DIXIEME COLLOQUE SUR LE TRAITEMENT DU
SIGNAL ET SES APPLICATIONS

NICE du 20 au 24 MAI 1985

STATISTICAL ANALYSIS AND IMPLEMENTATION OF SINGULAR VALUE PREPROCESSORS

Leon H. Sibul

APPLIED RESEARCH LABORATORY, The Pennsylvania State Univeristy
P. O. Box 30, State College, PA 16804 USA

RESUME

Afin d'éviter d'importantes erreurs numériques ou une convergence trop lente des algorithmes itératifs, les processeurs pour antennes adaptatives doivent corriger le mauvais conditionnement qui apparait fréquemment en pratique. Cela peut être realisé à l'aide d'un preprocesseur à vecteur singulier ou à vecteur propre, qui transforme le vecteur en sortie de l'antenne en un vecteur de dimension plus faible. Ces transformations changent les processeurs adaptatifs mal conditiennés en processeurs bien conditionnés et réduisent la dimension du processeur pour antennes à la dimension optimale nécessaire pour exploiter le nombre de sources directionnelles distinctes qui composent l'antenne. D'autre part, cette méthode basée sur la décomposition en valeuer singulière (SVD) permet d'identifier le nombre de sources. Il y a cependant deux difficultés dans l'implémentation de ces techniques: le manque de méthodes valables du point de vue statistique pour mettre une limite inferieure aux valeurs singulières et limiter la complexité des calculs sur ordinateur.

Dans la pratique, pour utiliser la technique de prétraitement d'un vecteur singulier on d'un vecteur propre, ie faut appliquer ces décompositons à des matrices de covariance estimées. Par conséquence, les valeurs singulières (propres) et les vecteurs singuliers (propres) sont des valeurs aléatoires et il faut analyser les erreurs de probabileté de ces quantités. Dans cet article, de simples limites supérieures pour les erreurs au moindre carré des valeurs propres estimées et des vecteurs propres sont derivées. Ces limites supérieures peuvent facilement être calculées par la matrice de covariance des erreurs d'estimation. Nous nous sommes servis de ces limites supèrieures pour analyser les performances des formeurs de faisceaux qui utilisent les preprocesseurs de vecteurs propres.

Une autre difficulté dans l'application de la décomposition en valeur singulière (SVD) au traitement du signal en temps réel est sa complexité sur ordinateur. Cette difficulté peut être aplanie si des structures specialisées de technique d'intégration sur une large échelle (VLSI) sont utilisées pour la décomposition en valeur singulière. L'idée essentielle de l'implémentation de l'intègration á large échelle de la décomposition á valeur singuliére et a valeur propre est basée sur le fait que ces décompositions peuvent être effectuées grace à des rotations, dans le plan, de séquences adéquates.

SUMMARY

To avoid large numerical errors or slow convergence of iterative algorithms, adaptive array processors must correct ill-conditioning that can frequently arise in practice. This can be accomplished by a singular vector or an eigenvector preprocessor which transforms the array output vector to a lower dimensional vector. The transformations transform the ill-conditioned adaptive processors into well-conditioned processors, and reduce the array processor's dimensionality to optimum dimensionality needed to process the number of distinct dimensionality to optimum dimensionality needed to process the number of distinct directional sources that impinge on the array. As a by-product, this SVD-based processing technique identifies the number of sources. There are two difficulties in implementation of these techniques: lack of statistically valid methods for thresholding singular values and computational complexity.

To use either a singular vector or an eigenvector preprocessing technique in practice, one must apply these decompositions to estimated covariance matrices. Therefore, the singular (eigen) values and singular (eigen) vectors are random variables and one needs to perform probabilistic error analyses of these quantities. In this paper we derive simple upper bounds for the mean square errors of estimated eigenvalues and vectors. These upper bounds can be easily calculated from the covariance matrix estimation errors. We have used these upper-bound expressions to analyze the performance of adaptive beamformers which use eigenvector preprocessors.

