis $\{u_{i0}: i = 1, \ldots, L\}$, i.e.

$$b_{ji} = u_{j0}^H u_{i1} \tag{25}$$

can be obtained from (23) and (24). Once the coefficients b_{ji} are known, u_{i1} is obtained from the reconstruction formula

$$u_{i1} = \sum_{j=1}^{L} b_{ji}u_{j0}. \tag{26}$$

Depending on the configuration of the unperturbed eigenvalues γ_{i0}, different cases occur in the solution of (22)–(24). In subsections A, B, and C below, three distinct cases of practical importance are considered. In each case, the coefficients γ_{i1} and u_{i1} are derived explicitly and the corresponding adaptive eigendecomposition algorithm is stated. The issues of initialization and computational complexity are addressed in subsections D and E.

A. Distinct Eigenvalues

In this subsection, (22)–(24) are solved for the first-order coefficients γ_{i1} and u_{i1}, under the assumption that the unperturbed eigenvalues γ_{i0} are distinct (i.e., multiplicity = 1). Let the unperturbed eigenvalues γ_{i0} be arranged in descending order, so that

$$\gamma_{10} > \gamma_{20} > \cdots > \gamma_{L0}. \tag{27}$$

Substituting (17) and (16) in (22) and using (20) together with the orthonormality of the unperturbed eigenvectors, we obtain

$$\gamma_{i1} = |y_i|^2 - \gamma_{i0} \tag{28}$$

where

$$y_i = u_{i0}^H x(k) \tag{29}$$

is the orthogonal projection coefficient of the data vector $x(k)$ on the unperturbed eigenvector u_{i0}.

Proceeding similarly with (23) and observing that $\gamma_{j0} - \gamma_{i0} \neq 0$ for $j \neq i$ under the assumption (27), one obtains

$$b_{ji} = -b_{ij}^* = \frac{y_i^* y_j}{(\gamma_{i0} - \gamma_{j0})}, \quad j \neq i. \tag{30}$$

Note that (30) is consistent with (24). The remaining coefficients b_{ii} can be obtained as follows. When $j = i$, (24) reduces to $\mathrm{Re}(b_{ii}) = 0$, where the notations Re(.) and Im(.) denote the real and imaginary parts of a complex number, respectively. Since no other constraint is imposed, $\mathrm{Im}(b_{ii})$ can be chosen arbitrarily. The simplest choice is $\mathrm{Im}(b_{ii}) = 0$, which results in

$$b_{ii} = 0. \tag{31}$$

According to (25), this means that u_{i1} is orthogonal to the corresponding unperturbed eigenvector u_{i0}.

Equations (28)–(31) (together with (26)) define the first-order perturbation coefficients γ_{i1} and u_{i1} explicitly in terms of the observed data vector $x(k)$ and the unperturbed eigendecomposition, i.e., $\gamma_{i0} = \gamma_i(k-1)$ and $u_{i0} = u_i(k-1)$ for $i = 1, \ldots, L$. Following the approach proposed in Section II, an adaptive algorithm for recursively updating the eigendecomposition estimates $\gamma_i(k)$ and $u_i(k)$ of the true data covariance matrix can now be obtained as follows: substitute γ_{i1} and u_{i1} calculated above in the power series expansion (14)–(15) and neglect all terms of order $O(\varepsilon^n)$ for

[1] Actually, this assumption can be relaxed by requiring that γ_{i0} and u_{i0} be *approximate* eigenvalues and orthonormalized eigenvectors of R_0, up to error terms of order $O(\varepsilon^2)$. This will not affect the following analysis since only first-order terms are considered.

$n \geq 2$. Some additional manipulations are needed to make the algorithm more parallel and to minimize the operation count. For instance, it is advantageous to perform the multiplication by ε directly on the input data $x(k)$, rather than on γ_{i1} and u_{i1} as in (14)–(15). Another simplification occurs if the unperturbed eigenvectors $u_{i0} = u_i(k-1)$ are multiplied by an appropriate complex number of absolute value 1 so as to make y_i (29) real. Observe that (20) and the orthonormality of the set $\{u_{i0}: i = 1, \ldots, L\}$ are not affected by this modification.

A stabilization mechanism is also needed to prevent an undesirable behavior of the algorithm, which may occur when the difference between successive eigenvalues is too small. To see this, consider a stationary signal model with true eigenvalues such that $\lambda_i = \lambda_j + \Delta$ for some $j > i$, where Δ is a small positive number, i.e. $\Delta \ll \lambda_i$. Assuming that the eigenvalue estimates $\gamma_i(k)$ and $\gamma_j(k)$ are well separated at time $k = 0$, the difference $\gamma_{i0} - \gamma_{j0} = \gamma_i(k-1) - \gamma_j(k-1)$ will initially decrease as the algorithm converges and the coefficient b_{ji} in (30) will become relatively large. Then, small random fluctuations of $\gamma_i(k-1) - \gamma_j(k-1)$ around the true difference $\gamma_{i0} - \gamma_{j0}$ will induce large variations in b_{ji}, limiting the steady-state performance of the algorithm, and in some cases making it unstable.

This problem can be overcome, without significantly altering the complexity of the algorithm, by thresholding the smallest possible eigenvalue difference used in the computation of b_{ji} (30) and by renormalizing the updated eigenvector $u_i(k)$ at regular interval. In this respect, the following thresholding approach has been found extremely useful:

$$b_{ji} = \frac{y_i^* y_j}{\max(\delta\gamma_{10}, \gamma_{i0} - \gamma_{j0})}, \quad i < j \tag{32}$$

where δ is a small positive number, typically 0.01. For $i > j$, b_{ji} is obtained from the symmetry relation in (30). When this stabilization technique is used, the algorithm can even handle the case of two repeated eigenvalues, as confirmed by simulations.

The stabilized version of the algorithm, denoted PA (for perturbation-A), is summarized in Table I where the notation $\|.\|$ indicates the Euclidean norm of a vector. In this algorithm, the scaled data vector $\sqrt{\varepsilon}x(k)$ is passed through L linear combiners (or tapped delay-lines in time series analysis) with complex weight vectors given by $u_i(k-1), i = 1, \ldots, L$. The outputs y_i of the combiners, which can be viewed as sufficient statistics, are then used in the calculation of the first-order correction terms. Finally, these are combined to $\gamma_i(k-1)$ and $u_i(k-1)$ to yield the updated eigendecomposition $\gamma_i(k)$ and $u_i(k)$. Note that only the observed data vector $x(k)$ and the previous eigendecomposition, i.e., $\gamma_{i0} = \gamma_i(k-1)$ and $u_{i0} = u_i(k-1)$ for $i = 1, \ldots, L$, enter the algorithm. Explicit knowledge of the sample data covariance matrix $R_0 = \hat{R}_x(k-1)$ is not needed, resulting in saving in data storage and computation rate.

