SIMULTANEOUS PRINCIPAL COMPONENT EXTRACTION WITH APPLICATION TO ADAPTIVE BLIND MULTIUSER DETECTION

Deniz Erdogmus, Yadunandana N. Rao, Kenneth E. Hild II, Jose C. Principe

Computational NeuroEngineering Laboratory, Electrical and Computer Engineering Department,

University of Florida, NEB 454, Gainesville, FL 32611

[deniz,yadu,hildk,principe]@cnel.ufl.edu

ABSTRACT

SIPEX-G is a fast converging, robust, gradient-based PCA algorithm that has been recently proposed by the authors. Its superior performance in synthetic and real data compared with its benchmark counterparts makes it a viable alternative in applications where subspace methods are employed. Blind multiuser detection is one such area, where subspace methods, recently developed by researchers, have proven effective. In this paper, the SIPEX-G algorithm is presented in detail, convergence proofs are derived, and the performance is demonstrated in standard subspace problems. These subspace problems include direction of arrival estimation for incoming signals impinging on a linear array of sensors, non-stationary random process subspace tracking, and adaptive blind multiuser detection.

I. INTRODUCTION

Principal component analysis (PCA) is a fundamental statistical technique that has established its significance in signal processing through numerous successful applications including, but not limited to, feature extraction, signal estimation, detection, speech separation, linear discriminant analysis, direction of arrival estimation and subspace filtering [1-5]. There are many algorithms that have been proposed for solving the PCA problem both off-line and on-line; specifically, Oja's rule ignited an interest among researchers for on-line PCA algorithms [6]. Sanger's rule [7], Rubner-Tavan algorithm [8,9], and APEX [2] are immediate extensions to Oja's update rule. These conventional topologies and their associated learning algorithms have been successfully applied to many problems of signal processing, yet they have shortcomings in speed of convergence and estimation accuracy mainly because they are gradient-based approaches that depend heavily on the deflation procedure. The eigenvectors, when using procedures of this type, are prevented from converging simultaneously. In fact, by definition it obliges them to converge sequentially. Although the more advanced methods, APEX and Rubner-Tavan algorithms, achieve deflation using a lateral network of weights in the output layer, the convergence of the minor components (or principal components depending on the sign on the gradient) is far from satisfactory, especially when the dimensionality grows large. There are also well-known fast-converging rules for PCA like the Natural Power Method and the Fixed-Point Rule [5,10]. However, they still use the deflation scheme to determine the intermediate eigenvectors after the principal component has converged, which prevents the algorithms from converging simultaneously to all the components. Xu's LMSER algorithm uses subspace techniques and a diagonal amplification matrix to extract the principal components simultaneously [11]. Although LMSER introduces a great improvement over the traditional methods in terms of speed and accuracy, it does not constrain the search space for the PCA weight matrix to the set of orthonormal matrices, and therefore must search the entire space when trying to orthonormalize the estimated eigenvectors. Our SIPEX-G algorithm (simultaneous principal component extraction – gradient based approach) on the other hand, employs Givens rotations as an orthonormal parameterization of the PCA weight matrix and uses a robust and consistent estimate of the output variances (based on the input vector's covariance estimate) in order to converge quickly and accurately to the eigenvectors of the underlying covariance matrix [12,13].

There has been considerable interest and recent research in the field of multiuser detection [14]; specifically *adaptive* multiuser detection [15-22]. Within this framework, subspace methods for CDMA (Code-Division Multiple-Access) channel estimation and for multiuser detection have also been investigated [18-23]. Blind multiuser detection is useful for inter-symbol interference (ISI) suppression in CDMA uplinks as well as in downlinks, where the mobile receiver has knowledge of only its own spreading sequence. The gain in uplink is relatively small as the base stations usually have access to all the spreading sequences in their cells. Group-blind methods may, in this case, be helpful in decreasing the interference from the users with unknown spreading sequences, if there are any [19]. As in any subspace-based approach, the performance of the PCA algorithm that is employed has a considerable impact on the overall success of the solution.

In this paper, we will present a complete derivation of the SIPEX-G algorithm and provide a proof demonstrating that the combination of the proposed cost function and topology form a system in which all of the stationary points are solutions to the PCA problem. We will also demonstrate the performance of SIPEX-G on a number of signal processing applications including non-stationary subspace tracking, direction of arrival estimation, and blind multiuser detection.

II. THE COST FUNCTION, THE TOPOLOGY, AND THE SIPEX-G ALGORITHM

It is well known that the directions of the principal components are given by the eigenvectors of the covariance matrix of the input data, ordered according to their corresponding eigenvalues in descending order of magnitude [24]. Thus PCA is nothing more than a coordinate transformation on the data, where in the new coordinate system, the axes are aligned with the directions of maximal variation. This immediately points out that the search for the weights of a principal component network can be restricted to the set of orthonormal matrices, since every orthonormal transformation corresponds to an axis-rotation in the input vector space. Consider the principal component network with **y**=**Rx**, where $\mathbf{x} \in \Re^{nx1}$ and $\mathbf{y} \in \Re^{nx1}$ are the input and output vectors, respectively, and $\mathbf{R} \in D \subset \Re^{nxn}$ is the PCA weight matrix, which is restricted to the subset *D* of orthonormal matrices. The cost function in (1) could be maximized (or minimized) in order to determine the principal (or minor) components of the input data, whose covariance matrix is given by $\mathbf{\Sigma_x}$. Considering the case where (1) is maximized, the scalar gains γ_s are chosen in descending order (ascending if (1) is minimized) such that $\gamma_1>\gamma_2>...>\gamma_{n-1}>0$. Thus, the cost function is just the weighted sum of the first (n-1) output variances. In the subsequent discussions, we assume that the input data **x** is zero-mean, without loss of generality.

$$J = \sum_{s=1}^{n-1} \gamma_s Var(\mathbf{y}_s) \qquad (1)$$

The utility of this cost function, when used together with the proposed orthonormal network topology, is established by the following theorem. This theorem states that when our objective is to determine the complete set of principle components, any stationary point of this cost-topology pair yields the desired solution, although they might not be in descending or ascending order. However, as we will see in the sequel, if necessary, this ordering could be achieved at no additional computational cost, since the algorithm already provides everything necessary to determine the ordering of the eigenvectors.

Theorem 1. For the constrained PCA network where **R** is an orthonormal matrix, the function J in (1) has a stationary point if and only if all the rows of **R** consist of all the eigenvectors of $\mathbf{\Sigma_x}$.

Proof. In the Appendix.

Corollary 1. There are totally n! stationary points of J of which (n-1)! are local maxima, (n-1)! are local minima, and (n-2)(n-1)! are saddle points.

Proof. This follows easily from the ideas in the proof of *Theorem 1*. The (n-1)! local maxima correspond to all possible permutations of the (n-1) maximum eigenvectors being placed in the first (n-1) rows of **R**. Similarly, the (n-1)! local minima correspond to all possible permutations of the (n-1) minimum eigenvectors being placed in the last (n-1) rows of **R**. All other permutations of the eigenvectors in **R**, which amount to (n-2)(n-1)! are saddle points.

Theorem 1 practically states that we can adapt a rotation matrix (in batch mode or on a sample-by-sample basis) in order to obtain all the principal components of the input data at the output of this linear network. As a consequence, the rows of the PCA weight matrix **R** will give us the desired eigenvectors of the input covariance matrix. It is also possible to include in the cost function given in (1), only the variances of the first m outputs. Global maximization of this new cost function will result in the convergence of the first m rows of the rotation matrix to the first m principal components. Otherwise, if the summation in (1) does not go up to n-1, due to *Corollary 1*, the first m rows of **R** will correspond to some collection of m eigenvectors depending on which local maximum is attained. The gains, γ_i, come into play at this point. By careful assignment of the scale factors to the outputs, it is possible to force the solution towards the global maximum. This procedure, however, is beyond the scope of this paper. In the following, we will only concentrate on the case where all the eigenvectors are to be determined and their order is not important.

Now we focus on finding a suitable parameterization for the orthonormal PCA matrix. Every orthonormal matrix can be considered a rotation matrix, thus they can be parameterized in terms of Givens rotation angles, each of which define a rotation in a single (two-dimensional) plane of the high-dimensional vector space. Then, these individual rotations can be cascaded together to span the whole set of orthonormal matrices. Every rotation matrix has a unique set of Givens rotation angles that characterize it. In n-dimensions, a Givens rotation matrix in the plane formed by the p^{th} and q^{th} axes is denoted by $\mathbf{R}^{pq}$, and is given by an identity matrix whose four entries at the intersection of p^{th} and q^{th} rows with p^{th} and q^{th} columns are modified as follows: the $(p,p)^{\text{th}}$ and $(q,q)^{\text{th}}$ entries are $cos\,\theta_{pq}$, and the $(p,q)^{\text{th}}$ and $(q,p)^{\text{th}}$ entries are $-sin\,\theta_{pq}$ and $sin\,\theta_{pq}$, respectively [5]. The complete rotation matrix is then formed from these sparse matrices according to

$$\mathbf{R} = \prod_{p=1}^{n-1} \prod_{q=p+1}^{n} \mathbf{R}^{pq} \tag{2}$$

The multiplication order can be always from the left or always from the right. It is not crucial to the generality of this formula as long as the same order is maintained when taking the derivative of the matrix with respect to each rotation angle. To summarize, our objective is to solve the following unconstrained optimization problem parameterized in terms of the Givens rotation angles.