Another difficulty in application of SVD to real-time signal processing is its computational complexity. This difficulty can be overcome if specialized VLSI structures are used for SVD. The essential idea for the VLSI implementation of singular value and eigenvalue decomposition is based on the observation that these decompositions can be achieved by appropriate sequences of planar rotations.

INTRODUCTION

In many signal processing applications such as linear prediction, maximum likelihood estimation and adaptive beamforming require inversion of an estimated covariance matrix which may be ill-conditioned.[1,2,12] This can cause large numerical errors or slow convergence of adaptive algorithms. To avoid these problems, adaptive signal processors can use either singular vector or eigenvector preprocessors.[1] Singular vector preprocessors have more general applications than eigenvector preprocessors, but in adaptive beamforming application it is sufficient to consider eigenvector preprocessors since the array output covariance matrices are Hermitian. This avoids some extraneous complications. With some additional complications the results of this paper can be extended to the singular value decomposition. Eigenvector preprocessor $\underline{U}_r^H$ transforms an array output vector $\underline{x}$ to a lower dimensional vector $\underline{y}$:

$$\underline{y} = \underline{U}_r^H \underline{x} \tag{1}$$

where $\underline{U}_r$ is the submatrix of eigenvectors which correspond to the r largest eigenvalues of the array output covariance matrix $\underline{\Gamma}$.[1] The covariance matrix $\underline{\Gamma}$ is given by

$$\underline{\Gamma} = E\{\underline{x}(i)\, \underline{x}^H(i)\} . \tag{2}$$

where subscript H denotes Hermitian transpose, E{ } expected value and $\underline{x}(i)$ is the array output vector. The vector $\underline{x}(i)$ is a vector of random variables with components

$$x_m(i) = \sum_{j=1}^{r} s_j(t_i - \tau_{mj}) + \eta_m(i) . \tag{3}$$

where s_j is the signal due to the j-th source, τ_{mj} is the propagation delay of the directional signal from sensor to sensor, and $\eta_m(i)$ is the sensor noise which is uncorrelated from the directional sources.[1] Both the noise terms and the directional signal terms are zero mean stochastic processes.

The array output covariance matrix is decomposed

$$\underline{\Gamma} = \underline{U}\, \underline{\Lambda}\, \underline{U}^H \tag{4}$$

where $\underline{\Lambda}$ is a diagonal matrix of non-negative eigenvalues of $\underline{\Gamma}$, and $\underline{U}$ is a matrix whose columns are the eigenvectors of $\underline{\Gamma}$.[3] Optimum array weights can be determined from this decomposition.[1] In practice one must determine the eigendecomposition and the optimum array weights from an estimated covariance matrix $\hat{\underline{\Gamma}}$ which is a sum of the true covariance matrix $\underline{\Gamma}$ and estimation error $\varepsilon\underline{\Delta}$.

In this paper, we determine expressions for the means and variances of the eigenvectors and eigenvalues of $\hat{\underline{\Gamma}}$, and to find an expression for the increase of mean square error caused by random errors in computing the eigenvalue matrix inverse. We consider the primary source of errors to be the noisy estimates of the array output covariance matrix. In this paper true or unperturbed covariances, eigenvectors, eigenvalues and array weights are considered deterministic quantities and the corresponding error terms or perturbed quantities are random variables. The analysis is restricted to small perturbation.

ESTIMATED COVARIANCE MATRIX

The output of an adaptive beamformer can be expressed in matrix form as

$$\underline{A}\, \underline{w} \cong \underline{b} \tag{5}$$

where $\underline{A}$ is a k x m data matrix, $\underline{w}$ is m x 1 weight vector and $\underline{b}$ is k x 1 desired signal vector. k denotes the number of data samples and m the number of sensors.[1] These equations are overspecified and inconsistent and only have a solution in the least squares sense.[4] That is, the optimum array weights are the weights that minimize $\|\underline{A}\, \underline{w} - \underline{b}\|^2$. Usually the number of sensors is very much smaller than the number of data samples k. In this case the least squares solution can be expressed in terms of generalized inverse $\underline{A}^+$