A simplified local convergence analysis of algorithm PA is presented in Appendix I. The results show that for $0 < \varepsilon < 1$, the algorithm is stable and converges in the mean to the true eigendecomposition. The convergence is in $(1-\varepsilon)^k$ for all adaptive modes. As a result, the algorithm is characterized by a single time constant which is given by $\tau = 1/\varepsilon$ for ε small. This is a rather remarkable property of the perturbation approach. Finally, we note that the weight-vector adaptation equation provided by the algorithm PA is nonlinear (cubic) in the $u_i(k-1)$. Based on the results of the convergence analysis, this nonlinearity may be viewed as a substitute for the orthogonalization step found in other algorithms.

TABLE I
ADAPTIVE EIGENDECOMPOSITION ALGORITHM PA

$x(k) \leftarrow \sqrt{\varepsilon}\, x(k)$

for $i = 1$ to L:
  $\eta = u_i^H(k-1)\, x(k)$
  $y_i = |\eta|$ (y_i is now real)
  $u_i(k-1) \leftarrow (\eta/y_i)\, u_i(k-1)$
end

for $i = 1$ to L:
  $b_{ii} = 0$
  for $j = i+1$ to L:
    $b_{ji} = y_i y_j / \max(\delta\gamma_1(k-1), \gamma_i(k-1) - \gamma_j(k-1))$
    $b_{ij} = -b_{ji}$
  end
end

for $i = 1$ to L:
  $\gamma_i(k) = (1-\varepsilon)\gamma_i(k-1) + y_i^2$
  $u_i(k) = u_i(k-1) + \sum_{j=1}^{L} b_{ji} u_j(k-1)$
  $u_i(k) \leftarrow u_i(k)/\|u_i(k)\|$
end

B. Repeated Noise-Subspace Eigenvalue

In practical applications of signal-subspace algorithms, the number of data points, L, often exceeds the number of source signals, M, by 2 or more. According to (5), the smallest eigenvalue of the true data covariance matrix is then repeated with multiplicity $L - M > 1$. By explicitly incorporating this multiplicity in the perturbation analysis, it is possible to reduce the computational complexity and improve the convergence behavior of the algorithm PA. To this end, the following condition is imposed on the $L - M$ smallest eigenvalue estimates:

$$\gamma_{M+1}(k) = \cdots = \gamma_L(k) \equiv \rho(k) \tag{33}$$

for all k, where $\rho(k)$ is an estimate of the noise variance. In (33), it is implicitly assumed that the signal-subspace dimensionality, M, is known. In practice, M can be estimated from the data using one of the several criteria presently available for detecting the number of sources (e.g., [14]).

Some of the signal-subspace eigenvalues can also be repeated. By imposing additional equality constraints on the corresponding $\gamma_i(k)$, as in (33), it would be possible to further reduce the complexity of the algorithm PA. In practice, however, the signal subspace eigenvalues are rarely exactly

repeated; when they are, this is usually over a limited time interval and the corresponding multiplicity is small. As our experience indicates, the most effective way of handling temporarily repeated signal-subspace eigenvalues is to use a thresholding technique such as the one described in Part A. Therefore, to simplify the discussion, it is assumed that

$$\gamma_1(k) > \cdots > \gamma_M(k) > \rho(k) \tag{34}$$

for all k.

In terms of the unperturbed eigenvalues $\gamma_{i0} = \gamma_i(k-1)$, (33) and (34) become:

$$\gamma_{10} > \cdots > \gamma_{M0} > \gamma_{M+1,0} = \cdots = \gamma_{L0} \equiv \rho_0 \tag{35}$$

where $\rho_0 = \rho(k-1)$ is the unperturbed noise variance. In the following paragraphs, (22)–(24) are solved for γ_{i1} and u_{i1} under the condition (35). The following integer sets are used: $\Omega_1 = \{1, \ldots, M\}$ and $\Omega_2 = \{M+1, \ldots, L\}$. Moreover, the linear spans of $\{u_{i0}: i \in \Omega_1\}$ and $\{u_{i0}: i \in \Omega_2\}$ are referred to as the unperturbed signal- and noise-subspaces, respectively.

The major difficulty in solving (22)–(24) occurs with (23), when both $i, j \in \Omega_2$. The difference $\gamma_{j0} - \gamma_{i0}$ is then equal to zero and it is not possible to solve directly for $u_{j0}^H u_{i1}$. In this case, (23) must be interpreted as a constraint on the unperturbed noise-subspace eigenvectors $\{u_{i0}: i \in \Omega_2\}$. The necessity of this constraint can be explained as follows. Since γ_{L0} has multiplicity $L - M > 1$, there is a large freedom in the choice of the unperturbed noise-subspace eigenvectors: Any orthonormal basis of this subspace can be used. However, when an arbitrary perturbation is applied, the noise-subspace eigenvalues will generally become distinct and the corresponding eigenvectors will no longer be arbitrary (up to a phase term). To express these perturbed eigenvectors as power series in ε around the unperturbed ones (in other words, to connect both sets analytically), a particular basis of unperturbed eigenvectors must be selected. The role of (23) with $i, j \in \Omega_2$ is precisely to ensure that the proper choice is made.

When $i, j \in \Omega_2$ and $i \neq j$, (23) reduces to

$$y_i^* y_j = 0 \tag{36}$$

where y_i is defined in (29). To satisfy this constraint, the unperturbed eigenvectors $u_{i0}, i \in \Omega_2$, can be selected as follows. First, define

$$x_s = \sum_{i=1}^{M} y_i u_{i0}, \quad x_n = x(k) - x_s. \tag{37}$$

The vectors x_s and x_n are the orthogonal projections of the data vector $x(k)$ on the unperturbed signal- and noise-subspaces, respectively. By construction, they are orthogonal: $x_n^H x_s = 0$. Now, set

$$u_{M+1,0} = x_n / \|x_n\| \tag{38}$$

and finally, select $u_{i0}, i = M+2, \ldots, L$, arbitrarily within the unperturbed noise-subspace, subject to the constraint that $\{u_{i0}: i \in \Omega_2\}$ form an orthonormal basis of this subspace. Then, using (29), it is easy to verify that

$$y_i = \begin{cases} \|x_n\|, & i = M+1 \\ 0, & i = M+2, \ldots, L \end{cases} \tag{39}$$

and (36) is automatically satisfied.

This choice of $\{u_{i0}: i \in \Omega_2\}$, which is prescribed here by (23), is equivalent to that made in [7] for the rank-one eigenstructure updating problem. We note that several techniques can be used to construct the eigenvectors $u_{i0}, i = M+2, \ldots, L$, including Gram-Schmidt orthogonalization and Householder reflections [17]. However, if one is interested only in updating the signal-subspace, explicit construction of these eigenvectors is not necessary (see below).

