TABLE II
COMPARATIVE PERFORMANCE BETWEEN THE ALTERNATING PROJECTION METHOD AND THE NEW PROPOSED ENHANCED SPATIAL SMOOTHING FOR THE SOURCE AT 5° AND BASED ON 100 INDEPENDENT TRIALS

	RMS Error		
	Number of Snapshots		
Method	50	100	500
Alternating projection	1.9	1.7	0.57
Paper's method	0.0223	0.0153	0.0065

fully coherent sources at $-1°$ and $5°$ in our simulation. The rms errors of the DOA estimate of the source at 5°, for different numbers of snapshots, are compared with those obtained from [8] in Table II.

From these results, it is clear that the signal enhancement method proposed here along with the spatial smoothing can significantly improve the resolution of the eigenspectrum for coherent signals.

V. CONCLUSION

In this correspondence, a new method for enhancement of the signal is proposed which is devised only for coherent signals. The proposed method along with the conventional spatial smoothing is used for estimation of the DOA's of coherent sources. Statistical behavior of the proposed approach, via the Monte Carlo method, demonstrate the significant improvement in the mean-square error and the resolution of the eigenspectrum.

REFERENCES

[1] J. E. Evans, J. R. Johnson, and D. F. Sun, "High resolution angular spectrum estimation techniques for terrian scattering analysis and angle of arrival estimation," in *Proc. 1st ASSP Workshop Spectral Estimation* (Hamilton, Ont., Canada), 1981.
[2] J. E. Evans, J. R. Johnson, and D. F. Fun, "Application of advanced signal processing techniques to angle-of-arrival estimation in ATC navigation and surveillance system," M.I.T. Lincoln Lab., Lexington, MA, Rep. 582, 1982.
[3] T. J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 4, Aug. 1985.
[4] R. T. Williams, S. Prasad, A. K. Mahalanabis, and L. H. Sibul, "An improved spatial smoothing technique for bearing estimation in a multipath environment," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 4, Apr. 1988.
[5] J. A. Cadzow, "A high resolution direction-of-arrival algorithm for narrow-band coherent and incoherent sources," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, July 1988.
[6] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," in *Proc. RADC Spectral Est. Workshop* (Griffin AFBS, NY), 1979.
[7] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 2, Apr. 1985.
[8] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 10, Oct. 1988.
[9] W. S. Ligget, "Passive sonar: Fitting models to multiple time series," in *NATO ASI on Signal Processing*, J. W. R. Griffiths *et al.*, Eds. New York: Academic, 1973, pp. 327–345.
[10] F. C. Schweppe, "Sensor array data processing for multiple signal sources," *IEEE Trans. Inform. Theory*, vol. IT, no. 14, pp. 294–305, 1968.

A Note on "Efficient Numerically Stabilized Rank-One Eigenstructure Updating"

Marc Moonen, Paul Van Dooren, and Joos Vandewalle

Abstract—**In a recent paper, DeGroat and Roberts developed a reorthogonalization scheme for stabilizing eigenstructure updating algorithms. In this correspondence, we show that only part of this scheme, namely the renormalization, is essential for the stability, so that a cheaper scheme, with roughly half as many computations, can perform equally well.**

INTRODUCTION

Many present-day signal processing algorithms involve a rank-one updating of an eigenvalue decomposition (EVD). In most cases, an estimated data covariance matrix R is updated after each sampling

$$R_k = \alpha \cdot R_{k-1} + (1-\alpha) \cdot x_k \cdot x_k^t$$

(where α is a weighting factor and x_k is a data vector at time step k) and the EVD of R is searched for at each time step

$$R_k = U_k \cdot \Lambda_k \cdot U_k^t.$$

Here U_k is orthogonal and Λ_k is a diagonal matrix. Previously published updating algorithms, however, suffer from a linear buildup of roundoff error, which makes them impractical when large numbers of recursive updates are needed. The core problem appears to be that U_k gradually looses its orthogonality, when it is iteratively being updated according to

$$\begin{aligned} U_k &= U_{k-1} \cdot Q_k \\ &= Q_1 \cdot Q_2 \cdots Q_k. \end{aligned}$$

Here Q_k is an orthogonal matrix (to working accuracy) which is derived one way or another, depending on the particular updating scheme. In any case, it has been demonstrated [1] that the error propagation for such a product of orthogonal matrices is bounded as

$$\|U_k U_k^t - I\|_F \leq (k+1)n^{1.5}\epsilon$$

(where $\|\cdot\|_F$ denotes the Frobenius norm). Clearly, if k is large, the error can be significant.