$$\underline{w} = \underline{A}^+ \underline{b} \tag{6}$$

where

$$A^+ = (A^H A)^{-1} A^H . \tag{7}$$

We observe that

$$\underline{A}^H \underline{A} = \sum_{i=1}^{k} \underline{x}^*(i)\, \underline{x}^T(i) = k\, \hat{\underline{\Gamma}}^* \tag{8}$$

and

$$\underline{A}^H \underline{b} = \sum_{i=1}^{k} \underline{x}^*(i)\, \underline{b(i)} = k\, \hat{\underline{R}}_{\underline{b}\,\underline{x}} \tag{9}$$

where $\underline{x}(i)$ is the data vector at time i, $\hat{\underline{\Gamma}}$ is the estimated covariance matrix and $\hat{\underline{R}}_{\underline{b}\,\underline{x}}$ is the estimated cross-covariance vector. The estimated covariance matrix $\hat{\Gamma}$ is an unbiased estimate of covariance matrix Γ, that is

$$E\{\hat{\underline{\Gamma}}\} = \frac{1}{k} \sum_{i=1}^{k} E\{\underline{x}(i)\, \underline{x}^H(i)\} = \underline{\Gamma} . \tag{10}$$

We will use this property in the next section.

ERROR ANALYSIS OF EIGENVALUE ESTIMATES

In adaptive beamformers estimated eigenvalues are used for two functions: to determine the number of spatially compact strong sources and to compute optimum array weighting coefficients. The first function is mathematically equivalent to the determination of a set of distinct eigenvalues of Γ which exceed the eigenvalues due to omnidirectional background noise. In both applications, the estimated covariance matrix $\hat{\underline{\Gamma}}$ is decomposed as follows

$$\hat{\underline{\Gamma}} = \underline{\Gamma} + \varepsilon\underline{\Delta} = \hat{\underline{U}}\, \hat{\Lambda}\, \hat{\underline{U}}^H \tag{11}$$

where $\underline{\Gamma}$ is the true covariance matrix, $\varepsilon\underline{\Delta}$ is the error in estimated covariance matrix, $\hat{\Lambda}$ and $\hat{\underline{U}}$ are the estimated matrices of eigenvalues and eigenvectors,[1] and ε is a perturbation parameter. The eigenvalue/eigenvector decomposition of the true covariance matrix is given by

$$\underline{\Gamma} = \underline{U}\ \underline{\Lambda}\ \underline{U}^{H} . \qquad (12)$$

The eigenvalues and eigenvectors (columns of matrix $\underline{U}$), satisfy the equation

$$\underline{\Gamma}\ \underline{u}_k = \lambda_k\ \underline{u}_k \quad k = 1, \ldots m . \qquad (13.a)$$

$$\hat{\underline{\Gamma}}\ \hat{\underline{u}}_k = \hat{\lambda}_k\ \hat{\underline{u}}_k \quad k = 1, \ldots m . \qquad (13.b)$$

We follow a perturbation analysis that Kammler has used to compute errors in singular vectors that have been computed with singular value decomposition (SVD).[6] We use this approach to compute perturbations in eigenvalues and eigenvectors. Following Kammler, we write

$$\begin{aligned}\underline{u}_k(\varepsilon) = \underline{u}_k &+ (\alpha_{11}\ \underline{u}_1 + \ldots + \alpha_{1m}\ \underline{u}_m)\ \varepsilon \\ &+ (\alpha_{21}\ \underline{u}_1 + \ldots + \alpha_{2m}\ \underline{u}_m)\ \varepsilon^2 + \\ &+ \ldots\end{aligned} \qquad (14)$$

and

$$\lambda_k(\varepsilon) = \lambda_k + \lambda_{k1}\varepsilon + \lambda_{k2}\varepsilon^2 \qquad (15)$$