Problem. Let $\theta_{pq},\ \ p = 1,\ldots,n-1,\ \ q = p+1,\ldots,n$ be the Givens rotation angles that constitute our parameter vector $\mathbf{\Theta}$. The cost function is explicitly given by

$$J(\mathbf{\Theta}) = \sum_{s=1}^{n-1} \gamma_s Var(\mathbf{y}_s) = \sum_{s=1}^{n-1} \gamma_s \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{R}_{si} \mathbf{R}_{sj} \mathbf{\Sigma}_{\mathbf{x},ij} \tag{3}$$

where $\mathbf{R}_{sj}$ is the $(s,j)^{\text{th}}$ entry of the orthonormal matrix $\mathbf{R}$, which is constructed using the Givens rotation angles as shown in (2), and $\mathbf{\Sigma}_{\mathbf{x},ij}$ is the $(i,j)^{\text{th}}$ entry of the input covariance matrix. Since all stationary points of this cost function yield the desired eigenvectors, minimization or maximization of (3) may be employed during optimization.

The Givens angles provide a suitable representation for the PCA weight matrix in that it guarantees that the search is limited to the set of orthonormal matrices. The cost function, ideally, provides an effective means of forcing the weight matrix towards the desired eigenvectors. In practice, however, we neither have access to the actual input covariance matrix nor to a parametric expression of the output variances in terms of the Givens angles. Therefore, the proposed algorithm utilizes a robust and consistent non-parametric estimator for the cost function. To this end, the following well-known recursive sample-estimate for the input covariance matrix is employed

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}(k) = \alpha\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}(k-1) + (1-\alpha)\mathbf{x}(k)\mathbf{x}^T(k) \tag{4}$$

In this expression, the notation $\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}(k)$ is used to distinguish the sample-estimate of the input covariance matrix at time k from its true value $\boldsymbol{\Sigma}_{\mathbf{x}}$. The adjustable parameter $\alpha \in [0,1)$ is the forgetting factor and $\mathbf{x}(k)$ is the k^{th} input vector sample. Note that the recursion in (4) can be used for both wide sense stationary (WSS) and non-stationary input signals. If the input is known to be WSS, the recursive estimator in (5), which is unbiased and consistent, may be utilized.

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}(k) = \frac{k-1}{k}\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}(k-1) + \frac{1}{k}\mathbf{x}(k)\mathbf{x}^T(k) \tag{5}$$

Using this estimate for the covariance matrix for WSS data will improve the performance of the estimate, and therefore the algorithm. In both recursions, the covariance estimate $\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}(k)$ can be initialized using the first $N > n$ samples using the unbiased sample estimate (for WSS data) of the covariance matrix (recall that we assumed $\mathbf{x}$ is zero mean)

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{x}}(0) = \frac{1}{N}\sum_{l=1}^{N}\mathbf{x}(l)\mathbf{x}^T(l) \tag{6}$$

The approach presented above, which uses the input covariance matrix estimate and the current values of the PCA weight matrix to estimate the output variances, assures a robust estimate of the cost function. More importantly, the estimate of the gradient of this cost function is robust, which leads to a smooth and fast convergence to the desired solution. If, instead, the output variances are estimated directly from the output samples using either a batch of output values or only the most recent output value, as in the traditional approach to PCA estimation, we would risk losing these advantageous properties; in fact, simulations with different schemes in estimating the output variances have demonstrated that the proposed approach provides the best convergence results both in terms of convergence speed and accuracy.

Although there are a variety of optimization procedures that could be employed in order to maximize the cost function in (3), in this paper, we will use the simple gradient ascent approach. We know from many applications in signal processing and communications that gradient-based algorithms are successful in on-line adaptation schemes because of their ability to handle non-stationary signals as well as time-varying models. Below, we briefly describe the SIPEX-G algorithm, which uses gradient ascent optimization.

Algorithm. Simultaneous principal component extraction using the gradient approach (SIPEX-G)

1. Initialize Givens angles (randomly or to all zeros so that the initial rotation matrix is the identity matrix).
2. Initialize the estimate of the input covariance matrix using (6).
3. For non-WSS signals and/or time-varying environments, update the covariance estimate using (4). If the input is WSS and the environment is time-invariant use (5).
4. Calculate the gradient of the cost function in (3) with respect to the Givens angles (substituting the input covariance with its most current sample-estimate). The gradient is

$$\frac{\partial J}{\partial \theta_{pq}} = \sum_{s=1}^{n-1} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\mathbf{R}_{si} \frac{\partial \mathbf{R}_{sj}}{\partial \theta_{pq}} + \frac{\partial \mathbf{R}_{si}}{\partial \theta_{pq}} \mathbf{R}_{sj} \right) \hat{\mathbf{\Sigma}}_{\mathbf{x},ij} \tag{7}$$

5. Update the Givens angles using gradient ascent/descent.

$$\mathbf{\Theta}(k+1) = \mathbf{\Theta}(k) + \eta \frac{\partial J}{\partial \mathbf{\Theta}} \tag{8}$$

   where η is an adjustable step size.

6. Go back to step 3 and continue as long as input samples arrive.

A key concern in many adaptive algorithms is the computational complexity. It is clear that if the multiplications in (2) are performed from the left, the first output is only affected from the Givens angles with indices θ_{1q}, q=2,…,n, the second is affected by all the angles θ_{1q}, q=1,...,n and θ_{2q}, q=3,...,n, and so on. Thus, if we wish to extract the first m principal components, we only need to adapt the angles θ_{ij}, i=1,…,m, j=i+1,…,n, which makes a total of mn-m(m+1)/2 parameters, which is less than the mn parameters required in many PCA algorithms. The trade-off is that the *sin* or *cos* of all these parameters must be evaluated, which increases computational complexity. This could be somewhat alleviated, however, by means of a look-up table. In addition, the necessary matrix and vector multiplications in the algorithm must be performed at each iteration, which amount to $O(mn^2)$ operations. Specifically, the gradient in (7) has a complexity of $O(n^3)$, which is more than the minimum computational complexity that a fully stochastic PCA algorithm can achieve, which is $O(n^2)$. Although SIPEX-G has a higher complexity, it has superior convergence properties due to the fact that the search is restricted to the set of orthonormal matrices only; this will also guarantee simultaneous convergence of all principal components.

Another crucial issue in all adaptive learning algorithms is stability. In gradient-based algorithms, the stability and speed of convergence are coupled and are controlled by the step size. For initial fast convergence one usually requires a large step size, however, increasing the step size without bound leads to algorithmic instability. We address the problem of selecting a stable step size for SIPEX-G when maximizing the cost function in (3) using (8) in the following theorem.

Theorem 2. The following upper bound on the step size of SIPEX-G guarantees stable convergence to the global maximum of (3) using the steepest ascent optimization approach. Nevertheless, larger step sizes might be stable as well.

$$\eta \leq \frac{1}{\sum_{p=1}^{n-1} \sum_{q=p+1}^{n} (-\gamma_p \lambda_p - \gamma_q \lambda_q + \gamma_p \lambda_q + \gamma_q \lambda_p)} \tag{9}$$

where λ_j is the j^{th} largest eigenvalue of the input covariance matrix.

Proof. In the Appendix.

We have previously studied the performance of the SIPEX-G algorithm and how it compares to several benchmark algorithms, including Sanger's rule [7], APEX [2], and LMSER [11], in two previous publications [12,13]. Extensive Monte Carlo simulations performed for this purpose demonstrated the superior performance of SIPEX-G both in terms of convergence speed and final accuracy in estimating the eigenvectors of the input covariance matrix. The case studies for these simulations included synthetic Gaussian data, recorded violin time-series analysis, and adaptive filter training (Wiener solution) using PCA. From these simulations, the LMSER algorithm proved to be the strongest competitor. For this reason, the proposed algorithm is only compared to LMSER in the simulations that follow.

III. NON-STATIONARY SUBSPACE TRACKING WITH SIPEX-G AND LMSER

In order to demonstrate the superior speed and accuracy of SIPEX-G in subspace tracking, we present in this section a synthetic non-stationary scenario where three colored random time series, having varying covariance matrices were concatenated. The composite time series is then presented to SIPEX-G, which used a step size of 0.002 and a forgetting factor of 0.999, to estimate the 3 principal components. The scalar gains of SIPEX-G were set to [3,2,1]. As expected, the speed of convergence and accuracy depend mainly on the forgetting factor.

The tracking results for this case study are depicted in Fig. 1, where the three eigenvalues of the non-stationary covariance matrix take 3 different values during the course of time, to wit, the actual changes to each eigenvalue is a sum of step functions. The three asterisks that occur at a time of 5000 samples represent the constant true values of each of the three eigenvalues hitherto (from time 0 to 5000). Likewise for the asterisks at time 10000 and 17000 samples represent the true eigenvalues for the preceding stationary segments. Clearly, SIPEX-G is able to track the changes in the subspace structure of the input signal fast and with acceptable accuracy.