Having made the proper choice of $\{u_{i0}: i \in \Omega_2\}$, the remaining equations in (22)–(24) can now be used to determine γ_{i1} and u_{i1}. From (22), it follows that

$$\gamma_{i1} = \begin{cases} |y_i|^2 - \gamma_{i0}, & i = 1, \ldots, M \\ \|x_n\|^2 - \rho_0, & i = M+1 \\ -\rho_0, & i = M+2, \ldots, L \end{cases} \tag{40}$$

If (40) is used directly in a first-order perturbation expansion to update the eigenvalues $\gamma_{i0} = \gamma_i(k-1), i = 1, \ldots, L$, which satisfy (35) by assumption, the multiplicity of the smallest eigenvalue estimate will decrease by one since $\gamma_{M+1,1} > \gamma_{M+2,1} = \cdots = \gamma_{L1}$. As a result, the condition (35) will no longer be satisfied at the next iteration. This difficulty can be overcome as follows. Let ρ_1 denote the arithmetic mean of γ_{i1} for $i = M+1, \ldots, L$:

$$\rho_1 = \frac{1}{L-M} \sum_{i=M+1}^{L} \gamma_{i1} = \frac{y_{M+1}^2}{L-M} - \rho_0. \tag{41}$$

Instead of updating the $\gamma_i(k-1)$ individually for $i = M+1, \ldots, L$, we update the noise variance estimate $\rho(k-1)$ using the first-order expansion

$$\rho(k) = \rho_0 + \varepsilon \rho_1. \tag{42}$$

A similar averaging technique is used in [9] to maintain a multiplicity constraint on the smallest eigenvalue.

Finally, it can be verified that (23)–(24) are satisfied when the coefficients b_{ji} (25) are chosen as follows:

$$b_{ji} = -b_{ij}^* = \begin{cases} y_i^* y_j / (\gamma_{i0} - \gamma_{j0}), & i, j \in \{1, \ldots, M+1\} \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

The first-order coefficient vectors u_{i1} are obtained upon substitution of (43) in (26). In particular, we note that $u_{i1} = 0$ for $i = M+2, \ldots, L$ while for $i = 1, \ldots, M+1$, the summation in (26) extends only from $j = 1$ to $M+1$. It follows that the unperturbed noise-subspace eigenvectors $\{u_{i0}: i = M+2, \ldots, L\}$ do not enter into the calculation of the u_{i1}. Moreover, since $u_{M+1,0}$ (38) can be calculated from the data vector $x(k)$ and the unperturbed signal-subspace eigenvectors $\{u_{i0}: i \in \Omega_1\}$, knowledge of these eigenvectors alone is sufficient. Hence, in applications where only the signal-subspace

TABLE II
ADAPTIVE EIGENDECOMPOSITION ALGORITHM PB

$x(k) \leftarrow \sqrt{\varepsilon}\, x(k)$

for $i = 1$ to M:
  $\eta = u_i^H(k-1)\, x(k)$
  $y_i = |\eta|$
  $u_i(k-1) \leftarrow (\eta / y_i)\, u_i(k-1)$
end

$x_n = x(k) - \sum_{i=1}^{M} y_i\, u_i(k-1)$
$y_{M+1} = \|x_n\|$
$u_{M+1}(k-1) = x_n / y_{M+1}$

for $i = 1$ to $M+1$:
  $b_{ii} = 0$
  for $j = i+1$ to $M+1$:
    $b_{ji} = y_i y_j / \max(\delta\gamma_1(k-1), \gamma_i(k-1) - \gamma_j(k-1))$
    $b_{ij} = -b_{ji}$
  end
end

for $i = 1$ to M:
  $\gamma_i(k) = (1-\varepsilon)\gamma_i(k-1) + y_i^2$
  $u_i(k) = u_i(k-1) + \sum_{j=1}^{M+1} b_{ji} u_j(k-1)$
  $u_i(k) \leftarrow u_i(k) / \|u_i(k)\|$
end

$\gamma_{M+1}(k) \equiv \rho(k) = (1-\varepsilon)\rho(k-1) + y_{M+1}^2/(L-M)$

eigenvectors are needed, it is not necessary to maintain and update the noise-subspace eigenvectors $\{u_i(k): i \in \Omega_2\}$. More generally, if a projector on the noise-subspace is needed, it can be obtained easily from the signal-subspace eigenvectors [10].

Using the proposed first-order perturbation approach and proceeding as in Part A, an adaptive eigendecomposition algorithm can now be formulated. The stabilized version of this algorithm, denoted PB, is summarized in Table II. According to the discussion in the previous paragraph, only the signal subspace is updated; the dominant noise-subspace eigenvector $u_{M+1}(k-1)$ is calculated explicitly within the algorithm. An alternative version of this algorithm can be obtained where all appearances of $u_{M+1}(k-1)$ are eliminated through (38) and (37) [15].

C. Approximation Resulting from Large Eigenvalue Ratios

In this subsection, considerable simplifications are made to the algorithm PB by assuming that the ratios of consecutive eigenvalues, i.e. $\gamma_{i0}/\gamma_{i+1,0}$ for $i = 1, \ldots, M$, are much larger than one, or equivalently:

$$\gamma_{10} \gg \cdots \gg \gamma_{M+1,0} = \cdots = \gamma_{L0}. \tag{44}$$

In the single source case ($M = 1$), (44) corresponds to the condition of high signal-to-noise ratio (SNR) found in several applications. Using the decomposition (4) of $R_x(k)$, practical examples of multiple source scenarios where (44) is satisfied can also be found (e.g., uncorrelated, orthogonal sources with very unequal power and high SNR). However, as indicated by simulations (see Section IV), the simplified version of the algorithm PB derived in this subsection is quite robust to the assumption (44) and can be used even when some of the signal-subspace eigenvalues are repeated. In this respect, (44) should be viewed as a mathematical device for the simplification of perturbation series, rather than as a necessary condition for the application of the resulting algorithm.

Under the assumption (44), the coefficients b_{ji} (43), for $i, j \in \{1, \ldots, M+1\}$, can be approximated as follows:

$$b_{ji} \approx y_i^* y_j / \gamma_{i0}, \quad i < j \tag{45}$$

with $b_{ji} = -b_{ij}^*$ for $i > j$ and $b_{ii} = 0$. Important computational simplifications result when the approximation (45) is incorporated in the algorithm PB. Indeed, the calculation of $u_i(k)$ for $i = 1, \ldots, M$ in the algorithm PB of Table II requires on the order of $O(LM^2)$ complex operations. However, when (45) is used properly, it is possible to compute the $u_i(k)$ in only $O(LM)$ complex operations. Explanations for this are provided below. (The issue of computational complexity is considered in more detail in subsection E.)