Substituting Eq. (14) and (15) into Eq. (13.b) we have:

$$\begin{aligned}&(\underline{\Gamma} + \varepsilon\underline{\Delta})[\underline{u}_k + \varepsilon(\alpha_{11}\underline{u}_1 + \ldots \alpha_{1m}\underline{u}_m) + \varepsilon^2(\alpha_{21}\underline{u}_1 \ldots) \\ &+ \ldots] = (\lambda_k + \varepsilon\lambda_{k1} + \varepsilon^2\lambda_{k2} + \ldots) \\ &[\underline{u}_k + \varepsilon(\alpha_{11}\underline{u}_1 + \ldots + \alpha_{1m}\underline{u}_m) + \varepsilon^2(\alpha_{21}\underline{u}_1 + \ldots + \\ &\alpha_{2m}\underline{u}_m) + \ldots] .\end{aligned} \qquad (16)$$

Performing the indicated multiplication and equating equal powers of ε we have

$$\varepsilon^0: \quad \underline{\Gamma}\ \underline{u}_k = \lambda_k \underline{u}_k \qquad (17)$$

$$\varepsilon^1: \quad \underline{\Delta}\ \underline{u}_k + \underline{\Gamma}(\alpha_{11}\underline{u}_1 + \ldots + \alpha_{1m}\underline{u}_m) = \lambda_{k1}\underline{u}_k + \lambda_k(\alpha_{11}\underline{u}_1 + \ldots + \alpha_{1m}\underline{u}_m) \qquad (18)$$

$$\begin{aligned}\varepsilon^2: \quad &\underline{\Delta}(\alpha_{11}\underline{u}_1 + \ldots \alpha_{1m}\underline{u}_m) + \underline{\Gamma}(\alpha_{21}\underline{u}_1 + \ldots + \alpha_{2m}\underline{u}_m) \\ &= \lambda_{k2}\underline{u}_k + \lambda_{k1}(\alpha_{11}\underline{u}_1 + \ldots + \alpha_{1m}\underline{u}_m) \\ &+ \lambda_k(\alpha_{21}\underline{u}_1 + \ldots + \alpha_{2m}\underline{u}_m) .\end{aligned} \qquad (19)$$

We premultiply Eq (18) by $\underline{u}_k^H$, then by orthogonality of eigenvectors, that is

$$\underline{u}_k^H \underline{u}_i = \delta_{ki} \qquad (20)$$

and Eq (13) we have an expression for first perturbation term of the eigenvalue

$$\lambda_{k1} = \underline{u}_k^H\ \underline{\Delta}\ \underline{u}_k . \qquad (21)$$

The expectation of λ_{k1} is

$$E\{\lambda_{k1}\} = \underline{u}_k^H\ E\{\underline{\Delta}\}\ \underline{u}_k = 0 \qquad (22)$$

because we showed that Eq (10) is an unbiased estimator of $\underline{\Gamma}$. The variance of λ_{k1} is

$$\text{var}\ \lambda_{k1} = \underline{u}_k^H\ E\{\underline{\Delta}\ \underline{\Delta}^H\}\ \underline{u}_k \qquad (23)$$

The coefficients α_{1i} can be obtained by premultiplying eq (18) by $\underline{u}_i^H$ and then using eq (13) and (20)

$$\alpha_{1i} = -(\underline{u}_i^H\ \underline{\Delta}\ \underline{u}_k)/(\lambda_i - \lambda_k) . \qquad (24)$$

$i \neq k$.

The α_{1k} coefficient can be computed from the normalization condition of $\underline{u}_k(\varepsilon)$:[6]

$$\begin{aligned}\|\underline{u}_k(\varepsilon)\|_2^2 &= \|(1 + \alpha_{1k}\varepsilon)\ \underline{u}_k + \varepsilon \sum_{i=1}^{m} \alpha_{1i}\underline{u}_i + 0(\varepsilon^2)\|_2^2 \\ &= 1 + 2\ \alpha_{1k}\varepsilon + 0(\varepsilon^2) = 1 ,\end{aligned} \qquad (25)$$

therefore

$$\alpha_{1k} = 0 \qquad (26)$$

Hence, the first order perturbation of eigenvectors of $\underline{\Gamma}$ due to the perturbation by $\varepsilon\underline{\Delta}$ is given by