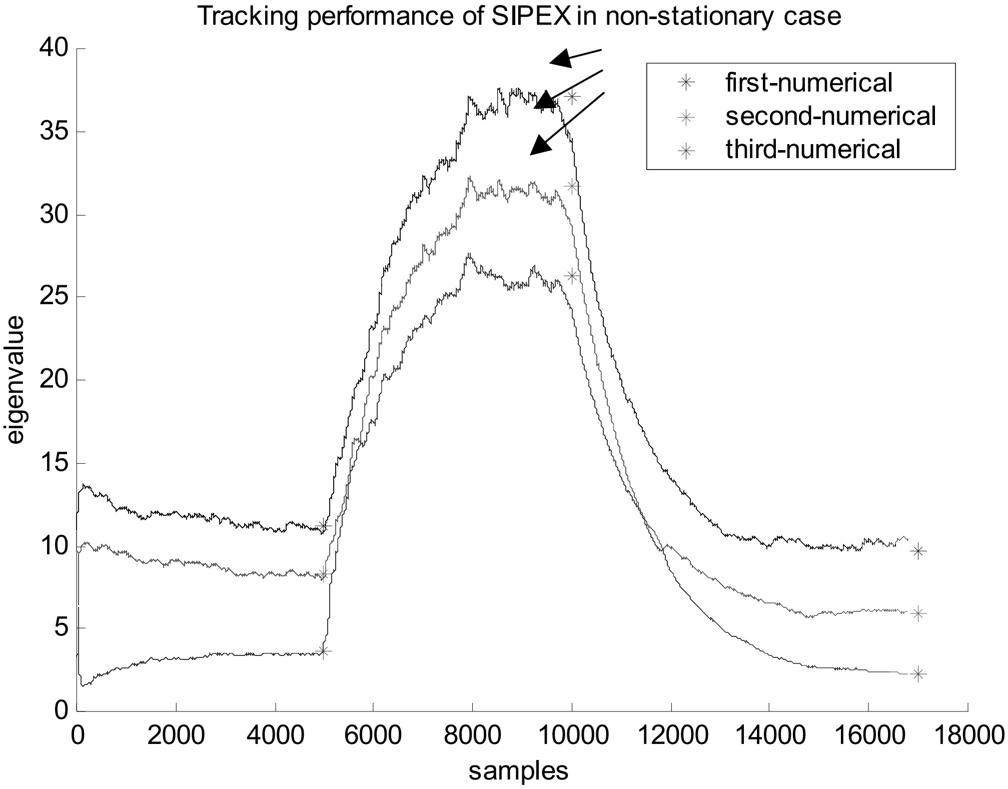


Figure 1. SIPEX-G tracking the eigenvalues of the covariance matrix of a 3-dimensional non-stationary random process. The reference eigenvalues indicated by (*) on the plot are determined using the MATLAB® *eig* function.

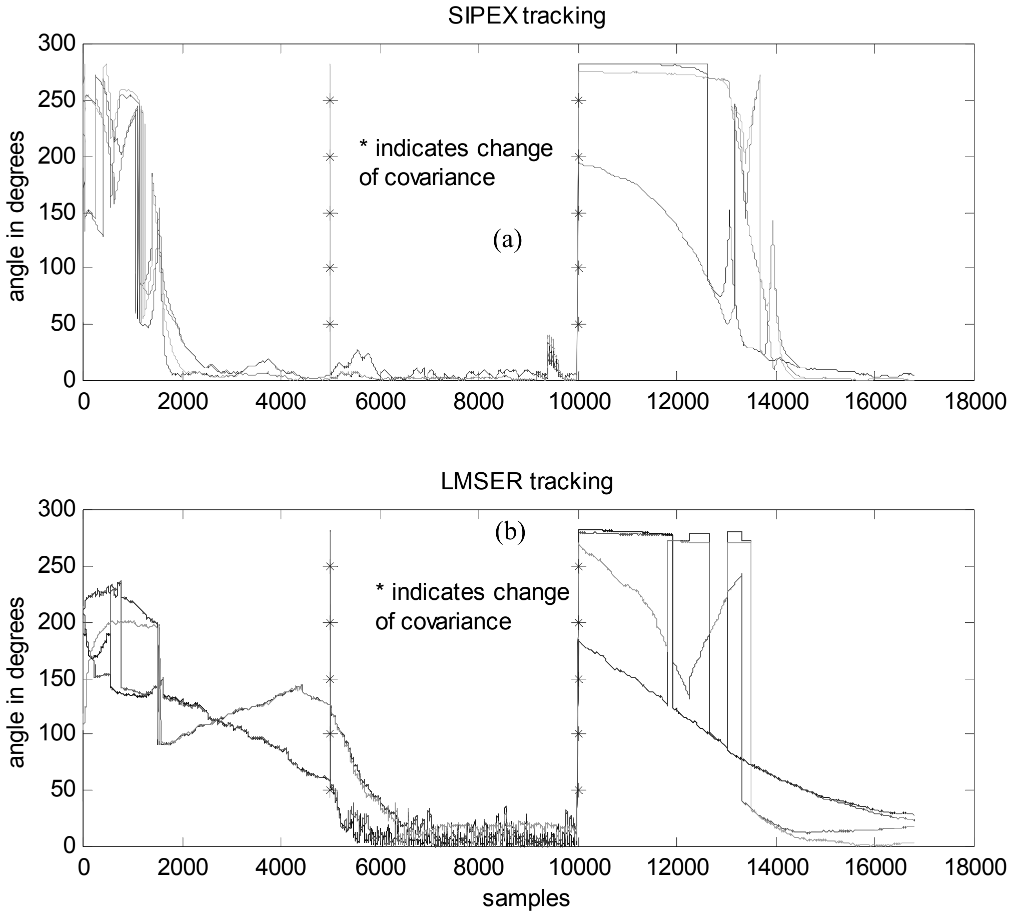


Figure 2. Comparison of a) SIPEX-G and b) LMSER tracking the eigenvectors of the non-stationary covariance matrix of a 4-dimensional random process. The angle between estimated and actual eigenvectors are presented in degrees. The reference eigenvectors are evaluated using the MATLAB® *eig* function. Sudden changes are introduced in the covariance matrix at times of 5000 and 10000 samples, which are indicated by asterisks in the plot.

As a second result, we present a comparison of the performances of SIPEX-G and LMSER on the same experimental setup. The same composite time series, having varying eigenspreads, is now embedded in a 4-dimensional space. We use both SIPEX-G and LMSER to estimate all four principal components. In the tracking results presented in Fig. 2, SIPEX-G uses a constant step size of 0.005 and LMSER uses the maximum step size of 0.001 to ensure stability in all stationary regions. The scalar gains of both algorithms were again set to [4,3,2,1]. The angles between the 4 estimated and actual eigenvectors of the non-stationary covariance matrix are shown (in degrees) to reduce to the desired value of zero, almost accurately for SIPEX-G, in particular. The sudden jumps in the angles within the 3rd regime are due to the switching between reference eigenvectors along the gradient trajectory; these are not of much concern.

The convergence results of SIPEX presented in Fig. 1 exhibits the expected transition behavior after each abrupt change in the input covariance matrix. In Fig. 2, however, the first abrupt change *seems* to be smoothly absorbed by SIPEX. On the other hand, LMSER, which was slow to converge during the first regime, converges to the solution rapidly after the first abrupt change. These observations should not be generalized! Notice that SIPEX still experiences the transition period after the second abrupt change, and LMSER is still slow to converge. The behavior of these two algorithms during the first transition can be explained by the fact that the first abrupt change does not instigate a large deviation in the eigenvectors and the state of both algorithms are, therefore, already close to their optimal solutions for the second regime. Since, SIPEX is faster to converge, in general, it acts quickly and recovers from the transition fast (still there is a short transition phase; notice the small *bump* in the angle errors immediately following the change). LMSER seems to converge faster because the change results in an optimal solution that is closer to the current state of this algorithm, compared to the optimal solution of the first regime; therefore, LMSER speeds-up.

These two experimental results obtained using non-stationary covariance matrices demonstrate the fast and accurate convergence of SIPEX-G as well as its ability to track changes in a non-stationary environment. Fig. 1 demonstrates that the eigenvalues of the changing covariance matrix are tracked, and Fig. 2 demonstrates that the eigenvectors (which are more crucial to the multiuser detection application) are also tracked.

IV. APPLICATION OF SIPEX-G TO COMPLEX-VALUED SIGNALS

Applications requiring the principal component analysis of complex-valued signals often arise in signal processing. In fact, multiuser detection in digital communications is one such application when modulation schemes with complex-valued symbols are employed. Another application is direction of arrival estimation; this problem deals with the estimation of multiple source directions when signals from these sources impinge on an array of antennas. Once the source directions are estimated, this information could be used to improve the signal quality for a specific user by re-orienting the antennas or by algorithmically processing the signal.

Direction of Arrival Estimation. Subspace-based methods for estimating the direction of arrival (DOA) of signals impinging on an array of sensors have been researched extensively. State-space method [25], ESPRIT [26], MUSIC [27], and Min-Norm [28] are examples of these approaches to the DOA problem. These methods all require the eigendecomposition of the covariance matrix of the signal, yet the alternative SWEDE [29], achieves subspace estimation without eigendecomposition. In this section, we will focus on the MUSIC algorithm, which basically uses Sanger's rule [7] for the eigendecomposition task. We will then replace Sanger's rule with our SIPEX-G algorithm.