In their paper,[1] DeGroat and Roberts suggest a reorthogonalization scheme where after each update the column vectors of U_k are pairwise reorthogonalized and renormalized as follows (pairwise Gram–Schmidt, PGS):

for $i = 2, \cdots, n$

$$u_i \Leftarrow u_i - (u_i^t \cdot u_{i-1}) \cdot u_{i-1}$$

$$u_i \Leftarrow \frac{u_i}{\|u_i\|_F}$$

end

Manuscript received July 24, 1990. This work was supported in part by the European Commission.

M. Moonen and J. Vandewalle are with ESAT Katholieke Universiteit Leuven, 3001 Heverlee, Belgium.
P. Van Dooren is with Philips Research Laboratory, 1348 Louvain-la-Neuve, Belgium.

IEEE Log Number 9100781.

[1] R. D. DeGroat and R. A. Roberts, *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 2, pp. 301-316, Feb. 1990.

1053-587X/91/0800-1911$01.00

where u_i is the ith column of U_k. In matrix form, this corresponds to an upper bidiagonal transformation

$$U_k \Leftarrow U_k \cdot B_k.$$

At the kth update, the eigenvectors are composed of a series of rotations followed by partial orthogonalizations

$$U_k = Q_1 \cdot B_1 \cdot Q_2 \cdot B_2 \cdots Q_k \cdot B_k.$$

Simulations seem to confirm that the scheme is long-term stable.

Still, it should clearly be stated that the PGS procedure as such is not able to orthogonalize any random matrix. Suppose we had a nonorthogonal Q_1. Applying the PGS procedure without rotations then gives

$$U_k = Q_1 \cdot B_1 \cdot B_2 \cdots B_k.$$

The matrix U_k is not necessarily closer to orthogonality. As an example, the procedure stagnates for

$$Q_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where then we will have

$$U_k = Q_1.$$

The reason for this is of course that u_1 and u_3 are never combined. A safe procedure would then consist in performing PGS orthogonalizations for all combinations (u_i, u_j) in a cyclic fashion

$$\begin{aligned} (i, j) = (1, 2) \quad & (1, 3) \cdots (1, n-1) \quad (1, n) \\ & (2, 3) \cdots (2, n-1) \quad (2, n) \\ & \qquad\qquad\qquad\qquad \vdots \\ & \qquad\qquad\qquad\qquad (n-1, n). \end{aligned}$$

Such a complete GS orthogonalization can be spread out over several updates, in order not to significantly increase the computational complexity.

On the other hand, the PGS procedure of DeGroat and Roberts clearly performs well in practice. The reason for this is that the rotations Q_k tend to permute or combine the columns of U_k, such that the PGS procedure is not likely to stagnate. The PGS procedure thus heavily relies on the intermediate rotations. In the next section we will show how this can be pushed to extremes, by leaving out the reorthogonalizations, and performing only renormalizations of the columns of U_k.

Renormalizations

When we leave out the orthogonalizations, the correction scheme reduces to the following:

for $i = 1, \cdots, n$

$$u_i \Leftarrow \frac{u_i}{\|u_i\|}$$

end.

In matrix form, this corresponds to a diagonal transformation

$$U_k \Leftarrow U_k \cdot D_k$$

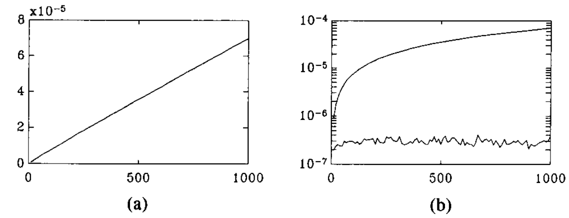

Fig. 1. (a) Orthogonality measure $\|I - U_k^t \cdot U_k\|_2$ (y axis, linear) versus number of updates k (x axis), for an initial 10×10 identity matrix, without intermediate renormalizations. (b) Orthogonality measure $\|I - U_k^t \cdot U_k\|_2$ (y axis, logarithmic) versus number of dates k (x axis), for an initial 10×10 identity matrix, with and without intermediate renormalizations (lower, respectively, upper curve).

so that

$$U_k = Q_1 \cdot D_1 \cdot Q_2 \cdot D_2 \cdots Q_k \cdot D_k.$$

The following small experiments illustrate the effectiveness of these normalizations.

