AN EFFICIENT, ROBUST, AND FAST CONVERGING PRINCIPAL COMPONENTS EXTRACTION ALGORITHM: SIPEX-G

Deniz Erdogmus[1], Yadunandana N. Rao[1], Jose C. Principe[1], Oscar Fontenla-Romero[2], Luis Vielva[3]
[1] Computational NeuroEngineering Laboratory, NEB 454, University of Florida, Gainesville, Florida 32611, USA
[2] Laboratorio de Investigacion y Desarrollo en Inteligencia Artificial, Universidad de A Coruña, 15071 A Coruña, Spain
[3] Grupo de Tratamiento Avanzado de Señal, Universidad de Cantabria, Santander, Spain
e-mail: [deniz ,yadu, principe]@cnel.ufl.edu, oscarfon@mail2.udc.es, luis@dicom.unican.es

ABSTRACT

Principal Components Analysis (PCA) is a very important statistical tool in signal processing, which has found successful applications in numerous engineering problems as well as other fields. In general, an on-line algorithm to adapt the PCA network to determine the principal projections of the input data is desired. The authors have recently introduced a fast, robust, and efficient PCA algorithm called SIPEX-G without detailed comparisons and analysis of performance. In this paper, we investigate the performance of SIPEX-G through Monte Carlo runs on synthetic data and on realistic problems where PCA is applied. These problems include direction of arrival estimation and subspace Wiener filtering.

1 INTRODUCTION

Principal Components Analysis (PCA) established its significance as a fundamental statistical signal processing technique through numerous successful applications including feature extraction, signal estimation, detection, speech separation, linear discriminant analysis, direction of arrival estimation, and subspace filtering [1-5]. Oja's rule [6] ignited an interest among researchers for on-line PCA algorithms. This interest led to the well-known methods Sanger's Rule [7], Rubner-Tavan method [8], and APEX [2], which is an improved version of Rubner-Tavan's method. These conventional methods, although have found their places in applications, have shortcomings in terms of convergence speed and estimation accuracy. There are also fixed-point rules for PCA like the Power rule and the CNEL rule [9,10], however, these procedures also rely heavily on the deflation (or inflation) procedure to determine the intermediate eigenvectors, which prevents the learning algorithms to converge simultaneously to all the principal components.

Xu's LMSER algorithm uses subspace techniques and a diagonal amplification matrix to extract principal components simultaneously [11]. Although LMSER introduces a great improvement over the traditional methods in terms of speed and accuracy, not constraining the search space for the PCA weight matrix to the set of orthonormal matrices, looses valuable information while trying to orthonormalize the estimated eigenvectors.

SIPEX-G (Simultaneous Principal Component Extraction Using a Gradient Algorithm) algorithm on the other hand, employs Givens rotations as an orthonormal parametrization to the PCA weight matrix and uses a robust and consistent estimate of the output variances based on the input covariance matrix in order to converge fast and accurately to the eigenvectors of the underlying covariance matrix [12]. In this paper, we perform an extensive comparison of the SIPEX-G algorithm with Sanger's Rule, APEX, and Xu's LMSER in terms of convergence time and eigenvector estimation accuracy. We will also demonstrate the superior performance of SIPEX-G in direction of arrival and subspace Wiener filtering case studies.

2 SIPEX-G ALGORITHM

It is well known that the solution to the PCA problem is an orthonormal matrix consisting of unit-norm eigenvectors of the input data in its rows. The principal components are defined as the directions in the input space along which the data exhibits maximal variance with an uncorrelatedness constraint between the output components.

Consider a PCA network $y=Rx$, where $x \in \Re^{nx1}$ and $y \in \Re^{nx1}$ are the input and output vectors respectively, and $R \in D \subset \Re^{nxn}$ is the weight matrix, which is restricted to the subset D of orthonormal matrices. It was shown in [12] that the cost function

$$J = \sum_{o=1}^{n-1} \gamma_o Var(y_o) \tag{1}$$

could be maximized (or minimized) to determine the principal components of the input, whose covariance matrix is given by Σ_x. The scalar gains γ_o are chosen such that $\gamma_1 > \gamma_2 > \ldots > \gamma_{n-1} > 0$. The input data x is assumed to be zero-mean.

Theorem 1: For the constrained network where the weight matrix R is an orthonormal matrix, the function J has a stationary point if and only if the rows of R consist of all the eigenvectors of Σ_x.
Proof: In [12]

This theorem 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.