The DOA problem is formulated as follows. A linear array of *n* sensors receive a mixture of *m* source signals plus an additive complex Gaussian noise whose variance is smaller than those of the signals.

$$\mathbf{x}(k) = \mathbf{D}(\mathbf{\Phi})\mathbf{s}(k) + \mathbf{v}(k) \tag{10}$$

where $\mathbf{D}(\mathbf{\Phi}) = [\mathbf{d}(\varphi_1)...\mathbf{d}(\varphi_m)]$ is the *n*x*m* steering matrix with $\mathbf{\Phi}$ denoting the vector of direction angles of the sources [27,29]. The random vectors $\mathbf{x}(k)$, $\mathbf{s}(k)$, and $\mathbf{v}(k)$ are *n*x1, *m*x1, and *n*x1 complex Gaussian distributed random vectors, respectively. Using this formulation, the covariance matrix of **x**, the received signal vector at the sensors, can be expressed as

$$\mathbf{\Sigma_x} = E[\mathbf{x}(k)\mathbf{x}^H(k)] = \mathbf{D}(\mathbf{\Phi})\mathbf{\Sigma_s}\mathbf{D}(\mathbf{\Phi})^H + \sigma^2\mathbf{I} \tag{11}$$

assuming uncorrelated, equal-power noise at each sensor. In order to solve for the complex eigenvectors of $\mathbf{\Sigma_x}$, we define

$$\mathbf{\Sigma}^c = \begin{bmatrix} \mathrm{Re}[\mathbf{\Sigma_x}] & -\mathrm{Im}[\mathbf{\Sigma_x}] \\ \mathrm{Im}[\mathbf{\Sigma_x}] & \mathrm{Re}[\mathbf{\Sigma_x}] \end{bmatrix}, \quad \mathbf{e}^c = \begin{bmatrix} \mathrm{Re}[\mathbf{e_x}] \\ \mathrm{Im}[\mathbf{e_x}] \end{bmatrix} \tag{12}$$

where $\mathbf{\Sigma}^c$ is a real matrix that is twice the number of rows and columns of $\mathbf{\Sigma_x}$, $\mathbf{e}^c$ is a single eigenvector of $\mathbf{\Sigma}^c$ corresponding to the eigenvector, $\mathbf{e_x}$, of $\mathbf{\Sigma_x}$. $\mathbf{\Sigma}^c$ can be computed from the samples by

$$\mathbf{\Sigma}^c = E[\mathbf{x}^c\mathbf{x}^{cT}], \quad \mathbf{x}^c = \begin{bmatrix} \mathrm{Re}[\mathbf{x}] & -\mathrm{Im}[\mathbf{x}] \\ \mathrm{Im}[\mathbf{x}] & \mathrm{Re}[\mathbf{x}] \end{bmatrix} \tag{13}$$

When the number of sources is known, the task of DOA estimation reduces to finding the *n-m* minor components of the covariance matrix $\mathbf{\Sigma_x}$ and determining the *m* minima of the cost function

$$f(\phi) = \mathbf{d}(\phi)^H (\mathbf{W_v} \mathbf{W_v}^H) \mathbf{d}(\phi) \tag{14}$$

where $\mathbf{W_v} = [\mathbf{e}_{m+1} \ldots \mathbf{e}_n]$ is the matrix formed by the eigenvectors corresponding to the minor components. This is called the multiple signal classification (MUSIC) algorithm for DOA estimation.

When we apply SIPEX-G, with gains [6,5,4,3,2,1] and step size 0.01, to the MUSIC algorithm outlined above, for a 2-source 3-sensor case with an SNR of approximately 20dB, we obtain the result presented in Fig. 3. The PCA problem for this case study is $2n = 6$ dimensional, and there are three eigenvalues, each of multiplicity two. In this simulation, the two minima of the function in (14) are determined by simply evaluating it over the ϕ values in the interval $[0,\pi]$.

Notice that, in order to apply the SIPEX-G algorithm, which was originally designed to extract real-valued eigenvectors from a real-valued symmetric covariance matrix, to the complex-valued DOA problem, we have introduced the modifications outlined in (12) and (13). As a direct consequence of the additional degrees of freedom introduced by the imaginary parts of the data, each n-dimensional PCA problem in the complex domain becomes a $2n$-dimensional problem in real values.

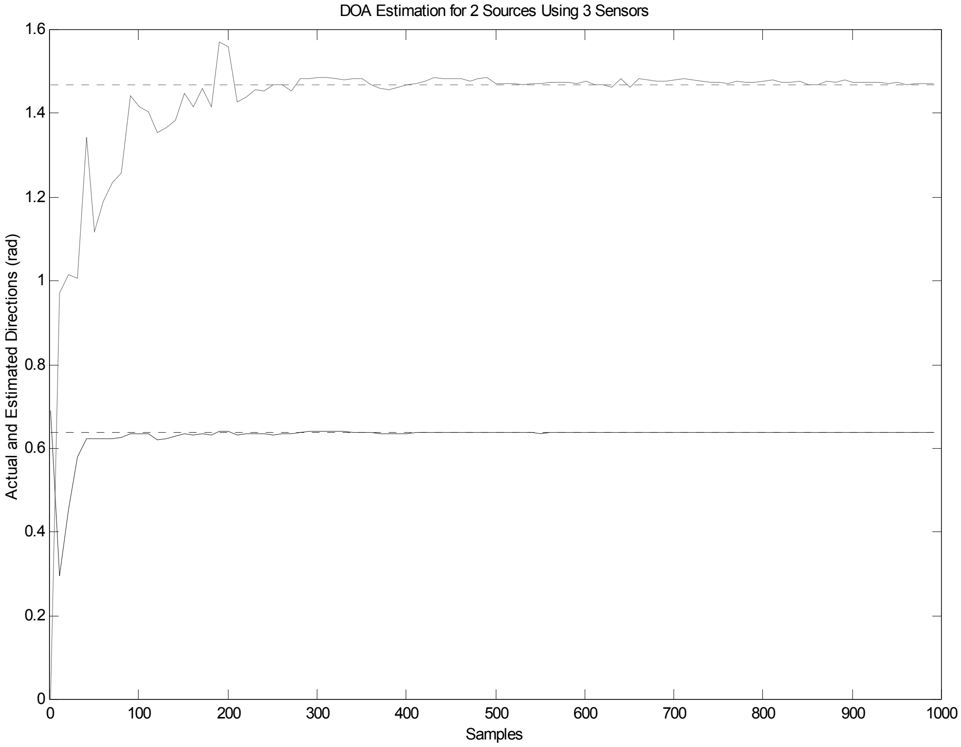


Figure 3. Actual and estimated DOA for a 2-source 3-sensor case with 20dB SNR.

V. APPLICATION OF SIPEX-G TO BLIND MULTIUSER DETECTION

In this section, we present results from the application of SIPEX-G to adaptive blind multiuser detection in DS-CDMA communication systems. For this purpose, we will adopt the synchronous signal model provided by Wang & Poor in [22]. For the sake of having a self-contained text (as much as possible), we will briefly describe this signal model below. Although we use the synchronous signal model in this section for simplicity, the extension of the application of SIPEX-G to the asynchronous case is possible (at the expense of increasing the PCA subspace dimensionality). The necessary steps to achieve this have already been clearly described by Wang *et al.* [19,20]. Therefore, we will not consider the asynchronous case here explicitly.

We consider a baseband digital direct sequence CDMA network of K users. In the synchronous scheme, the received signal can be expressed as $r(t) = S(t) + \sigma n(t)$, where $n(t)$ is zero-mean unit-variance white Gaussian noise and $S(t)$ is the superposition of the data signals of the K users active in the system. We have

$$\begin{aligned} s_k(t) &= \sum_{j=0}^{N-1} \beta_j^k \psi(t - jT_c) \\ S(t) &= \sum_{k=1}^{K} A_k \sum_{i=-M}^{M} b_k[i] s_k(t - iT) \end{aligned} \tag{15}$$

where $\psi(t)$ is the chip waveform with support $[0,T_c]$, $b_k[i]$ is the iid equi-probable bit sequence of user k, A_k is the channel gain for user k's signal and $\beta_j^k, j = 0,..., N-1$ is the signature sequence (also known as the spreading sequence) for user k. T is the bit duration and, by definition, it is $T = NT_c$. If we collect the signature sequence of the k^{th} user in a vector $\mathbf{s}_k$ and normalize its norm to unity, and if we collect all the samples (at fractions of the chip duration T_c) during a time interval of T seconds (called the bit duration) in a vector $\mathbf{r}$, we get the following vector equation

$$\mathbf{r} = \sum_{k=1}^{K} A_k b_k \mathbf{s}_k + \sigma\ \mathbf{n} \tag{16}$$

where $\mathbf{s}_k = \begin{bmatrix}\beta_0^k & \cdots & \beta_{N-1}^k\end{bmatrix}^T / \sqrt{N}$ and $\mathbf{n}$ is a zero-mean random process having an identity ($\mathbf{I}_N$) covariance matrix. We assume that the signature waveform vectors of different users are linearly independent. Defining $\mathbf{A} = diag(A_1^2,..., A_K^2)$ and $\mathbf{S} = [\mathbf{s}_1 \quad \cdots \quad \mathbf{s}_K]$, the covariance matrix of the random vector $\mathbf{r}$ is found to be

$$\mathbf{\Sigma_r} = E[\mathbf{r} \cdot \mathbf{r}^T] = \mathbf{SAS}^T + \sigma^2 \mathbf{I}_N \tag{17}$$