Experiment 1: As an example, successive random rotations Q_k (to the right) were performed on an initial 10×10 identity matrix. The relative machine precision ϵ was set equal to 10^{-8}. When no intermediate normalizations are performed, orthogonality is gradually lost, as can be seen from Fig. 1(a) or (b), upper curve. When, additionally, the columns in U_k are normalized after each rotation, the accumulation is apparently stable (Fig. 1(b), lower curve).

Experiment 2: In order to clearly demonstrate the effect of the normalizations, one can also start from a completely random Q_1-matrix. Successive rotations/normalizations remarkably enough transform the random matrix into an orthogonal matrix. Figs. 2(a) and 2(b) give a linear and a semilog plot. The value for $\|I - U_k^t \cdot U_k\|_2$ settles at the same level as in experiment 1.

Let us now try to explain the above results. A crucial observation is the following: Each time two *normalized* vectors u_i and u_j (column vectors in $U_k \cdot D_k$ in our case) are combined through a 2×2 rotation

$$u_i^* = u_i \cdot \cos\theta - u_j \cdot \sin\theta$$

$$u_j^* = u_i \cdot \sin\theta + u_j \cdot \cos\theta$$

the angle between these vectors gets closer to 90°

$$|\cos \alpha_{i,j}^*| \leq |\cos \alpha_{ij}|$$

where $\alpha_{i,j}(\alpha_{i,j}^*)$ is the angle between u_i and u_j (u_i^* and u_j^*). In other words, each rotation partially orthogonalizes two normalized vectors. The proof of this is straightforward.

Proof: If we let

$$\begin{bmatrix} u_i^t \\ u_j^t \end{bmatrix} \cdot [u_i \quad u_j] = \begin{bmatrix} 1 & \epsilon_{i,j} \\ \epsilon_{i,j} & 1 \end{bmatrix}$$

($\epsilon_{i,j} = \cos \alpha_{i,j}$) then

$$\begin{bmatrix} u_i^{*t} \\ u_j^{*t} \end{bmatrix} \cdot [u_i^* \quad u_j^*] = \begin{bmatrix} 1 - \epsilon_{i,j} \sin 2\phi & \epsilon_{i,j} \cos 2\phi \\ \epsilon_{i,j} \cos 2\phi & 1 + \epsilon_{i,j} \sin 2\phi \end{bmatrix}.$$

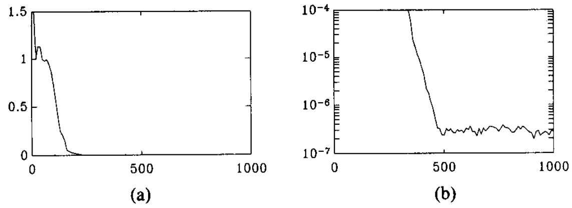

Fig. 2. (a) Orthogonality measure $\|I - U_k^t \cdot U_k\|_2$ (y axis, linear) versus number of updates k (x axis), for an initial random 10×10 matrix, with intermediate renormalizations. (b) Orthogonality measure $\|I - U_k^t \cdot U_k\|_2$ (y axis, logarithmic) versus number of updates k (x axis), for an initial random 10×10 matrix, with intermediate renormalizations.

The new angle can be computed after a normalization as follows:

$$\begin{bmatrix} \dfrac{u_i^{*'}}{\|u_i^*\|} \\ \dfrac{u_j^{*'}}{\|u_j^*\|} \end{bmatrix} \cdot \begin{bmatrix} \dfrac{u_i^*}{\|u_i^*\|} & \dfrac{u_j^*}{\|u_j^*\|} \end{bmatrix} = \begin{bmatrix} 1 & \epsilon_{i,j} \dfrac{\cos 2\phi}{\sqrt{1 - \epsilon_{i,j}^2 \sin^2 2\phi}} \\ \epsilon_{i,j} \dfrac{\cos 2\phi}{\sqrt{1 - \epsilon_{i,j}^2 \sin^2 2\phi}} & 1 \end{bmatrix}.$$