Define the square matrix

$$B = [b_{ji}], \quad i, j = 1, \ldots, M+1. \tag{46}$$

Using (45), B can be decomposed in the form

$$B = Y\, P\, Z^H - Z\, P^T\, Y^H \tag{47}$$

where

$$Y = \operatorname{diag}(y_1, \ldots, y_{M+1}) \tag{48}$$

$$Z = \operatorname{diag}(z_1, \ldots, z_{M+1}), \quad z_i = y_i/\gamma_{i0} \tag{49}$$

$$P = [p_{ji}], p_{ji} = \begin{cases} 1, & i < j \\ 0, & i \geq j. \end{cases} \tag{50}$$

In (48) and (49), diag(.) denotes a diagonal matrix with diagonal elements given by the arguments. Now, let

$$U_0 = [u_{10}, \ldots, u_{M+1,0}] \tag{51}$$

$$U_1 = [u_{11}, \ldots, u_{M+1,1}]. \tag{52}$$

Expressing (26) in matrix notation and using (47), we have

$$U_1 = U_0 B = VZ^H - WY^H \tag{53}$$

where

$$V = [v_1, \ldots, v_{M+1}] = U_0\, Y\, P \tag{54}$$

$$W = [w_1, \ldots, w_{M+1}] = U_0\, Z\, P^T. \tag{55}$$

Simplifications occur in the computation of U_1 (53) because of the particular structure of the matrix P (50). Indeed, the matrix product $V = U_0 YP$ in (54) can be evaluated in only $O(LM)$ operations through the following recursion:

$$v_i = v_{i-1} - y_i u_{i0}, \quad i = 1, \ldots, M+1. \tag{56}$$

TABLE III
ADAPTIVE EIGENDECOMPOSITION ALGORITHM PC

$x(k) \leftarrow \sqrt{\varepsilon}\, x(k)$

for $i = 1$ to M:
  $\eta = u_i^H(k-1)\, x(k)$
  $y_i = |\eta|$
  $z_i = y_i/\gamma_i(k-1)$
  $u_i(k-1) \leftarrow (\eta/y_i)\, u_i(k-1)$
end

for $i = 1$ to M:
  $\gamma_i(k) = (1-\varepsilon)\gamma_i(k-1) + y_i^2$
end

$v = x(k)$
$w = 0_{L\times 1}$
for $i = 1$ to M:
  $v \leftarrow v - y_i u_i(k-1)$
  $u_i(k) = u_i(k-1) + z_i v - y_i w$
  $u_i(k) \leftarrow u_i(k)/\|u_i(k)\|$
  $w \leftarrow w + z_i u_i(k-1)$
end

$y_{M+1} = \|v\|$
$\gamma_{M+1}(k) \equiv \rho(k) = (1-\varepsilon)\rho(k-1) + y_{M+1}^2/(L-M)$

The initial condition is given by

$$\begin{aligned} v_0 &= \sum_{i=1}^{M+1} y_i u_{i0} \\ &= x_s + x_n \\ &= x(k) \end{aligned} \tag{57}$$

where we have used (37)–(39). We also note that

$$v_M = x_n. \tag{57}$$

Similarly, the matrix W (55) can be evaluated recursively as follows:

$$w_{i+1} = w_i + z_i u_{i0}, \quad i = 1, \ldots, M \tag{59}$$

with initial condition

$$w_1 = 0_{L\times 1}. \tag{60}$$

Finally, according to (53), the first-order coefficient vectors u_{i1} are given by

$$u_{i1} = z_i^* v_{i1} - y_i^* w_{i1}, \quad i = 1, \ldots, M+1. \tag{61}$$

As indicated in part B, $u_{i1} = 0$ for $i = M+2, \ldots, L$.

When the above computational technique is incorporated in the algorithm PB, the signal-subspace eigenvectors $u_i(k), i = 1, \ldots, M$, can be updated in only $O(LM)$ complex operations. This represents a major improvement over the original algorithm PB in Table II. The resulting algorithm, denoted PC, is summarized in Table III.

TABLE IV
OPERATION COUNTS OF ALGORITHMS PA, PB, AND PC

Algorithm	Operation count
PA	$(1/2)L^3 + O(L^2)$
PB	$(1/2)LM^2 + O(LM)$
PC	$5LM + O(L)$

D. Initialization

Startup values of $\gamma_i(0)$ and $u_i(0)$ for $i = 1, \ldots, L$ are needed to initialize the algorithm PA in Table I. These can be obtained by performing a single eigendecomposition of an initial estimate $\hat{R}_x(0)$ of the data covariance matrix. A simpler, but less effective approach consists of making a particular choice of $\hat{R}_x(0)$, for which the initial eigendecomposition step can be avoided. An example of this is provided by $\hat{R}_x(0) = \mathrm{diag}(\gamma_{10}, \ldots, \gamma_{L0})$ where $\gamma_{10} > \cdots > \gamma_{L0}$ are reasonable *a priori* guesses of the true eigenvalues, obtained from signal and noise power considerations.

For the initialization of the algorithms PB and PC in Tables II and III, we must fix the value of M (signal-subspace dimensionality) and chose startup values of $\gamma_i(0), i = 1, \ldots, M+1$, and $u_i(0), i = 1, \ldots, M$. If M is *a priori* unknown, it must be estimated from the data. Typically [14], this involves evaluating the eigenvalues of an initial estimate $\hat{R}_x(0)$ of the data covariance matrix. Once the eigenvalues are known, a criterion such as AIC or MDL is used to estimate M. In this respect, we note that the performance of the algorithms PB and PC is not significantly affected by overestimating M, owing to the stabilization mechanisms described earlier. Once M is fixed, startup values of γ_{i0} and u_{i0} can be obtained in the same manner as for the algorithm PA. A better startup value for $\gamma_{M+1,0} = \rho_0$ is actually given by the arithmetic mean of γ_{i0} for $i = M+1, \ldots, L$.

E. Computational Complexity

We use the term operation to mean one complex multiplication and one complex addition, which is about the same as four real multiplications and four real additions. The operation counts of the algorithms PA, PB, and PC are given in Table IV.

In terms of operation count, the algorithm PB is comparable to, or better than other existing algorithms with similar performance. Indeed, referring to the two families of algorithms mentioned in the introduction, the best algorithms in the first family require $O(L^2M)$ complex operations per iterations to achieve a level of performance comparable to direct eigendecomposition of the updated sample covariance matrix (10). Algorithms in the second family require $(1/2)LM^2 + O(LM)$ operations for an exact rank-one update of the covariance matrix, but they involve the solution of a nonlinear eigenvalue equation by means of interpolation techniques. In this respect, perturbation algorithms are simpler to implement since they completely avoid this step.

The operation count of the algorithm PC is one order of magnitude smaller. This is comparable to the fastest algorithms in the first family, which are based on instantaneous estimation of the data covariance matrix and as a result cannot achieve a

level of performance comparable to direct eigendecomposition of (10). One remarkable feature of the algorithm PC is the fact that, despite its low operation count, it can achieve such a level of performance. This is shown by simulations in Section IV.

IV. Simulations

The convergence properties of the new algorithms presented in this paper were studied by means of computer simulations. Other existing algorithms were also considered for the purpose of comparison. Both stationary and nonstationary computer-generated narrow-band array data were used to evaluate the algorithms. The case of stationary data is considered first, followed by the nonstationary one.

The scenario considered for the stationary simulations consisted of $M = 2$ uncorrelated narrow-band plane waves with directions of arrival (DOA) $\theta_1 = 9°$ and $\theta_2 = 12°$, impinging on an uniform linear array of $L = 8$ sensors, with intersensor spacing equals to half the wavelength. The sensor outputs were corrupted by additive circular Gaussian white noise with signal-to-noise ratios set to $\mathrm{SNR}_1 = \mathrm{SNR}_2 = 20$ dB.