The covariance matrix can be decomposed into its signal and noise subspaces as

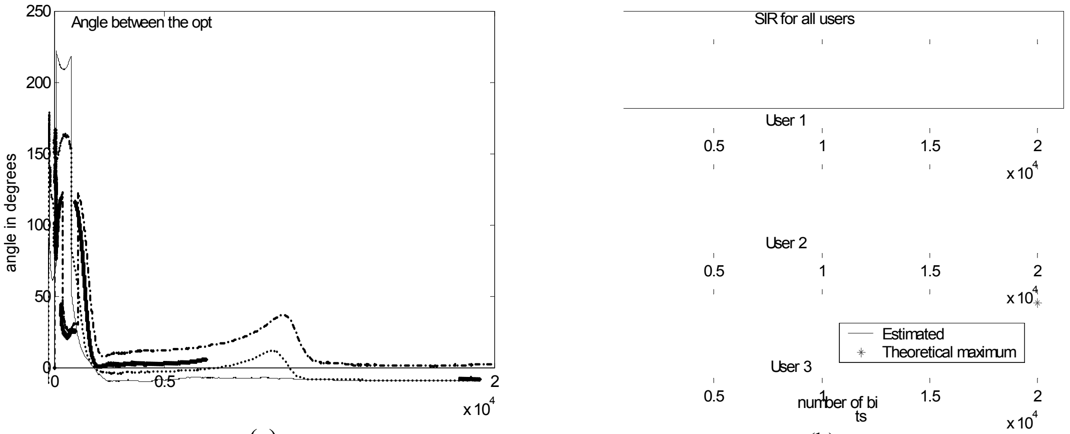
$$\mathbf{\Sigma_r} = \mathbf{U \Lambda U}^T = [\mathbf{U_s} \quad \mathbf{U_n}] \begin{bmatrix} \mathbf{\Lambda_s} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda_n} \end{bmatrix} \begin{bmatrix} \mathbf{U_s}^T \\ \mathbf{U_n}^T \end{bmatrix} \tag{18}$$

where the eigenvalues and their corresponding eigenvectors are in descending order (notice that the intrinsic assumption in this approach is that the noise power is smaller than the smallest signal power). The signal subspace is *K* dimensional in the *N* dimensional complete space (assuming we get one sample for every chip duration). Ideally, the signal-space eigenvectors of the covariance matrix are the spreading sequences of all the users, as is evident from (17). In the blind detection scheme, we assume that the receiver has knowledge of the signature waveform of only one user. For the k^{th} user, it can be shown that the minimum MSE receiver is given by [22]

$$\hat{b}_k[i] = \mathbf{w}^T(k)\mathbf{r}[i] \tag{19}$$

where $\mathbf{r}[i]$ is the received signal vector of dimension *N* at the i^{th} symbol duration [22], and

$$\mathbf{w}(k) = \frac{\mathbf{U_s \Lambda_s}^{-1} \mathbf{U_s}^T \mathbf{s}_k}{[\mathbf{s}_k^T \mathbf{U_s \Lambda_s}^{-1} \mathbf{U_s}^T \mathbf{s}_k]} \tag{20}$$



(a) (b)

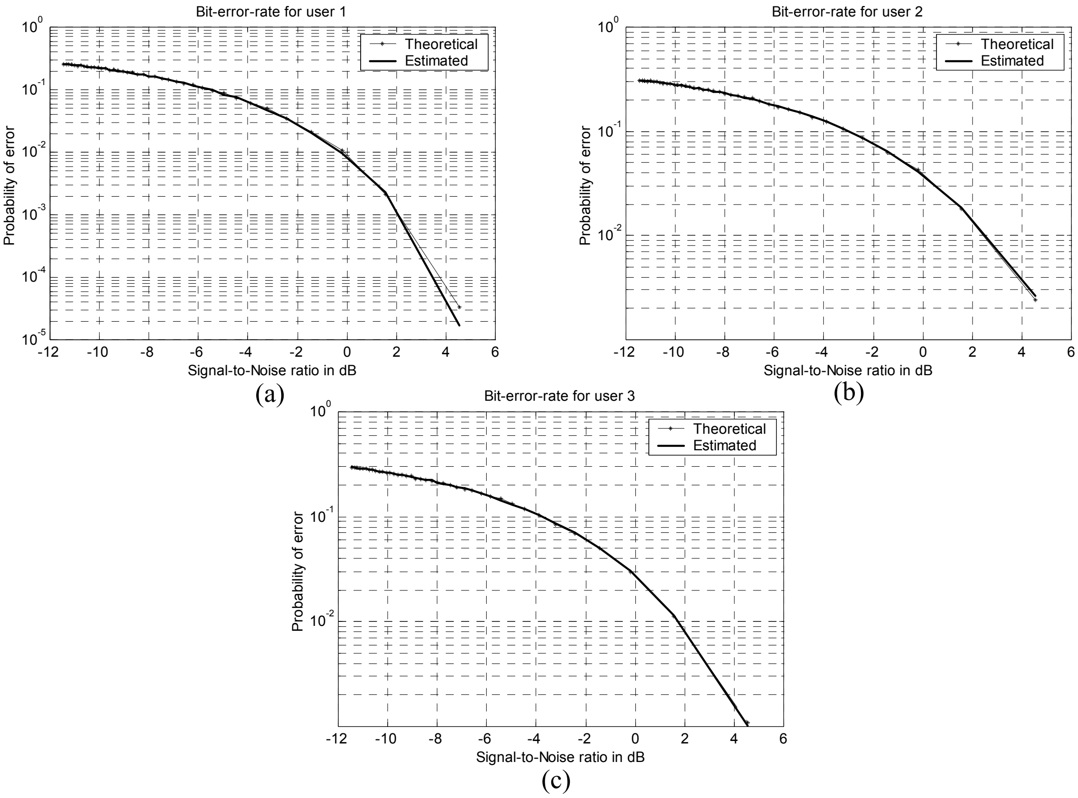
Figure 4. SIPEX-G in blind multiuser detection a) the angle (in degrees) between the estimated and true optimal receiver gain vectors b) the theoretical maximum and actual SIR performances obtained by the estimated optimal detectors for all users versus the number of symbols.

The task is to estimate the *K*-dimensional signal subspace $\mathbf{U_s}$. In the simulations, we use the SIPEX-G algorithm on the samples $\mathbf{r}[i]$ of the received signal vector. For the spreading sequences, we utilize (BPSK) random signature sequences of length $N = 7$. There are 3 active users in the system and each transmits a 60000-length (BPSK) symbol sequence (synchronously). The step size for SIPEX-G is held constant at 0.002. In order to show the convergence of the detectors to their optimal values, we use two measures: the angle between the estimated and optimal detector coefficients (evaluated by taking the *arccos* of the direction cosine between these two vectors), and the signal-to-interference ratio (SIR) between the desired user's signal and the combined multiple access interference (MAI) and channel noise defined in [16]. The SIR for the j^{th} user is defined as

$$SIR_j = \frac{A_j^2\left(\mathbf{w}_j^T s_j\right)^2}{N_o\left\|\mathbf{w}_j\right\|^2 + \sum_{\substack{k=1\\k\neq j}}^{K} A_k^2\left(\mathbf{w}_j^T \mathbf{s}_k\right)^2} \tag{21}$$

First, we show the convergence of the receivers computed using the estimated eigendecomposition of the signal covariance matrix to their optimal values (evaluated using the MATLAB® *eig* function). Notice in Fig. 4a that the receivers of all users converge to their corresponding optimal values and in Fig. 4b, the maximum theoretical SIR possible for each user is achieved in less than 2000 samples. Only one update was performed for each sample; however, if time and computational bandwidth permits, multiple updates per sample could be performed to allow convergence to the solution with an even smaller number of symbols.

Finally, we depict the bit-error-rate (BER) versus signal-to-noise-ratio (SNR) plots for all three users in the above adaptive blind multiuser detection scenario, where SIPEX-G is used to determine the optimal blind MMSE receivers for the users. The BER curves shown in Fig. 5 exhibit perfect match with the theoretical expectations, which are superimposed on each plot. In Fig. 5a, the numerically estimated BER is smaller than its theoretical expectation at large SNR values, because of the limited number of symbols to estimate such small probabilities.



(a) (b)

(c)

Figure 5. The estimated (60000 symbols) and theoretical (*) BER versus SNR curves for all users using detectors determined by the SIPEX-G algorithm.

In this section, we have considered the application of SIPEX-G to adaptive blind multiuser detection in CDMA communications. We have demonstrated that since the signal model permits the use of subspace approaches, and since SIPEX-G is a fast, robust subspace tracking PCA algorithm, its utilization in the blind multiuser detection problem yields very good results. Although our multiuser detection example did not involve complex-valued modulation schemes, the algorithm could be applied to these cases with ease following the modifications outlined in the DOA example of the previous section.