From

$$\cos^2 2\phi \leq 1 - \epsilon_{i,j}^2 \sin^2 2\phi$$

it then follows that

$$\underbrace{\epsilon_{i,j} \frac{\cos 2\phi}{\sqrt{1 - \epsilon_{i,j}^2 \sin^2 2\phi}}}_{\cos \alpha_{i,j}^*} \leq \underbrace{\epsilon_{i,j}}_{\cos \alpha_{i,j}}$$

for every value of ϕ and $|\epsilon_{i,j}| = |\cos \alpha_{i,j}|$. (Note that the equality only holds for either $|\epsilon_{i,j}| = 1$ or $\phi = 0$.)

In the EVD updating scheme, at a certain time step we will have a U_k matrix with normalized columns. The subsequent rotation Q_{k+1} can implicitly be split up into a sequence of 2×2 rotations, each one of which will bring two vectors closer to orthogonality.[2] The combined transformation $D_k \cdot Q_{k+1}$ is thus seen to bring U_{k+1} closer to orthogonality.

Conclusion

We have shown how intermediate renormalizations can stabilize a recursive accumulation of orthogonal transformations. Such accumulations appear in EVD-based adaptive signal processing algorithms. This procedure is a cheap alternative to the partial reorthogonalization scheme of DeGroat and Roberts.[1]

[2]Note that we do not have intermediate normalizations within this sequence of 2×2 transformations. The corresponding terms $1/\sqrt{1 - \epsilon_{i,j}^2 \sin^2 2\phi}$, however, represent second-order effects.

Acknowledgment

This work was carried out at the ESAT Laboratorium of the Katholieke Universiteit Leuven, in the framework of the community research program ESPRIT Basic Research Action 3280.

References

[1] B. N. Parlett, *The Symmetric Eigenvalue Problem.* Englewood Cliffs, NJ: Prentice-Hall, 1980.

Author's Reply[3]

R. D. DeGroat

The computational simplification presented in "A Note on 'Efficient, Numerically Stabilized Rank-One Eigenstructure Updating'" offers an interesting tradeoff between computation and roundoff error performance. The experiments presented are also very intriguing. However, two points need to be addressed concerning the recursive rank-one eigenstructure update: 1) It is shown in the paper by DeGroat and Roberts[1] that pairwise Gram–Schmidt (PGS) followed by normalization virtually guarantees working precision orthogonality for the rank-one updated eigenvectors. However, Fig. 1 demonstrates that normalization by itself *does not* yield working precision orthogonality when applied to rank-one updating. On the other hand, normalization does an amazingly good job at keeping the accumulated orthogonality error small and appar-

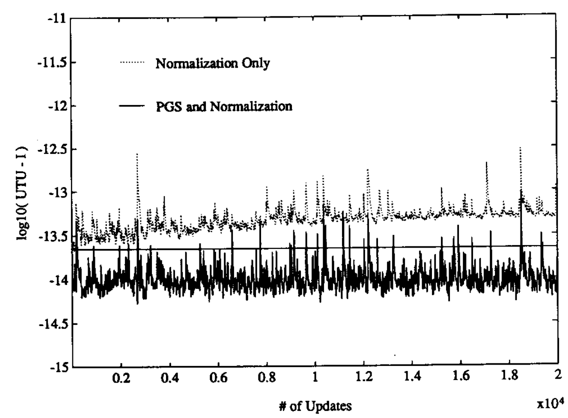

Fig. 1. A comparison of the orthogonality error produced by pairwise Gram-Schmidt with normalization, and normalization only. Both methods are applied to recursive rank-one eigenstructure updating of a 10×10 correlation matrix. The same data (which consists of a 10-dB real sinusoid in white Gaussian noise) is used in both simulations. The solid horizontal line represents the working precision orthogonality error, $n^2\epsilon$, where $n = 10$ is the dimension of the matrices and $\epsilon = 2.22 \times 10^{-16}$ is the machine precision.

[3]Manuscript received February 20, 1991.
The author is with the Erik Jonsson School of Engineering and Computer Science, Richardson, TX 75083-0688.
IEEE Log Number 9100780.