For practical implementation, SIPEX-G parameterizes the orthonormal matrix R using Givens rotations. There exist a unique set of Givens angles that characterize each and every orthonormal matrix. In n-dimensions, the orthonormal matrix R can be decomposed into a cascade of rotation matrices as

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

where R^{ij}, is given by an identity matrix whose four entries at the intersection of i^{th} and j^{th} rows with i^{th} and j^{th} columns are modified as follows: The $(i,i)^{th}$ and $(j,j)^{th}$ entries are $\cos\theta_{ij}$, and the $(i,j)^{th}$ and $(j,i)^{th}$ entries are $-\sin\theta_{ij}$ and $\sin\theta_{ij}$, respectively [9].

Writing the cost function explicitly in terms of the entries of the rotation and covariance matrices yields (3), where R_{ij} is the $(i,j)^{th}$ entry of the rotation matrix R, which is constructed using the Givens angles as shown in (2).

$$J = \sum_{o=1}^{n-1} \gamma_o Var(y_o) = \sum_{o=1}^{n-1} \gamma_o \sum_{i=1}^{n} \sum_{j=1}^{n} R_{oi} R_{oj} \Sigma_{x,ij} \tag{3}$$

In practice, since the covariance matrix of the input data is not known, it has to be estimated, perhaps recursively, from the samples acquired up to that point in time. In non-stationary environments, a forgetting factor can be incorporated into this recursion to account for changing statistics. This approach of estimating the output variances from the input covariance matrix and the current PCA weight matrix results in a stable, robust, and fast algorithm. This algorithm is as follows:

Algorithm: SIPEX-G

Step 1. Initialize Givens angles, $\Theta = [\theta_{pq}]$.

Step 2. Use the first $N>n$ samples of the input data to obtain an unbiased estimate to the covariance matrix Σ_x.

$$R_x = \frac{1}{N-n}\sum_{k=1}^{N} x_k x_k^T \tag{4}$$

Step 3. If the input data is WSS, use the following recursion to update the covariance estimate with every new sample.

$$R_x(k) = \frac{k-n-1}{k-n} R_x(k-1) + \frac{1}{k-n} x_k x_k^T \tag{5}$$

Step 4. Evaluate the gradient of the cost function with respect to the Givens angles from

$$\frac{\partial J}{\partial \theta_{kl}} = \sum_{o=1}^{n-1}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(R_{oi}\frac{\partial R_{oj}}{\partial \theta_{kl}} + \frac{\partial R_{oi}}{\partial \theta_{kl}} R_{oj} \right) R_{x,ij} \tag{6}$$

Step 5. Update the Givens angles using gradient ascent.

$$\Theta(k+1) = \Theta(k) + \eta \frac{\partial J}{\partial \Theta} \tag{7}$$

Step 6. Go back to step 3 and continue until convergence.

3 MONTE CARLO ANALYSIS

In this section, we compare the convergence properties of the SIPEX-G algorithm with those of Sanger's rule, APEX, and Xu's LMSER. Especially, we focus on the convergence time (in terms of iterations) and number of samples necessary for convergence, and the average estimation accuracy of the eigenvector directions after convergence.

To this purpose, 100 Monte Carlo runs using randomly chosen 3-dimensional Gaussian data with randomly chosen covariance matrices are performed. For a given data set, each algorithm is started from the same initial condition (identity matrix). The step sizes of SIPEX-G, Sanger's rule and LMSER are set such that the first component converges in less than 500 iterations and the oscillations are small. For APEX, the sequential approach with the proposed optimal adaptive step size is utilized. The gain vectors for each output in SIPEX-G and LMSER are both set to [3 2 1].

Fig. 1 and Fig. 2 summarize the results of these Monte Carlo runs. The average eigenvector direction estimation error for a single run (10000 iterations) is defined as the average of 3 angles between the actual eigenvectors and their corresponding estimates also averaged over the last 1000 iterations (samples) for smoothness. Notice in Fig. 1 that in almost all cases, SIPEX-G achieved an average direction error less than one degree, whereas the error levels of the other algorithms were mostly on the order of degrees. As for the convergence time, two values are defined: 10^0-threshold convergence time and 1^0-threshold convergence time. These are defined as the latest time instants (iteration index) after which direction estimation errors for all eigenvectors are below the threshold 99% of the time. The histograms for 10^0- and 1^0-convergence times for SIPEX-G and LMSER are given in Fig. 2. Notice that SIPEX-G crosses the 10^0-threshold very fast, almost always in less than 200 iterations; whereas, LMSER requires more time to achieve this accuracy level.