Synthetic data were generated and processed with the following algorithms: perturbation algorithms PB and PC; Yang and Kaveh's covariance matrix gradient and instantaneous LMS-type adaptive algorithms [5], referred to as YK1 and YK2, respectively[2]; and exact eigendecomposition of the updated covariance matrix estimate (11), simply referred to as EIG. Except for possible roundoff errors (which do not play a significant role in this study), the EIG approach produces the same output as the rank-one adaptive eigendecomposition algorithms of the second family, which are based on the algorithm of Bunch *et al.* [12]. The use of the algorithm PA is not recommended here because of the relatively large multiplicity of the noise subspace eigenvalue, which is repeated $L-M = 6$ times.

The algorithms were initialized as follows: A sequence of 10 independent data vectors $x(k)$ was combined as in (9) to form an initial covariance matrix estimate $\hat{R}_x(0)$, which served as startup value in the EIG approach. Then, the eigenvalues and eigenvectors of $\hat{R}_x(0)$ were calculated and used as startup values in the algorithms PB, PC, YK1, and YK2, as described in Section III-D. Unless otherwise indicated, the correct value of $M = 2$ was used in these algorithms.

Three different indicators were used to evaluate the performance of the various algorithms. The first one is a vector with components

$$e_i(k) = |\gamma_i(k) - \lambda_i|/\lambda_i, \quad i = 1, \ldots, M+1 \tag{62}$$

where λ_i are the eigenvalues of the true covariance matrix (4). Hence, $e_i(k)$ represents the relative error in the ith eigenvalue estimate, $\gamma_i(k)$, at time k. The second indicator is given by

$$J(k) = \|U_s(k)U_s^H(k) - Q_s(k)Q_s^H(k)\| \tag{63}$$

where $\|.\|$ denotes the Frobenius norm of a matrix, $Q_s(k)$ is given by (8), and

$$U_s(k) = [u_1(k), \ldots, u_M(k)]. \tag{64}$$

[2]In the notation of [5], the equivalence is YK1 = ∇_1 = A2 and YK2 = ∇_2 = A3.

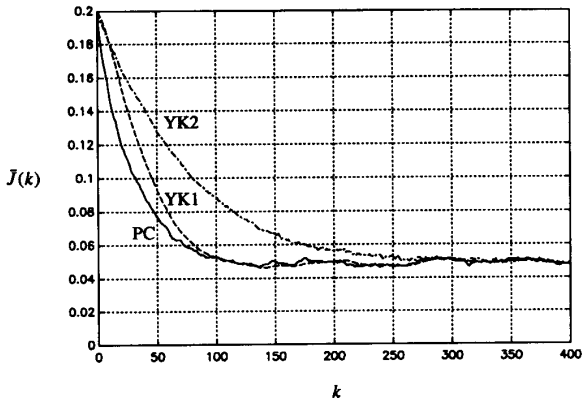

Fig. 1. Signal-subspace error $\bar{J}(k)$ as a function of k for the algorithms PC, YK1, and YK2 under the same steady-state error ($M = 2$ sources; $\theta_1 = 9°$, $\theta_2 = 12°$; $\mathrm{SNR}_1 = \mathrm{SNR}_2 = 20$ dB; 40 independent runs).

The quantity $J(k)$ provides a measure of the error between the projector on the true signal subspace, i.e., $Q_s(k)Q_s(k)^H$, and an estimate of this projector at time k given by $U_s(k)U_s^H(k)$. When the columns of $U_s(k)$ are exactly orthonormal, $J(k)$ is equal to the distance between the subspace spanned by the columns of $U_s(k)$ and the signal-subspace spanned by $Q_s(k)$ [17]. The third indicator is a vector of DOA estimates, $[\hat{\theta}_1(k), \hat{\theta}_2(k)]$, obtained with the root-MUSIC algorithm [16]. In the following discussion, $\bar{e}_i$, $\bar{J}_i$ and $\bar{\theta}_i(k)$ represent the averaged values of these indicators over 40 independent runs (with independent re-initializations).

Fig. 1 shows the "learning curves" of $\bar{J}(k)$ as a function of k for the algorithms PC, YK1, and YK2, under the same steady-state error. To obtain these curves, the parameters controlling the algorithms YK1 and YK2 were first set as follows: $\alpha = 0.075$, $\mu_1 = 0.00024$, and $\mu_2 = 0.00012$, where μ_1 and μ_2 are the corresponding convergence factors (such values are typical of those used in [5]). To obtain the same steady-state error, the perturbation parameter of the algorithms PB and PC was then set to $\varepsilon = 0.02$. As can be seen from Fig. 1, the new algorithm converges more rapidly than the gradient-based algorithms. Fig. 2 shows the corresponding DOA estimates for the algorithms PC and YK1. It is seen that the DOA's obtained with the algorithm PC are relatively unbiased, while those obtained with the algorithm YK1 are slightly biased. The DOA's of the algorithm YK2, not shown, are noisier and more biased than those of the algorithm YK1. Fig. 3 shows the eigenvalue errors $\bar{e}_i(k)$ $(i = 1, 2, 3)$ as a function of k for the algorithm PC. The lower steady-state value of $\bar{e}_3(k)$ (by a factor of about $1/\sqrt{6}$) is due to the averaging operation in (41).

The corresponding curves for the algorithms PB and EIG, not shown in Figs. 1–3 for the sake of clarity, are almost undistinguishable from those of the algorithm PC. Two important conclusions can be drawn from this observation. First, despite a much lower computational complexity (i.e., $O(LM)$ as compared to $O(LM^2)$) resulting from simplifying assumptions, the algorithm PC can achieve the same level of performance as the algorithm PB. Secondly, although perturbation algorithms perform only an approximate update

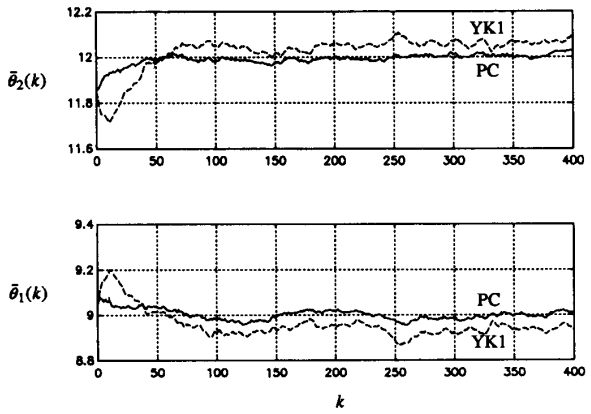

Fig. 2. DOA estimates $\bar{\theta}_1(k)$ and $\bar{\theta}_2(k)$ as a function of k for the algorithms PC and YK1 (same conditions as in Fig. 1).

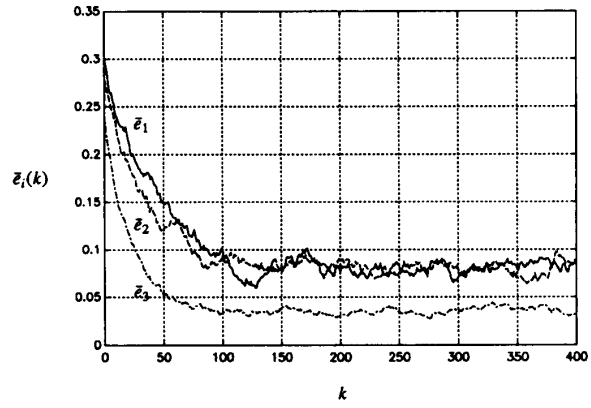

Fig. 3. Eigenvalue errors $\bar{e}_i(k)(i = 1, 2, 3)$ as a function of k for the algorithm PC (same conditions as in Fig. 1).

of the eigendecomposition of the covariance matrix estimate $\hat{R}_x(k)$, their performance is comparable to an exact, more costly approach in which the eigendecomposition of $\hat{R}_x(k)$ is updated exactly. This suggests that the extra accuracy gained from an exact eigendecomposition is overshadowed by statistical errors in the estimate $\hat{R}_x(k)$ itself.