VI. CONCLUSIONS

PCA is an important statistical methodology that finds applications in the solutions to many important practical problems of engineering including signal processing and communication. In this paper, we have presented a new, fast, and robust PCA algorithm, where the topology is based on Givens rotations to guarantee orthonormality of the estimated PCA weight matrix at each iteration step. The cost function is parameterized in terms of the weight matrix and the input covariance matrix, which is estimated recursively from the input samples, to guarantee robustness and accuracy of the final eigenvector estimates.

We have previously established the superior performance of SIPEX-G by comparing with benchmark PCA algorithms such as Sanger's rule, APEX, and LMSER in a variety of problems including both synthetic and real data. In this paper, we have demonstrated the superiority of SIPEX-G over LMSER in subspace tracking when the covariance matrix of the random-process is non-stationary. In addition, we have demonstrated the application of SIPEX-G to complex-valued PCA through the use of a direction of arrival estimation example.

The SIPEX-G algorithm was also tested on the blind multiuser detection problem in DS-CDMA communication systems using the synchronous signal model with a subspace approach. The simulation results indicated that SIPEX-G was successful in determining the optimal blind minimum MSE receivers accurately using a very small number of samples. In practical applications, the data efficiency of an adaptive algorithm is extremely important. Therefore, SIPEX-G is a valuable alternative to existing blind multiuser detection algorithms as it utilizes the samples efficiently to determine the optimal detector coefficients.

Further study might focus on investigating the performance of SIPEX-G on the asynchronous case, although we believe that the results will demonstrate similar successful performance both in terms of SIR and BER. In addition, fixed-point versions of SIPEX-G could be derived, resulting in improved convergence speed.

Acknowledgments: This work was partially supported by NSF grant ECS-9900394.

APPENDIX

Proof of Theorem 1. In accordance with the conditions stated in the theorem, suppose that **y**=**Rx**, where **R** is an orthonormal matrix. Let $\boldsymbol{\Sigma}_{\mathbf{x}}$ be the covariance matrix of the random vector **x**. Let $\boldsymbol{\Sigma}_{\mathbf{x}}\mathbf{=Q}_{\mathbf{x}}\boldsymbol{\Lambda}_{\mathbf{x}}\mathbf{Q}_{\mathbf{x}}$ be the eigendecomposition of this covariance matrix, where $\mathbf{Q}_{\mathbf{x}}$ is an orthonormal matrix consisting of the eigenvectors ordered in accordance with the ordering of the eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}_{\mathbf{x}}$. Notice that we can express an arbitrary orthonormal matrix **R** as the product $\mathbf{R} = \overline{\mathbf{R}}\mathbf{Q}_{\mathbf{x}}^T$, where $\overline{\mathbf{R}}$ is also an orthonormal matrix. Consider the following parameterization of the cost function *J* in terms of $\overline{\mathbf{R}}$.

$$J(\overline{\mathbf{R}}) = \sum_{s=1}^{n-1}\gamma_s[\mathbf{R}\boldsymbol{\Sigma}_{\mathbf{x}}\mathbf{R}^T]_{ss} = \sum_{s=1}^{n-1}\gamma_s[\overline{\mathbf{R}}\boldsymbol{\Lambda}_{\mathbf{x}}\overline{\mathbf{R}}^T]_{ss} \tag{A.1}$$

In general, if $\boldsymbol{\delta}\overline{\mathbf{R}}$ is free to take any (small) value to represent perturbations to any allowed direction, we can write the following for the value of the cost function at the perturbed point.

$$J(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})=\sum_{s=1}^{n-1}\gamma_s[(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})\boldsymbol{\Lambda}_{\mathbf{x}}(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})^T]_{ss} \tag{A.2}$$

Expanding the product of parentheses and dropping the quadratic term on the perturbation in (A.2), and subtracting both sides of (A.1) from both sides of (A.2), we obtain the increment in J as

$$J(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})-J(\overline{\mathbf{R}})\approx 2\sum_{s=1}^{n-1}\gamma_s[\overline{\mathbf{R}}\boldsymbol{\Lambda}_{\mathbf{x}}\boldsymbol{\delta}\overline{\mathbf{R}}^T]_{ss} \tag{A.3}$$

Notice that the constraint of orthonormality of $\overline{\mathbf{R}}$ and $\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}}$ translates to the condition $(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})^T(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})=\mathbf{I}$, which, after dropping the quadratic term once again, becomes

$$\overline{\mathbf{R}}^{\mathbf{T}}\boldsymbol{\delta}\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}}^{\mathbf{T}}\overline{\mathbf{R}}\approx\mathbf{0} \tag{A.4}$$

In addition, the orthonormality constraint on $\overline{\mathbf{R}}$ and $\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}}$ dictates that they have entries whose absolute values are smaller than or equal to 1. Thus, for $\overline{\mathbf{R}}$ values, where $\overline{\mathbf{R}}\neq\mathbf{P}$ ($\mathbf{P}$ is a permutation matrix which is allowed to take both ± 1 values at its nonzero entries), there is no problem considering perturbations in (+) and (-) directions, i.e. $\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}}$ and $\overline{\mathbf{R}}-\boldsymbol{\delta}\overline{\mathbf{R}}$. Even if $\overline{\mathbf{R}}$ has some, but not all, rows that are of the form $\mathbf{e}_i^T=[0\ \cdots\ 1\ \cdots\ 0]$, there still exists valid $\pm\boldsymbol{\delta}\overline{\mathbf{R}}$ choices, which allow us to perturb $\overline{\mathbf{R}}$ in both directions without violating the orthonormality condition. In this situation, however, the additional constraint $\left|(\overline{\mathbf{R}}\pm\boldsymbol{\delta}\overline{\mathbf{R}})_{ij}\right|\leq 1$ must be considered in the choice. Once we have determined a suitable $\pm\boldsymbol{\delta}\overline{\mathbf{R}}$ matrix, we see that

$$\begin{aligned}J(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})-J(\overline{\mathbf{R}})&\approx 2\sum_{s=1}^{n-1}\gamma_s[\overline{\mathbf{R}}\boldsymbol{\Lambda}_{\mathbf{x}}\boldsymbol{\delta}\overline{\mathbf{R}}^T]_{ss}\\ J(\overline{\mathbf{R}}-\boldsymbol{\delta}\overline{\mathbf{R}})-J(\overline{\mathbf{R}})&\approx 2\sum_{s=1}^{n-1}\gamma_s[\overline{\mathbf{R}}\boldsymbol{\Lambda}_{\mathbf{x}}(-\boldsymbol{\delta}\overline{\mathbf{R}}^T)]_{ss}=-\left(J(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})-J(\overline{\mathbf{R}})\right)\end{aligned} \tag{A.5}$$

$\therefore$ At any point where $\overline{\mathbf{R}}\neq\mathbf{P}$, the cost function J has no stationary points.

Now specifically consider the case where

$$\overline{\mathbf{R}}=\begin{bmatrix}\mathbf{P}_{n-1} & \mathbf{0}\\ \mathbf{0} & \pm 1\end{bmatrix} \tag{A.6}$$

where $\mathbf{P}_{n\text{-}1}$ is an (n-1)x(n-1) permutation matrix as described above with possible negative entries. We can show that the cost function J has stationary points at these values of the rotation matrix. Consider

$$\Delta J \overset{\Delta}{=} J(\overline{\mathbf{R}}+\boldsymbol{\delta}\overline{\mathbf{R}})-J(\overline{\mathbf{R}})=2\sum_{s=1}^{n-1}\gamma_s[\overline{\mathbf{R}}\boldsymbol{\Lambda}_{\mathbf{x}}\boldsymbol{\delta}\overline{\mathbf{R}}^T]_{ss}=\ldots=2\sum_{s=1}^{n-1}\sum_{j=1}^{n}\gamma_s\lambda_j\bar{r}_{sj}\delta\bar{r}_{sj} \tag{A.7}$$

with $\bar{r}_{sj}$ and $\delta\bar{r}_{sj}$ denote the entries of $\overline{\mathbf{R}}$ and $\boldsymbol{\delta}\overline{\mathbf{R}}$, respectively. Notice that, for the specific choice of (A.6), $\bar{r}_{sj}$ are mostly zeros. In fact, only the entries $\bar{r}_{sj_s}=\pm 1$, for s=1,…,n-1. In addition, due to the constraint $\left|(\overline{\mathbf{R}}\pm\boldsymbol{\delta}\overline{\mathbf{R}})_{ij}\right|\leq 1$, the corresponding perturbation entries are $\delta\bar{r}_{sj_o}=0$, for s=1,…,n-1. Therefore, in the last expression of (A.7), for all terms in the double summation, either $\bar{r}_{sj}=0,\ \ j\neq j_s$ or $\delta\bar{r}_{sj}=0,\ \ j=j_s$ leading to $\Delta J=0$, therefore, points of the form in (A.6) are stationary points. Similarly, we can show that all points of the form $\overline{\mathbf{R}}=\mathbf{P}$ are stationary points.