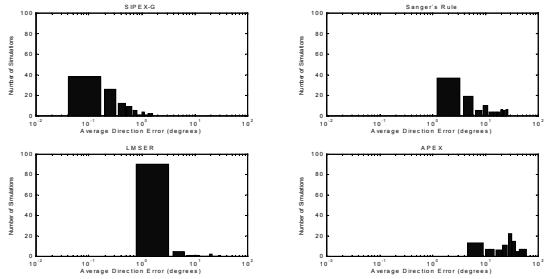

Figure 1. Histograms of average eigenvector direction estimation errors for SIPEX-G, Sanger's rule, LMSER, and APEX.

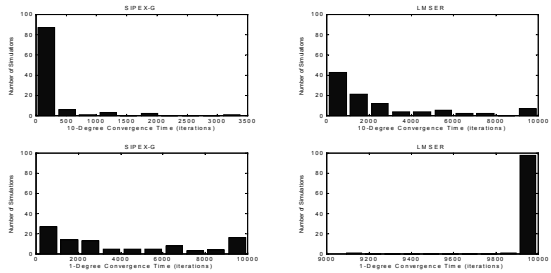

Figure 2. Histograms of 10-degree and 1-degree threshold convergence times for SIPEX-G and LMSER.

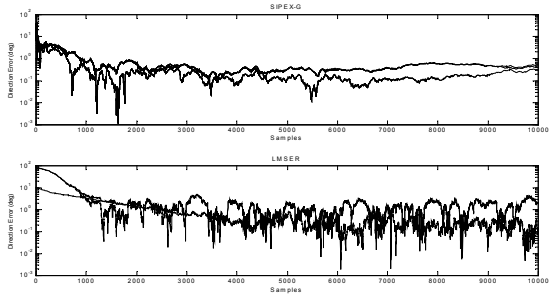

Figure 3. Average direction estimation errors for three eigenvectors using SIPEX-G and LMSER.

	Direction Error (deg)	10^0-Conv. Achieved (iter)	1^0-Conv. Achieved (iter)
SIPEX-G	0.37 ± 0.35	181 ± 474	4059 ± 3420
LMSER	2.50 ± 4.08	2374 ± 2684	Sometimes
SANGER	8.93 ± 7.90	8370 ± 2593	Never
APEX	30.10 ± 14.89	Sometimes	Never

Table 1. Summary of convergence properties of the algorithms in terms of their average eigenvector direction estimation errors and convergence times

The step size of LMSER could be further increased for speed, but this would also increase the magnitude of oscillations from sample to sample, thus diminish the consistency of the estimations.

To summarize these results, we present in Table 1, the averages and the standard deviations of these quantities over the 100 Monte Carlo runs. Notice that SIPEX-G achieves consistently a high level of accuracy in estimating the eigenvectors in a very short time, which also means data efficiency as well as high convergence speed. Since SIPEX-G is the only algorithm that achieves 1^0-convergence sufficiently often, only its convergence time is provided in the table.

For convenience, we also provide in Fig. 3 a sample convergence plot for both SIPEX-G and LMSER (the two simulations are for different data sets with different covariances, therefore should not be compared directly).

In this section, the performance of the SIPEX-G algorithm is analyzed and compared with those of benchmark PCA algorithms, namely, Sanger's rule, APEX, and Xu's LMSER. Monte Carlo simulations using synthetic data had shown that LMSER is the closest competitor of SIPEX-G, yet its performance in terms of speed and accuracy is still an order of magnitude lower. SIPEX-G owes its superior performance to its robust output variance estimation through the input covariance and to its restricted weight matrix expressed through Givens rotations.

4 SIPEX-G IN REAL WORLD PROBLEMS

We have seen in the preceding section that SIPEX-G outperforms the competing PCA algorithms in performance. Encouraged by these results, in this section, we will employ SIPEX-G in two real-world problems where the traditional PCA algorithms are commonly utilized, direction of arrival estimation and subspace Wiener filtering.