$$\begin{aligned}\underline{u}_k(\varepsilon) &= \underline{u}_k + \varepsilon(\alpha_{11}\underline{u}_1 + \ldots + \alpha_{1m}\underline{u}_m) \\ &= \underline{u}_k - \varepsilon \sum_{i=k}^{m} \underline{u}_i(\underline{u}_i^H \underline{\Delta}\ \underline{u}_k)/(\lambda_i - \lambda_k) \\ &= \underline{u}_k - \varepsilon\underline{c}\end{aligned} \qquad (27)$$

where vector $\underline{c}$ is defined by the sum that follows ε. This equation clearly shows that an eigenvalue which is close to eigenvalue λ_k causes serious instability in determination of the eigenvector $\underline{u}_k(\varepsilon)$. Since $E\{\underline{\Delta}\} = 0$, $E\{\underline{u}_k(\varepsilon)\} = \underline{u}_k$, i.e., $\underline{u}_k(\varepsilon)$ is an unbiased estimator of $\underline{u}_k$. A measure of error in $\underline{u}_k(\varepsilon)$ is

$$E\{\|\underline{u}_k(\varepsilon) - \underline{u}_k\|_2^2\} = \varepsilon^2\ E\{\underline{c}^H\underline{c}\} . \qquad (28)$$

Computation of higher order perturbation of eigenvalues and eigenvectors is messy, it is therefore instructive to examine these errors in the light of some well-known error bounds. Weyl's inequality for the eigenvalues of m x m Hermitian matrices is

$$\lambda_k(\underline{\Gamma}) + \lambda_k(\underline{\Delta}) \leq \lambda_k(\underline{\Gamma} + \underline{\Delta}) \leq \lambda_k(\underline{\Gamma}) + \lambda_1(\underline{\Delta})$$

$$k = 1, 2, \ldots m \qquad (29)$$

where $\lambda_1(\Delta)$ is the largest eigenvalue of $\underline{\Delta}$.[7,8] Bounds for expected eigenvalues ($\underline{\Delta}$ is a random matrix and $\underline{\Gamma}$ is a deterministic matrix) are

$$\lambda_k(\Gamma) + E\{\lambda_k(\underline{\Delta})\} \leq E\{\lambda_k(\underline{\Gamma} + \underline{\Delta})\} \leq \lambda_k(\underline{\Gamma}) + E\{\lambda_1(\underline{\Delta})\} \qquad (30)$$

Since

$$E\{\lambda_k(\underline{\Delta})\} \geq 0$$

$$\lambda_k(\underline{\Gamma}) \leq E\{\lambda_k(\underline{\Gamma} + \underline{\Delta})\} \leq \lambda_k(\underline{\Gamma}) + E\{\lambda_1(\underline{\Delta})\}. \qquad (31)$$

These are interesting error bounds, unfortunately it is difficult to compute $E\{\lambda_1(\Delta)\}$.

A bound of mean square error of the eigenvalue estimate can be computed using inequalities (29) and (30)

$$E\{[\lambda_k(\underline{\Gamma} + \underline{\Delta}) - \lambda_k(\underline{\Gamma})]^2\}$$

$$= E\{\lambda_k^2(\underline{\Gamma} + \underline{\Delta}) - 2\lambda_k(\underline{\Gamma} + \underline{\Delta})\lambda_k(\underline{\Gamma}) + \lambda_1^2(\underline{\Gamma})\}$$

$$\leq 2\lambda_k^2(\underline{\Gamma}) + 2\lambda_k(\underline{\Gamma})\, E\{\lambda_1(\underline{\Delta})\} + E\{\lambda_1^2(\underline{\Delta})\}$$

$$- 2\lambda_k(\underline{\Gamma})\, E\{\lambda_k(\underline{\Gamma} + \underline{\Delta})\}$$

$$= 2\lambda_k(\underline{\Gamma})\left(E\{\lambda_1(\Delta)\} - E\{\lambda_k(\underline{\Delta})\}\right) + E\{\lambda_1^2(\underline{\Delta})\} \quad (32)$$