The effects of overestimating the signal-subspace dimensionality M were also investigated. The results indicate that both perturbation and gradient-based algorithms are not sensitive to small errors in M. In particular, when the above experiment was repeated with an estimated value of $M = 3$, no significant performance degradation was observed with the algorithms PB, PC, YK1, and YK2.

To demonstrate the effectiveness of the stabilization technique described in Section III-A and to evaluate the robustness of the algorithm PC to the assumption (44), the algorithms were tested under the severe condition of equal signal-subspace eigenvalues. To this end, the same scenario as above was considered, but the DOA's were set to $\theta_1 = 0°$ and $\theta_2 = \arcsin(1/4) = 14.4775°$. With this choice, $\lambda_1 = \lambda_2 = 801$ and $\lambda_i = 1, i = 3, \ldots, 8$. The DOA estimates of the algorithms PB and PC are shown in Fig. 4. In terms of bias and

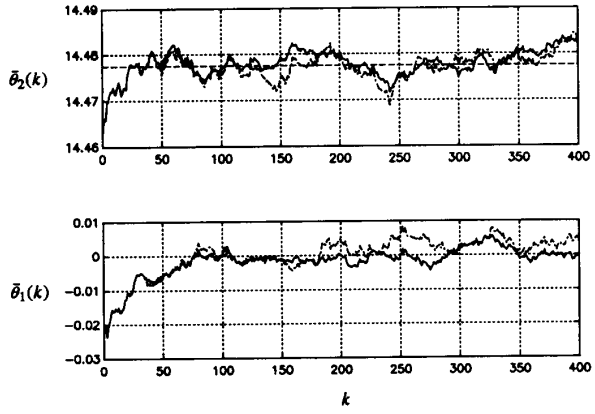

Fig. 4. DOA estimates $\bar{\theta}_1(k)$ and $\bar{\theta}_2(k)$ versus k for the algorithms PB (—) and PC (–·) ($M = 2$ sources; $\theta_1 = 0°$, $\theta_2 = 14.4775°$; $\mathrm{SNR}_1 = \mathrm{SNR}_2 = 20$ dB; 40 independent runs).

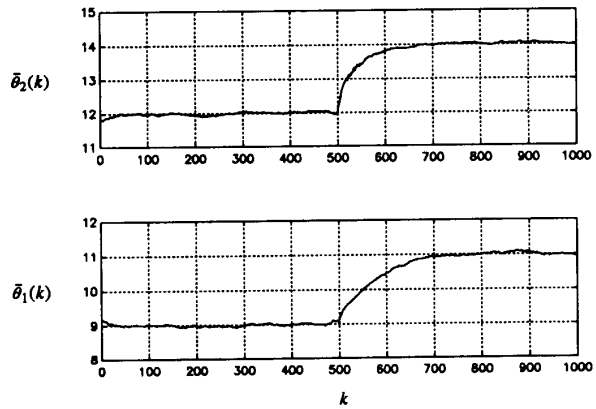

Fig. 5. DOA estimates $\bar{\theta}_1(k)$ and $\bar{\theta}_2(k)$versus k for the algorithm PC when submitted to a sudden environmental change ($M = 2$; $[\theta_1, \theta_2] = [9°, 12°]$ for $0 \leq k < 500$,$[\theta_1, \theta_2] = [11°, 14°]$ for $500 \leq k < 1000$; $\mathrm{SNR}_1 = \mathrm{SNR}_2 = 20$ dB; 10 independent runs).

variance, these estimates are comparable to those of the EIG approach. In general, the simulation results indicate that the performance of the algorithms PB and PC is not significantly affected by the condition $\lambda_1 = \lambda_2$.

The algorithms were also tested on nonstationary data corresponding to two different scenarios. The purpose of the first scenario was to evaluate the ability of the algorithms to recover from a sudden change in the environment. Initially, the same simulation environment as in Fig. 1–3 was used to generate data for $0 \leq k < 500$ (i.e., two stationary sources with $\theta_1 = 9°$ and $\theta_2 = 12°$ and SNR $= 20$ dB). Then, at time $k = 500$, the DOA's were changed abruptly to $\theta_1 = 11°$ and $\theta_2 = 14°$. Fig. 5 shows the corresponding DOA estimates obtained with the algorithm PC (10 independent runs). It is seen that the algorithm is able to fully recover from such a sudden change in the environment. Similar results were obtained with the other algorithms.

With the second scenario, we tested the ability of the algorithms to track two crossing sources. To this end, one

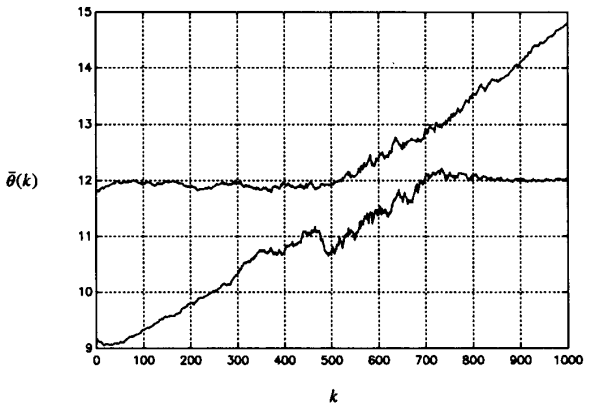

Fig. 6. DOA estimates of the algorithm PC for two crossing sources ($\theta_1 = 9° + 6°k/1000$, $\theta_2 = 12°$; $\mathrm{SNR}_1 = \mathrm{SNR}_2 = 20$ dB; 10 independent runs).

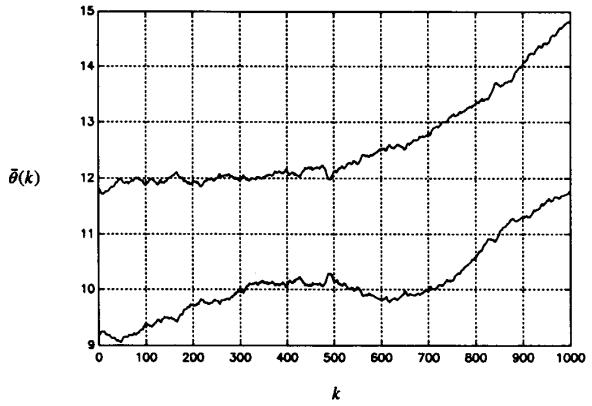

Fig. 7. DOA estimates of the algorithm YK1 for two crossing sources ($\theta_1 = 9° + 6°k/1000$, $\theta_2 = 12°$and $\mathrm{SNR}_1 = \mathrm{SNR}_2 = 20$ dB; 10 independent runs).

of the source was maintained fixed at $\theta_2 = 12°$, while the other was changing DOA at a rate of $0.006°$ per iteration, according to $\theta_1(k) = 9° + 6°k/1000$. As before, the SNR was set to 20 dB. Fig. 6 shows the corresponding DOA estimates of the algorithm PC. (The results could be further improved by the application of smoothing and track association algorithms.) Except for a small time interval around the crossing point at $k = 500$, where the root MUSIC algorithm is unable to correctly resolve the two sources, the algorithm shows good tracking performance. The algorithms PB and EIG have similar behavior. However, as shown in Fig. 7, the tracking performance of YK1 is not as good.