4.1 Subspace Wiener Filtering

In general, subspace Wiener filtering refers to assisting the traditional LMS training algorithm with PCA algorithms in order to speed up convergence [Haykin,Farhang]. Conventionally, in ADALINE training (note that FIR filters are special cases corresponding to an input vector formed by a tap-delay-line) PCA is first applied to sphere the input vectors to the actual LMS algorithm does not suffer from the eigenspread of the input vector's covariance matrix. Alternatively, PCA could be used in determine the Wiener solution for the linear adaptive system of the form $y=w^{\mathrm{T}}x$ where x is the input vector, w is the weight vector, and y is the system output for which the desired signal d is assigned. Wiener solution for the weight vector is the optimal in terms of the mean-square-error (MSE) criterion and it is given by $w_* = R_x^{-1}P$, where $R_x=E[xx^{\mathrm{T}}]$ and $P=E[dx]$. Consider the eigenvector equation for R: $R_x=Q_x\Lambda_xQ_x^{\mathrm{T}}$ yields $R_x^{-1}= Q_x\Lambda_x^{-1}Qx^{\mathrm{T}}$, where Q_x is the orthonormal eigenvector matrix, which can be determined by applying an on-line PCA algorithm on the input vector samples and Λ_x is the diagonal eigenvalue matrix, which can be determined from the output variances of the PCA operation. An estimate of the crosscorrelation vector P also can be obtained using the recursive formula $P(k)= ((k\text{-}1)/k)P(k\text{-}1)+(1/k)d_kx_k$. Combining these on-line estimates of R_x^{-1} and P, we can obtain an accurate estimate of the Wiener solution for the weights in an on-line fashion without having to invert the covariance matrix.

In order to demonstrate the speed and accuracy of the proposed SIPEX-G algorithm, we performed 100 Monte Carlo simulations to obtain the Wiener solution of a 3-tap ADALINE using 10000 samples of input-desired data using the procedure described above. The input covariance matrix, the optimal solution and the samples are selected randomly for each run. For comparison, SIPEX-G, LMSER and simple LMS are applied to the same data sets, all with a learning rate of 10^{-3}. Fig. 4 illustrates a typical convergence plot for these three algorithms. The normalized distance to the Wiener solution is defined as the ratio of the norm of the weight vector error (with respect to the Wiener solution that can be obtained from all the samples seen by the algorithm so far as the reference) to the norm of the Wiener solution for the weight vector. Notice that SIPEX-G converges to the Wiener solution fast and accurately compared to LMSER. Regular LMS algorithm, on the other side, converges slow but steadily towards the solution. LMSER, however, cannot converge to the desired solution as accurately as the other two algorithms.

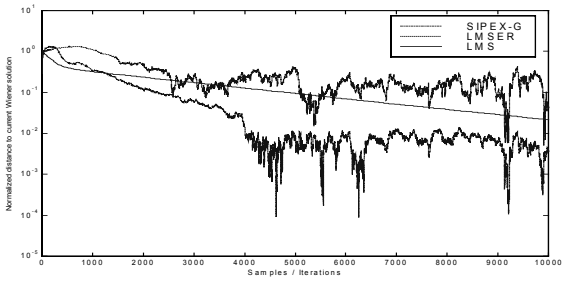

Figure 4. Typical convergence plot of SIPEX-G, LMSER and LMS to the Wiener solution in an on-line ADALINE training scenario.

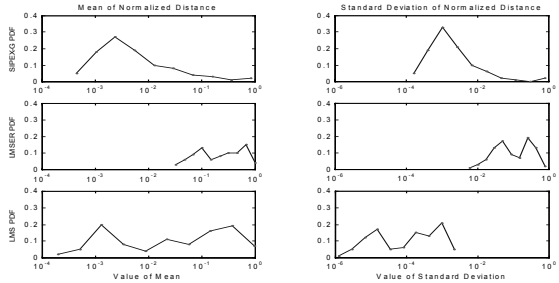

Figure 5. Pdf estimates of mean (first column) and standard deviation (second column) of normalized distance to the Wiener solution over 100 Monte Carlo runs.

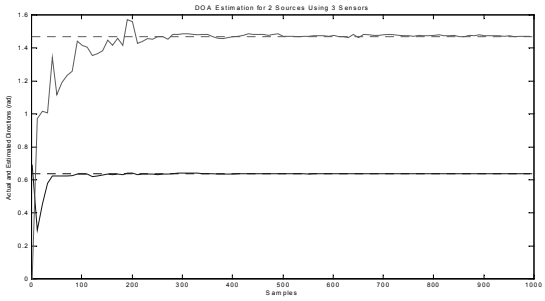

Figure 6. Actual and estimated DOA for a 2-source 3-sensor case with 20dB SNR.

Fig. 5 shows the histograms of the mean and the standard deviation of normalized distance over the last 200 iterations of each Monte Carlo run. Observe that SIPEX-G achieves a small error in estimating the Wiener solution with a small variance, whereas LMSER canoot perform as well in general. LMS, on the other hand can be susceptible to the eigenspread of the input covariance matrix, resulting in a wider spread in the pdf of final accuracy achieved by this algorithm. Since LMS converges smoothly towards the optimal weight vector, its standard deviation over the last 200 iterations of each run is very small compared to the PCA based algorithms.