Alternatively, we could show that these points are stationary points by evaluating the gradient by brute force and showing its equivalence to zero. Consider the following explicit form of the cost function J in terms of $\bar{r}_{sj}$.

$$J(\overline{\mathbf{R}})=\sum_{s=1}^{n-1}\gamma_s[\overline{\mathbf{R}}\boldsymbol{\Lambda}_{\mathbf{x}}\overline{\mathbf{R}}^T]_{ss}=\ldots=\sum_{s=1}^{n-1}\gamma_s\sum_{j=1}^{n}\lambda_j\bar{r}_{sj}^2 \tag{A.8}$$

We represent the orthonormal matrix $\overline{\mathbf{R}}$ in terms of the Givens rotations according to

$$\overline{\mathbf{R}}=\prod_{p=1}^{n-1}\prod_{q=p+1}^{n}\mathbf{R}^{pq}(\bar{\theta}_{pq})=\mathbf{R}^{12}\mathbf{R}^{13}\ldots\mathbf{R}^{n-1,n} \tag{A.9}$$

Recall that we have

$$\mathbf{R}^{pq}(\theta)\overset{\Delta}{=}\begin{bmatrix}1&0&0&0&0&0&0\\0&\ddots&0&0&0&0&0\\0&0&\cos\theta&0&-\sin\theta&0&0\\0&0&0&1&0&0&0\\0&0&\sin\theta&0&\cos\theta&0&0\\0&0&0&0&0&\ddots&0\\0&0&0&0&0&0&1\end{bmatrix}\quad \mathbf{R}'^{pq}(\theta)\overset{\Delta}{=}\begin{bmatrix}0&0&0&0&0&0&0\\0&\ddots&0&0&0&0&0\\0&0&-\sin\theta&0&-\cos\theta&0&0\\0&0&0&0&0&0&0\\0&0&\cos\theta&0&-\sin\theta&0&0\\0&0&0&0&0&\ddots&0\\0&0&0&0&0&0&0\end{bmatrix} \tag{A.10}$$

where the terms with the angle parameter appear at the intersection of p^{th} and q^{th} rows and columns. From this configuration, we see the following. *Fact:* $\overline{\mathbf{R}} = \mathbf{P} \Leftrightarrow \overline{\theta}_{pq} = k\pi/2$, $\forall p, q$ where k is some integer. Letting $\overline{\mathbf{R}} = \mathbf{R}^{(1)}\mathbf{R}^{\mathbf{kl}}\mathbf{R}^{(2)}$, $\overline{\mathbf{R}}' \overset{\Delta}{=} \partial\overline{\mathbf{R}}/\partial\overline{\boldsymbol{\theta}}_{\mathbf{kl}} = \mathbf{R}^{(1)}\mathbf{R}'^{\mathbf{kl}}\mathbf{R}^{(2)}$, where $\mathbf{R}'^{kl} \overset{\Delta}{=} \partial\mathbf{R}^{kl}/\partial\overline{\theta}_{kl}$, the matrix $\mathbf{R}^{(1)}$ is the product of matrices preceding $\mathbf{R}^{\mathbf{pq}}$, and $\mathbf{R}^{(2)}$ is the product of matrices following $\mathbf{R}^{\mathbf{pq}}$ in the definition given in (A.9). They are both independent of $\overline{\theta}_{pq}$. Assume $\overline{\theta}_{kl} = (odd)\pi/2$, then we have

$$\mathbf{R}^{kl} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pm 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}'^{pq}(\theta) \overset{\Delta}{=} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \pm 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.11}$$

When all Givens angles are multiples of $\pi/2$, $\mathbf{R}^{(1)}$=$\mathbf{P}^{(1)}$ and $\mathbf{R}^{(2)}$= $\mathbf{P}^{(2)}$ are also permutation matrices (with ± 1 entries). And of course we have $\overline{\mathbf{R}} = \mathbf{P}$, so

$$\mathbf{P} = \mathbf{P}^{(1)}\mathbf{R}^{\mathbf{kl}}\mathbf{P}^{(2)} = \mathbf{P}^{(1)} \cdot \begin{bmatrix} \mathrm{P}_1^2 \\ \vdots \\ \pm\mathrm{P}_l^2 \\ \vdots \\ \pm\mathrm{P}_k^2 \\ \vdots \\ \mathrm{P}_n^2 \end{bmatrix}, \quad \mathbf{P}' = \mathbf{P}^{(1)}\mathbf{R}'^{\mathbf{kl}}\mathbf{P}^{(2)} = \mathbf{P}^{(1)} \cdot \begin{bmatrix} 0 \\ \vdots \\ \pm\mathrm{P}_k^2 \\ \vdots \\ \pm\mathrm{P}_l^2 \\ \vdots \\ 0 \end{bmatrix} \tag{A.12}$$

Notice that in **P**, the rows *k* and *l* are interchanged for $\mathbf{P}^{(2)}$. Since $\mathbf{P}^{(1)} \neq (\mathbf{R}^{\mathbf{kl}})^T$ and $\mathbf{P}^{(1)} \neq (\mathbf{R}'^{\mathbf{kl}})^T$ (i.e. not the inverse permutation of both), the element-wise product of **P** and $\mathbf{P}'$ yields a zero matrix. This result is clearly seen for all rows except for the k^{th} and l^{th} rows, since $\mathbf{P}'$ has all zero entries everywhere except for these two rows. Without loss of generality, we assume that $\mathbf{P}^{(1)}$=$\mathbf{I}_n$, is the *n*x*n* identity matrix. We can do this because $\mathbf{P}^{(1)}$ applies the same permutation in both **P** and $\mathbf{P}'$. At the k^{th} row of **P**, we have $\pm\mathbf{P}_l^2$, which is the l^{th} row of $\mathbf{P}^{(2)}$. At the k^{th} row of $\mathbf{P}'$, we have $\pm\mathbf{P}_k^2$, which is the k^{th} row of $\mathbf{P}^{(2)}$. Clearly, the element-wise product of **P** and $\mathbf{P}'$ is also zero at the k^{th} and l^{th} rows, since the element-wise product of $\pm\mathbf{P}_l^2$ and $\pm\mathbf{P}_k^2$ is certainly zero. Hence, for $\overline{\theta}_{kl} = (odd)\pi/2$, the element-wise product of $\overline{\mathbf{R}}$ and $\overline{\mathbf{R}}'$ is a zero matrix. Consequently, the gradient of the cost function vanishes. Similarly, we can show for $\bar{\theta}_{kl} = (even)\pi/2$ that the gradient vanishes at $\overline{\mathbf{R}} = \mathbf{P}$. $\therefore$ For all $\overline{\mathbf{R}} = \mathbf{P}$, where **P** is a permutation matrix (with ± 1 entries), the cost function *J* has a stationary point.

$\therefore$ This two-part proof shows that the cost function *J*(**R**) in (1), where **y=Rx** and **R** is an orthonormal matrix, has stationary points iff $\mathbf{R}=\mathbf{P}\mathbf{Q}_\mathbf{x}^{\mathrm{T}}$, where **P** is a permutation matrix with ± 1 entries. □

Proof Theorem 2. Let $\mathbf{R} = \overline{\mathbf{R}}\mathbf{Q}_\mathbf{x}^T$, where $\overline{\mathbf{R}}$ is also an orthonormal matrix and $\mathbf{Q}_\mathbf{x}$ is the eigenvector matrix ordered in a descending manner. Then, like in (3), we can write the cost function as

$$J(\overline{\mathbf{R}}) = \sum_{s=1}^{n-1}\sum_{j=1}^{n} \gamma_s \lambda_j \bar{r}_{sj}^2 \tag{A.13}$$

where $\bar{r}_{sj}$ is the (*s*,*j*)th entry of $\overline{\mathbf{R}}$. Consider also the parametric expression $\bar{\mathbf{r}}(\overline{\boldsymbol{\Theta}})$ of $\overline{\mathbf{R}}$ in terms of the Givens angles vector (assuming that we rearrange the entries of $\overline{\mathbf{R}}$ into vector form). Then, we can write the first and second order derivatives of *J* with respect to a given angle $\bar{\theta}_{pq}$ as

$$\begin{aligned} &\frac{\partial J}{\partial \bar{\theta}_{pq}} = \frac{\partial J}{\partial \bar{\mathbf{r}}}\frac{\partial \bar{\mathbf{r}}}{\partial \bar{\theta}_{pq}}, \quad \frac{\partial^2 J}{\partial \bar{\theta}_{pq}^2} = \left(\frac{\partial \bar{\mathbf{r}}}{\partial \bar{\theta}_{pq}}\right)^T \frac{\partial^2 J}{\partial \bar{\mathbf{r}}^2}\left(\frac{\partial \bar{\mathbf{r}}}{\partial \bar{\theta}_{pq}}\right) + \frac{\partial J}{\partial \bar{\mathbf{r}}}\frac{\partial^2 \bar{\mathbf{r}}}{\partial \bar{\theta}_{pq}^2} \\ &\frac{\partial^2 J}{\partial \bar{\theta}_{rs}\partial \bar{\theta}_{pq}} = \left(\frac{\partial \bar{\mathbf{r}}}{\partial \bar{\theta}_{rs}}\right)^T \frac{\partial^2 J}{\partial \bar{\mathbf{r}}^2}\left(\frac{\partial \bar{\mathbf{r}}}{\partial \bar{\theta}_{pq}}\right) + \frac{\partial J}{\partial \bar{\mathbf{r}}}\frac{\partial^2 \bar{\mathbf{r}}}{\partial \bar{\theta}_{rs}\partial \bar{\theta}_{pq}} \end{aligned} \tag{A.14}$$