V. Summary and Discussion

New algorithms for adaptive eigendecomposition of time-varying data covariance matrices were presented. The algorithms are based on a first-order perturbation analysis of the rank-one update for covariance matrix estimates with exponential windows. Different assumptions on the eigenvalue structure led to three distinct algorithms with varying degrees of complexity. A technique for stabilizing the algorithms was presented and the issues of initialization and computational complexity were discussed briefly. Finally, the convergence behavior of the new algorithms was evaluated via computer simulations and mathematical analysis.

The results show that the new algorithms can achieve the same level of performance as a direct approach in which the exact eigendecomposition of the sample covariance matrix is obtained at each iteration. Existing algorithms with similar performances require on the order of $O(LM^2)$ complex operations per iteration and involve either a full Gram-Schmidt orthogonalization of the adaptive eigenvectors or a nonlinear eigenvalue search. The new algorithms have comparable or lower operation counts and do not involve any of these steps. One particular perturbation algorithm, i.e., PC, can be used to update the complete signal-subspace eigenstructure in $5LM$ complex operations while maintaining the same level of performance as more complex $O(LM^2)$ algorithms. Moreover, a simplified local convergence analysis of algorithm PA shows that it converges geometrically to the true eigendecomposition and is characterized by a single time constant.

Considering their numerical complexity (operation count, parallelism, etc.) and their convergence behavior, the new perturbation-based algorithms are particularly well suited to real-time implementations of various signal-subspace processing methods.

Appendix I

In this Appendix, we present a simplified convergence analysis of the algorithm PA. Only the local behavior of the mean eigenvalues and eigenvectors around the true eigendecomposition is considered. Nevertheless, the results obtained are quite pleasing and provide additional motivations for using the new algorithms. Another virtue of this simplified analysis is to help the reader develop a better understanding of the internal operation of the new algorithms and of the conditions necessary for their successful application.

For the present discussion, we consider the following form of the algorithm PA, where the final normalization step has been omitted:

$$\gamma_i(k) = (1-\varepsilon)\gamma_i(k-1) + \varepsilon u_i^H(k-1)x(k)x^H(k)u_i(k-1), \tag{A1}$$

$$u_i(k) = u_i(k-1) + \varepsilon \sum_{j\neq i} \frac{u_j^H(k-1)x(k)x^H(k)u_i(k-1)}{\gamma_i(k-1)-\gamma_j(k-1)} u_j(k-1). \tag{A2}$$

Furthermore, the data vector $x(k)$ is assumed to be stationary with $E[x(k)] = 0$ and $E[x(k)x^H(l)] = R_x\delta_{kl}$. The eigenvalues and normalized eigenvectors of the time-independent covariance matrix R_x are denoted by λ_i and q_i, respectively.

We first note that $\gamma_i(k) = \lambda_i$ and $u_i(k) = q_i$ for $i = 1, \ldots, L$ is a stochastic stationary solution of (A1)–(A2) in the sense that conditioned on $\gamma_i(k-1) = \lambda_i$ and $u_i(k-1) = q_i$, these equations imply $E[\gamma_i(k)] = \lambda_i$ and $E[u_i(k)] = q_i$.

To study the local behavior of $\gamma_i(k)$ and $u_i(k)$ around this stationary solution, the following assumptions are made:

(1) We pose

$$u_i(k) = q_i + n_i(k) \tag{A3}$$

where $n_i(k)$ represents small time-varying fluctuations around the true eigenvector q_i.

(2) Second- and higher-order terms in $n_i(k)$ are neglected in the analysis.

(3) Let $Q = [q_1, \ldots, q_L]$ and $U(k) = [u_1(k), \ldots, u_L(k)]$. Since the eigenvectors q_i are orthonormal, it follows that

$$QQ^H = I. \tag{A4}$$

Similarly, we shall assume that

$$E[U(0)]E[U^H(0)] = I. \tag{A5}$$

(4) From (A4) and (A5), it follows that

$$E[U(0)] = QR \tag{A6}$$

where $R = Q^H E[U(0)]$ is also unitary, i.e.

$$RR^H = I. \tag{A7}$$

Hence, according to (A6), the orthonormal coordinate systems defined by $E[U(0)]$ and Q are related by a unitary transformation (i.e., generalized rotation) in the complex L-dimensional vector space. Since we are concerned with the local behavior of $U(k)$ near the stationary point Q, we shall assume that the generalized rotation R is small, i.e.

$$R = I + S \tag{A8}$$

where $\|S\| \ll 1$. Substituting (A8) in (A7) and neglecting the second order term SS^H, it follows that

$$S + S^H = 0 \tag{A9}$$

so that S is anti-hermitian[3].

(5) The reader will note that the evolution equations (A1) and (A2) for $\gamma_i(k)$ and $u_i(k)$ are coupled. To simplify the discussion, we consider an uncoupled approximation to (A1) and (A2) obtained by setting $u_i(k-1) = q_i$ in (A1) and $\gamma_i(k-1) = \lambda_i$ in (A2). (Note that this assumption is not absolutely essential.)

Using the above assumptions, one can derive linear evolution equations for $E[\gamma_i(k)]$ and $E[u_i(k)]$. From (A1), we obtain

$$E[\gamma_i(k) - \lambda_i] = (1-\varepsilon)E[\gamma_i(k-1) - \lambda_i] \tag{A10}$$

whose solution is

$$E[\gamma_i(k) - \lambda_i] = (1-\varepsilon)^k E[\gamma_i(0) - \lambda_i]. \tag{A11}$$

[3]The fact that a small rotation can be expressed in the form $R = I + S$ with $S = -S^H$ is well known from the theory of continuous transformation groups [18]. In this framework, $R = I + S$ is identified as an infinitesimal generator of the group of unitary matrices.