4.2 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 [13], ESPRIT [14], MUSIC [15], and Min-Norm [16] are examples to these approaches to the DOA problem. These methods all require the eigendecomposition of the covariance matrix of the signal, yet the alternative SWEDE [17], achieves subspace estimation without eigendecomposition. In this section, we will focus on the MUSIC algorithm, which basically uses Sanger's rule for the

eigendecomposition task. We will replace that 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.

$$x_k = D(\Phi)s_k + v_k \tag{8}$$

where $D(\Phi) = [d(\phi_1)...d(\phi_m)]$ is the nxm steering matrix with Φ denoting the vector of direction angles for the sources [15,17]. The random vectors x_k, s_k, and v_k are nx1, mx1, and nx1 complex Gaussian distributed respectively. Under this formulation, the covariance matrix of x, the received signal vector at the sensors, can be expressed as

$$R_x = E[x_k x_k^H] = D(\Phi)R_s D(\Phi)^H + \sigma^2 I \tag{9}$$

assuming uncorrelated, equal-power noise on each sensor. In order to solve for the complex eigenvectors of R_x, we define

$$R_c = \begin{bmatrix} \mathrm{Re}[R_x] & -\mathrm{Im}[R_x] \\ \mathrm{Im}[R_x] & \mathrm{Re}[R_x] \end{bmatrix},\ e_c = \begin{bmatrix} \mathrm{Re}[e_x] \\ \mathrm{Im}[e_x] \end{bmatrix} \tag{10}$$

where R_c is a real matrix twice the size of R_x, and with the eigenvectors as shown in (10), in terms of the complex eigenvectors of the R_x. R_c can be computed from the samples by

$$R_c = E[x_c x_c^T],\ x_c = \begin{bmatrix} \mathrm{Re}[x] & -\mathrm{Im}[x] \\ \mathrm{Im}[x] & \mathrm{Re}[x] \end{bmatrix} \tag{11}$$

When the number of sources is known, the task of DOA estimation reduces to finding the n-m minor components of the covariance matrix R_x and determining the m minima of the cost function

$$f(\phi) = d(\phi)^H (W_v W_v^H) d(\phi) \tag{12}$$

where $W_v = [e_{m+1}...e_n]$ 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 in 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. 6. Notice that, the PCA problem that is solved in this problem is 2xn=6 dimensional with three eigenvalues each of multiplicity two.

5 CONCLUSIONS

PCA plays a central role in statistical signal processing and in the last two decades many on-line algorithms have been proposed in the literature to determine the eigenvectors of the input covariance matrix. These conventional algorithms, however, do not take advantage of the well-known fact that the solution sought dwell in the set of orthonormal matrices. Due to this, most of these algorithms employ deflation, which nothing more than the Gram-Schmidt orthogonalization, and Oja's 1st order approximation to vector normalization to accomplish orthonormality. The proposed SIPEX-G algorithm, on the other hand, addresses the question of whether there is an on-line PCA algorithm that avoids deflation and converges to all the desired principal components simultaneously. Exploiting the Givens rotations to parameterize the weight matrix, SIPEX-G guarantees that the eigenvector estimates form an orthonormal set at all times, saving time by avoiding the orthonormalization process. Furthermore, it uses a robust estimate of the output variances that employs directly the input covariance matrix, therefore achieves high accuracy. SIPEX-G, however, is still a gradient algorithm, thus its performance is susceptible to poor choice of the step size.

Extensive Monte Carlo simulations performed and presented in this paper demonstrated that, nevertheless, SIPEX-G is still superior to conventional benchmark gradient-based PCA algorithms Sanger's rule and APEX and even to the LMSER algorithm, which also outperforms the traditional PCA approaches. We have also demonstrated the high performance of SIPEX-G in two real world problems, namely on-line direction of arrival estimation using subspace methods and subspace Wiener filtering.

Future work will be directed towards the development of a fixed-point algorithm to further increase the speed and efficiency of the algorithm and also to reduce the computational burden.

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

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, May 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, March 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, May 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 Unsupevised Learning in a Single Layer Linear Feedforward Neural Network," Neural Networks, vol. 2, no. 6, pp. 459-473, 1989.

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

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

[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," submitted to ICASSP'02, Nov 2001.

[13] 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, Nov. 1994.

[14] 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, Dec. 1994.

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

[16] R. Kumaresan, D.W. Tufts, "Estimating the Angles of Arrival of Multiple Source Plane Waves," IEEE Transactions on Aerospace Electrical Systems, vol. 19, pp.134-149, Jan. 1983.

[17] 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, Sep. 1994.