We easily see that for the (*a*,*b*)th and (*c*,*d*)th entries of $\overline{\mathbf{R}}$ (indices not equal)

$$\frac{\partial J}{\partial \bar{r}_{ab}} = 2\gamma_a \lambda_b \bar{r}_{ab} \ , \ \frac{\partial^2 J}{\partial \bar{r}_{cd}\partial \bar{r}_{ab}} = 0 \ , \ \frac{\partial^2 J}{\partial \bar{r}_{ab}^2} = 2\gamma_a \lambda_b \tag{A.15}$$

In addition, we have

$$\frac{\partial \overline{\mathbf{R}}}{\partial \bar{\theta}_{pq}} = \overline{\mathbf{R}}^{(\mathbf{1})}\overline{\mathbf{R}}'^{\mathbf{pq}}\overline{\mathbf{R}}^{(\mathbf{2})} \ , \ \frac{\partial^2 \overline{\mathbf{R}}}{\partial \bar{\theta}_{pq}^2} = \overline{\mathbf{R}}^{(\mathbf{1})}\overline{\mathbf{R}}''^{\mathbf{pq}}\overline{\mathbf{R}}^{(\mathbf{2})} \ , \ \frac{\partial^2 \overline{\mathbf{R}}}{\partial \bar{\theta}_{rs}\partial \bar{\theta}_{pq}} = Don't \ \ Care \tag{A.16}$$

We know that SIPEX-G would converge in a stable fashion to the global maximum if the step size satisfies $\eta \le 2/|\lambda_{\max}|$, where $\lambda_{\max}$ is the maximum eigenvalue (in terms of absolute value) of the Hessian of the cost function with respect to the Givens angles evaluated at the global maximum. Since it is very difficult to determine the eigenvalues of the Hessian matrix, we will use the trace instead. Therefore, the (loose) upper bound for the step size to guarantee stability becomes $\eta \le 2 / \left| trace(\partial^2 J / \partial \overline{\mathbf{\Theta}}^2) \right|$. The trace of the Hessian of the cost function with respect to the Givens angles is given by

$$trace\left(\frac{\partial^2 J}{\partial \overline{\mathbf{\Theta}}^2}\right) = \sum_{p=1}^{n-1} \sum_{q=p+1}^{n} \frac{\partial^2 J}{\partial \overline{\theta}_{pq}^2} = \sum_{p=1}^{n-1} \sum_{q=p+1}^{n} \left[\left(\frac{\partial \overline{\mathbf{R}}}{\partial \overline{\theta}_{pq}} \right)^T \frac{\partial^2 J}{\partial \overline{\mathbf{R}}^2} \left(\frac{\partial \overline{\mathbf{R}}}{\partial \overline{\theta}_{pq}} \right) + \frac{\partial J}{\partial \overline{\mathbf{R}}} \frac{\partial^2 \overline{\mathbf{R}}}{\partial \overline{\theta}_{pq}^2} \right] \tag{A.17}$$

Evaluating this at the global maximum yields (at this point $\overline{\mathbf{R}} = \mathbf{I}$ and all Givens angles are 0)

$$\left| trace\left(\left. \frac{\partial^2 J}{\partial \overline{\Theta}^2} \right|_{\overline{\Theta}=0} \right) \right| = 2 \sum_{p=1}^{n-1} \sum_{q=p+1}^{n} [-\gamma_p \lambda_p - \gamma_q \lambda_q + \gamma_p \lambda_q + \gamma_q \lambda_p] \tag{A.18}$$

Thus, the result in *Theorem 2* is obtained. Notice that this result can be generalized to any local maxima or minima of the proposed cost function by selecting the eigenvalues corresponding to that solution. □

REFERENCES

[1] R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[2] S.Y. Kung, K.I. Diamantaras, J.S. Taur, “Adaptive Principal Component Extraction (APEX) and Applications,” IEEE Transactions on Signal Processing, vol. 42, no. 5, pp. 1202-1217, 1994.

[3] J. Mao, A.K. Jain, “Artificial Neural Networks for Feature Extraction and Multivariate Data Projection,” IEEE Transactions on Neural Networks, vol. 6, no. 2, pp. 296-317, 1995.

[4] Y. Cao, S. Sridharan, M. Moody, “Multichannel Speech Separation by Eigendecomposition and its Application to Co-Talker Interference Removal,” IEEE Transactions on Speech and Audio Processing, vol. 5, no. 3, pp. 209-219, 1997.

[5] G. Golub, C.V. Loan, *Matrix Computation*, John Hopkins University Press, Baltimore, MD, 1993.

[6] E. Oja, *Subspace Methods for Pattern Recognition*, Wiley, New York, 1983.

[7] T.D. Sanger, “Optimal Unsupervised Learning in a Single Layer Linear Feedforward Neural Network,” *Neural Networks*, vol. 2, no. 6, pp. 459-473, 1989.

[8] J. Rubner, K. Schulten, “Development of Feature Detectors by Self Organization,” Biological Cybernetics, vol. 62, pp. 193-199, 1990.

[9] J. Rubner, P. Tavan, “A Self Organizing Network for Principal Component Analysis,” Europhysics Letters, vol. 10, pp. 693-698, 1989.

[10] Y.N. Rao, J. C. Principe, “A Fast, On-Line Algorithm for PCA and its Convergence Characteristics,” Proceedings of NNSP X, vol. 1, pp. 299-307, 2000.

[11] L. Xu, “Least Mean Square Error Reconstruction Principle for Self-Organizing Neural-Nets,” Neural Networks, vol. 6, pp. 627-648, 1993.

[12] D. Erdogmus, Y.N. Rao , J.C. Principe, J. Zhao , K.E. Hild II, "Simultaneous Extraction of Principal Components Using Givens Rotations and Output Variances," to appear in Proceedings of ICASSP’02, May 2002.

[13] D. Erdogmus, Y.N. Rao, J.C. Principe, O. Fontenla-Romero, L. Vielva, "An Efficient, Robust, and Fast Converging Principal Components Extraction Algorithm: SIPEX-G," submitted to EUSIPCO'02, Sep 2002.

[14] S. Verdu, *Multiuser Detection*, Cambridge University Press, Cambridge, UK, 1998.

[15] M. Honig, H.V. Poor, "Adaptive Interference Suppression," in *Wireless Communications: A Signal Processing Perspective* (H.V. Poor & G.W. Wornell eds.), Prentice Hall, New Jersey, 1998, pp. 64-128.

[16] M. Honig, U. Madhow, S. Verdu, "Blind Adaptive Multiuser Detection," IEEE Transactions on Information Theory, vol. 41, no. 4, pp. 944-960, 1995.

[17] M.K. Tsatsanis, "Inverse Filtering Criteria for CDMA Systems," IEEE Transactions on Signal Processing, vol. 45, no. 1, pp. 102-112, 1997.

[18] X. Wang, H.V. Poor, "Blind Equalization and Multiuser Detection in Dispersive CDMA Channels," IEEE Transactions on Communications vol. 46, no. 1, pp. 91-103, 1998.

[19] X. Wang, A. Host-Madsen, "Group-Blind Multiuser Detetion for Uplink CDMA," IEEE Journal on Selected Areas in Communications, vol. 17, no. 11, pp. 1971-1984, Nov 1999.

[20] D. Reynolds, X. Wang, "Adaptive Group-Blind Multiuser Detection Based on a New Subspace Tracking Algorithm," IEEE Transactions on Communications, vol. 49, no. 7, pp. 1135-1141, 2001.

[21] S.E. Bensley, B. Aazhang, "Subspace-based Channel Estimation for Code-Division Multiple-Access Communication Systems," IEEE Transactions on Communications, vol. 44, no. 8, pp. 1009-1020, 1996.

[22] X. Wang, H.V. Poor, "Blind Multiuser Detection: A Subspace Approach," IEEE Transactions on Information Theory, vol. 44, no. 2, pp. 677-691, 1998.

[23] A.N. Barbosa, S.L. Miller, "Adaptive Detection of DS/CDMA Signals in Fading Channels," IEEE Transactions on Communications, vol. 46, no. 1, pp. 115-124, 1998.

[24] I.T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, 1986.

[25] R.J. Vaccaro, Y. Ding, "A New State-Space Approach for Direction Finding," IEEE Transactions on Signal Processing, vol. 42, no. 11, pp.3234-3237, 1994.

[26] K.J.R. Liu, D.P. O'Leary, G.W.Stewart, Y.J.J. Wu, "URV ESPRIT for Tracking Time-Varying Signals," IEEE Transactions on Signal Processing, vol. 42, no. 12, pp. 3441-3448, 1994.

[27] R.O. Schimidt, "Multiple Emitter Location and Signal Parameter Estimation," IEEE Transactions on Antennas Propagation, vol. 34, no. 3, pp.276-280, Mar. 1996.

[28] R. Kumaresan, D.W. Tufts, "Estimating the Angles of Arrival of Multiple Source Plane Waves," IEEE Transactions on Aerospace Electrical Systems, vol. 19, no. 1, pp.134-149, 1983.

[29] A. Eriksson, P. Stoica, T. Soderstrom, "On-Line Subspace Algorithms for Tracking Moving Sources," IEEE Transactions on Signal Processing, vol. 42, no. 9, pp. 2319-2330, 1994.