From (A2), we obtain

$$E[u_i(k)] = E[u_i(k-1)] + \varepsilon \sum_{l \neq i} \frac{\lambda_i E[u_l^H(k-1)]q_i + \lambda_l q_l^H E[u_i(k-1)]}{\lambda_i - \lambda_l} q_l. \tag{A12}$$

To solve this linear recursion, we first define

$$\beta_{ji}(k) = q_j^H E[u_i(k)], \quad i,j = 1, \ldots, L. \tag{A13}$$

In terms of these new variables, (A12) can be expressed as

$$\beta_{ii}(k) = \beta_{ii}(k-1) \tag{A14}$$

$$\beta_{ji}(k) = \beta_{ji}(k-1) + \frac{\varepsilon}{\lambda_i - \lambda_j}[\lambda_i \beta_{ij}^*(k-1) + \lambda_i \beta_{ji}(k-1)], \quad j \neq i. \tag{A15}$$

The above recursion for the parameters $\beta_{ji}(k)$, $j \neq i$, can be uncoupled easily by introducing new variables as follows:

$$v_{ij}(k) = \beta_{ji}(k) + \beta_{ij}^*(k), \tag{A16}$$

$$w_{ij}(k) = \lambda_j \beta_{ji}(k) + \lambda_i \beta_{ij}^*(k). \tag{A17}$$

Using (A16), (A17), and (A15), we obtain:

$$v_{ij}(k) = v_{ij}(k-1) \tag{A18}$$

$$w_{ij}(k) = (1-\varepsilon)w_{ij}(k-1). \tag{A19}$$

The solution of these uncoupled equations is straightforward, namely

$$v_{ij}(k) = v_{ij}(0) \tag{A20}$$

$$w_{ij}(k) = (1-\varepsilon)^k w_{ij}(0). \tag{A21}$$

An important simplification occurs at this point as a result of the assumption that $E[U(0)]$ and Q are related by a small rotation. Indeed, using (A6) and (A8) in (A13), it can be verified that

$$\beta_{ii}(0) = 1 \tag{A22}$$

$$\beta_{ji}(0) = s_{ji}, \quad j \neq i \tag{A23}$$

where s_{ji} denotes the corresponding element of the matrix S. As a consequence of (A9), it follows that

$$v_{ij}(0) = 0 \tag{A24}$$

$$w_{ij}(0) = (\lambda_j - \lambda_i)s_{ji}. \tag{A25}$$

Finally, substituting (A24)–(A25) in (A20)–(A21), inverting the transformation (A16)–(A17) to obtain $\beta_{ji}(k)$ in terms of $v_{ij}(k)$ and $w_{ij}(k)$, and expressing $E[u_i(k)]$ in terms of the $\beta_{ji}(k)$, one obtains the desired solution of the evolution equation (A12), namely

$$E[U(k)] = Q[I + (1-\varepsilon)^k S]. \tag{A26}$$

For $0 < \varepsilon < 1$, which is the situation of interest in this paper, the following conclusions can be drawn from the above local convergence analysis:

1) From (A11), it follows that $\gamma_i(k)$ converges in the mean to the true eigenvalue λ_i.

2) From (A26), it follows that $u_i(k)$ converges in the mean to the true eigenvector q_i.
3) According to (A26), the coordinate systems defined by $E[U(k)]$ and Q are related by a small rotation whose magnitude decay with time so that the two systems eventually get correctly aligned.
4) The convergence is geometrical, with all adaptive modes characterized by a single time constant which is given by $1/\varepsilon$ for ε small. This is a rather remarkable property of this algorithm.
5) Contrarily to other adaptive algorithms such as those based on stochastic gradient search, the time constant of algorithm PA is independent of the true eigenvalues.

The above analysis can be extended to the algorithms PB and PC. Moreover, it is possible to use the same type of approach to study the sensitivity of the new algorithms to the choice of initial conditions. This would involve determining all the stochastic stationary points of a particular algorithm and studying the local behavior of the algorithm around these points. Depending on whether these stationary points attract or repel the estimated eigendecomposition, it would be possible to formulate general rules concerning the choice of initial conditions. Such an analysis is currently under way and will be reported independently along with a more comprehensive convergence analysis of the new algorithms including a study of steady-state variance.

ACKNOWLEDGMENT

The author would like to thank Dr. J. Krolik for an interesting discussion on perturbation theory leading to this work.

REFERENCES

[1] *Modern Spectrum Analysis,* Vol II, S. B. Kesler, Ed. New York: IEEE Press, 1986.
[2] *Advances in Spectrum Analysis and Array Processing,* Vol. 2, S. Haykin, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[3] N. L. Owsley, "Adaptive data orthogonalization," *IEEE Int. Conf. ASSP* (Tulsa), 1978, pp. 109–112.
[4] P. Thompson, "An adaptive spectral analysis technique for unbiased frequency estimation in the presence of white noise," *Proc. 13th Asilomar Conf. Circuits Syst.* (Pacific Grove, CA), Nov. 1980, p. 529.
[5] J. -F. Yang and M. Kaveh, "Adaptive eigensubspace algorithms for direction or frequency estimation and tracking," *IEEE Trans. ASSP*, vol. 36, pp. 241–251, Feb. 1988.
[6] P. A. Regalia, "An adaptive unit norm filter with applications to signal analysis and Karhunen-Loeve transformations," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 646–649, May 1990.
[7] R. Schreiber, "Implementation of adaptive array algorithms," *IEEE Trans. ASSP*, vol. 34, pp. 1038–1045, Oct. 1986.
[8] R. D. DeGroat and R. A. Roberts, "Efficient, numerically stabilized rank-one eigenstructure updating," *IEEE Trans. ASSP*, vol. 38, pp. 301–316, Feb. 1990.
[9] K.-B. Yu, "Recursive updating the eigenvalue decomposition of a covariance matrix," *IEEE Trans. ASSP*, vol. 39, pp. 1136–1145, May 1991.
[10] M. D. Ortigueira and M.-A. Lagunas, "Eigendecomposition versus singular value decomposition in adaptive array signal processing," *Signal Processing*, vol. 25, pp. 35–49, Oct. 1991.
[11] G. H. Golub, "Some modified matrix eigenvalue problems," *SIAM Rev.*, vol. 15, pp. 318–334, Apr. 1973.
[12] J. R. Bunch, C. P. Nielsen and D. C. Sorensen, "Rank-one modification of the symmetric eigenproblem," *Numer. Math.*, vol. 31, pp. 31–48, 1978.
[13] F. Rellich, *Perturbation Theory of Eigenvalue Problems.* New York: Gordon and Breach, 1969.
[14] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. ASSP*, vol. 33, pp. 387–392, Apr. 1985.
[15] B. Champagne, "Adaptive signal-subspace processing based on first-order perturbation analysis," *IEEE Pacific Rim Conf. CCSP* (Victoria, Canada), May 1991, pp. 120–123.
[16] A. J. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," *IEEE Int. Conf. ASSP* (Boston), 1983, pp. 336–339.
[17] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore: Johns Hopkins Univ. Press, 1989.
[18] M. Hamermesh, *Group Theory and its Application to Physical Problems.* Reading, MA: Addison-Wesley, 1962.

Benoit Champagne (S'87–M'89) was born in Joliette, Quebec, Canada on January 13, 1961. He received the B.Ing. degree in engineering physics from the Ecole Polytechnique of Montreal in 1983, the M.Sc. degree in physics from the University of Montreal in 1985, and the Ph.D. degree in electrical engineering from the University of Toronto in 1990.

Since 1990, he has been an Assistant Professor at INRS-Télécommunications, Université du Québec. His current research interests lie in the areas of statistical signal processing, sensor array processing, adaptive filtering and communication theory.