If the expectations of the eigenvalues of the error matrix are equal, the mean square error bound becomes:

$$E\{[\lambda_k(\underline{\Gamma} + \underline{\Delta}) - \lambda_k(\underline{\Gamma})]^2\} \leq E\{\lambda_1^2(\underline{\Delta})\} \quad (33)$$

EXCESS MEAN SQUARE ERROR

The weight vector which minimizes the mean square error is given by

$$\underline{w}_o = \underline{\Lambda}_{ro}^{-1} R_{by} , \quad (34)$$

and the corresponding mean square error is

$$\xi_{min} = |b|^2 - \underline{R}_{yb}^H \underline{\Lambda}_{ro}^{-1} \underline{R}_{yb} . \quad (35)$$

When the actual weight vector w deviates from the optimum weight vector $\underline{w}_o$, the actual mean square error is

$$\xi = \xi_{min} + E\{\underline{w}_\varepsilon^H \underline{R}_{yy} \underline{w}_\varepsilon\} = \xi_{min} + \xi_\varepsilon \quad (36)$$

where

$$\underline{w}_\varepsilon = \underline{w} - \underline{w}_o . \quad (37)$$

The last term in Eq (36), ξ_ε, is called the excess mean square error.[9] In this section we are particularly interested in computing the excess mean square error which is caused by errors in determination of $\underline{\Lambda}_r^{-1}$. To avoid extraneous complications, due to effects of which are of no direct interest in this section, we assume that $\hat{R}_{by} \approx R_{by}$. With this assumption and by use of results from Appendix A, the we have an expression for the error in the weight vector

$$\underline{w}_\varepsilon = (\underline{\Lambda}_r^{-1} - \underline{\Lambda}_{ro}^{-1})\, \underline{R}_{by}$$

$$\approx (\underline{\Lambda}_{ro}^{-1} - \underline{\Lambda}_{ro}^{-1} \underline{\Lambda}_{r\varepsilon} \underline{\Lambda}_{ro}^{-1} - \underline{\Lambda}_{ro}^{-1})\, \underline{R}_{by}$$

$$= -\underline{\Lambda}_{ro}^{-1} \underline{\Lambda}_{r\varepsilon} \underline{\Lambda}_{ro}^{-1} \underline{R}_{by} \quad (38)$$

and the excess mean square error

$$\xi_\varepsilon = E\{w_\varepsilon^H R_{yy} w_\varepsilon\} = E\{w_\varepsilon^H \Lambda_{ro} w_\varepsilon\}$$

$$= \underline{R}_{yb}^H \underline{\Lambda}_{ro}^{-1} E\{\underline{\Lambda}_{r\varepsilon}^2\} \underline{\Lambda}_{ro}^{-2} \underline{R}_{yb} \quad (39)$$

where terms in the matrix $E\{\underline{\Lambda}_{r\varepsilon}^2\}$ can be computed by Eq (23). This is the desired expression for the excess mean square error. It is important to observe that small eigenvalues of $\underline{\Lambda}_{ro}$ result in large contributions to excess mean square error. Therefore, one must preprocess the data to avoid inverting an ill-conditioned matrix.

IMPLEMENTATION OF SINGULAR VALUE AND EIGENVALUE DECOMPOSITION

Computation of singular value and eigenvalue decompositions is computationally very intensive. However, methods exist which allow design of specialized VLSI structures for singular value decomposition. The essential idea is based on the observation that the SVD can be achieved by an appropriate sequence of planar rotations in an appropriate coordinate system. This sequence of rotations converges to the diagonal matrix of singular values. SVD is obtained in the process. This method is a generalization of the method originally used by Jacobi to show that the eigenvalues of a real symmetric matrix are real. The basic operation in Jacobi-like algorithms is the planar coordinate rotation. Coordinate rotations in either circular, linear or hyperbolic coordinate systems can be effected by the CORDIC algorithm or its generalizations. The CORDIC algorithm can be efficiently implemented in fast VLSI structures using only shifts, adds, and local memory. Hence, special computational cells can implement the basic coordinate rotations required for SVD and eigenvalue decomposition. These computational cells can be used in either parallel or pipelined VLSI structures. Details of this method are discussed in a paper by Sibul and Fogelsanger.[11]

CONCLUSIONS

We have shown that eigenvalue estimates are unbiased if they are determined from an unbiased estimate of the covariance matrix. A perturbation analysis shows that closely spaced eigenvalues can cause serious errors in determination of eigenvectors. This point needs careful consideration in design of systems which require accurate knowledge of eigenvectors. An expression for excess mean square error shows that small eigenvalues of the preprocessed covariance matrix result in large contributions to the excess mean square error. This further emphasizes need for careful preprocessing of signals that have ill-conditional covariance matrices.

For simplicity we have analyzed processors which are based on the eigenvector decomposition. Results can be extended to the processor which are based on the singular value decomposition (SVD). Error analysis which is not restricted to the small perturbations needs to be done. Development of statistical methods for thresholding singular values and eigenvalues is in progress.

ACKNOWLEDGMENTS

This work has been supported by ONR under the Fundamental Research Initiatives Program. Many constructive comments by Curtis Caldwell are very much appreciated.

REFERENCES

[1] Sibul, L. H. Application of singular value decomposition to adaptive beamforming, Proc. of 1984 Intern. Conf. on Acoustics, Speech, and Signal Processing, San Diego, CA, pp. 33.11.1-4, (March 1984)

[2] Tufts, D. W. and R. Kumaresan. Estimation of frequencies of multiple sinusoids: making linear prediction perform like maximum likelihood, Proc. of IEEE, 70, pp. 975-989, (1982).

[3] Bellman, R. Introduction to Matrix Analysis, McGraw-Hill, New York (1970).

[4] Lawson, C. L. and R. J. Hanson. Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs, NJ (1974).

[5] Jenkins, G. M. and D. G. Watts. Spectral Analysis and its Applications, Holden-Day, San Francisco (1968).

[6] Kammler, D. W. A perturbation analysis of the intrinsic conditioning of an approximate null vector computed with a SVD, Journal of Computational and Applied Mathematics, 9, pp. 201-204 (1983).

[7] Franklin, J. N. Matrix Theory, Prentice-Hall, Englewood Cliffs, NJ (19680.

[8] Golub, G. H. and C. F. Van Loan. Matrix Computations, The Johns Hopkins University Press, Baltimore, MD (1983).

[9] Widrow, B., et al. Stationary and nonstationary learning characteristics of the LMS adaptive filter, Proc. of IEEE, 64(8), pp. 1151-1162 (1976).

[10] Gohberg, I. and S. Goldberg. Basic Operator Theory, Birkhäuser, Boston (1981).

[11] Sibul, L. H. and A. L. Fogelsanger. Application of Coordinate Rotation Algorithm to Singular Value Decomposition, Proc. 1984 IEEE Int. Symp. on Circuits and Systems, Montreal, Canada, pp. 821-824, (May 1984).

[12] Mermoz, H. Imagerie, Correlation et Models, Ann. Tele Commnic., Vol. 31, pp. 17-36 (1976).

APPENDIX A

In this appendix we obtain an approximate expression for $\underline{\Lambda}_r^{-1}$. Using the classical approximation for an inverse operator[10]

$$\underline{\Lambda}_r^{-1} = (\underline{\Lambda}_{ro} + \underline{\Lambda}_{r\varepsilon})^{-1} = [(I + \Lambda_{r\varepsilon}\, \Lambda_{ro}^{-1})\, \Lambda_{ro}]^{-1}$$

$$= \underline{\Lambda}_{ro}^{-1} (I - \underline{\Lambda}_{r\varepsilon}\, \underline{\Lambda}_{ro}^{-1} + (\underline{\Lambda}_{r\varepsilon}\, \underline{\Lambda}_{ro}^{-1})^2 - \ldots) . \qquad \text{(A-1)}$$

This expansion converges in the mean square sense

$$E\{\| \underline{\Lambda}_{r\varepsilon}\, \underline{\Lambda}_{ro}^{-1} \|_2^2\} < 1 . \qquad \text{(A-2)}$$

This condition must also be satisfied to ensure that the excess mean square error is kept small.