1053-587X/91/0800-1913$01.00

ently bounded. For orthogonal matrix products that are slightly more stable than the rank-one sequence, normalization alone may be sufficient to maintain working precision orthogonality. 2) It should be emphasized that the PGS-normalization method in DeGroat and Roberts' work[1] does not add significant computational cost to the rank-one update. Nevertheless, normalization (without PGS) offers a possible tradeoff for the rank-one update: reduced computation for orthogonality error.

A VLSI System for Real-Time Linear Operations and Transforms

Matthew Newell and John Rasure

***Abstract*—A VLSI system utilizing sixteen systolic array multipliers and designed to compute vector–matrix products at a rate of 640 × 10^6 MAC's is presented. The 448 000-transistor, 1.6-μm CMOS device incorporates a dual timing scheme which allows multiplexing of hardware units over identical operations. This hardware sharing balances maximum internal operating frequency with external data bandwidth and results in an improved ratio of the signal throughput to silicon area. This system has wide application because of its ability to compute correlation, convolution, linear transforms, and connections in multilayer perceptrons.**

I. Introduction

The continued improvement in VLSI systems matched with new developments in digital signal processing has resulted in many applications for general purpose digital signal processors. In fact, over the last few years, general purpose chips such as Texas Instruments' TMS320C30 have offered significant help in overcoming the major barriers to rapid design and cost-effective implementation of complex DSP systems [13]. Also, the now common short lead times for ASIC development have led to the quick implementation of digital signal processing algorithms for the mass market or for critical real time applications [1]–[3], [7], [8]. Despite these successes, there is a considerable gap between low cost, readily available, programmable digital signal processors and custom designed, single application VLSI systems. To fill this gap, an architecture and implementation that shares the beneficial attributes of a general purpose processor and a very high speed application specific processor is presented.

The VLSI system described in this correspondence has a computational rate that compares well with other real-time ASIC's [4], [7]–[9], but it also has the functionality to perform convolution, correlation, and linear transformations. This attribute is obtained by designing a system that does not rely upon the idiosyncrasies of a specific algorithm to obtain its high computational rate [1], [2], but obtains its performance by using a fundamental signal processing algorithm that maps into an architecture that can be highly parallelized. The linear operation implemented is the general vector–matrix multiplication with the matrix coefficients stored on chip. Reorganization of the matrix coefficient memory data allows the vector–matrix multiplier to perform a variety of signal processing algorithms. The vector–matrix multiplier chip can be referred to as an ASIC (because it is designed for a specific algorithm) with general purpose application (because of the multiple functions that this algorithm can implement).

ASIC's are typically designed to have an internal operating frequency that matches the input/output data requirements [9], therefore, the data rate determines the internal operating frequency. The limiting data rate for a general purpose DSP chip is the rate at which a single multiply accumulate can be performed [10]. Again there is a gap between ASIC's and general purpose DSP's. A VLSI system should be designed such that the internal operating frequency is only limited by the technology and if the particular application allows for a lower external data rate, internal components can be reused [5]. In the system described in this correspondence, an internal operating frequency of four times the external data rate allows time multiplexing of the internal components. This dual frequency operating scheme improves the data processing rate to silicon area ratio [6].

The VLSI system presented below can be characterized as follows: highly parallel (16 parallel, systolic array multipliers), reusable components (different internal/external data rates allow hardware reuse), real-time computation for audio and image processing applications, and a general purpose signal processor. The following section describes vector–matrix multiplication and describes the resulting architecture and its operation. Section III summarizes how the system can be applied to signal processing algorithms.

II. Vector–Matrix Multiplier Design

A. Linear Operation Algorithm

Vector–matrix multiplication can be described as $y = xA$, where x is an m element real input vector, A is an $m \times n$ real data matrix and y is an n element real output vector. The architecture described in this correspondence can compute this multiplication along with the adding of an offset vector to the solution. Therefore, the function becomes $y = xA + b$, where b is an n element offset vector. Each element of the output vector can be described by the following multiplication and accumulation:

$$y_i = \sum_{j=0}^{m-1} [x_j * A_{ji}] + b_i; \quad 0 < i \le n - 1. \tag{2.1}$$

Using a single chip this architecture can accomplish a vector–matrix multiplication with an input vector x that can vary in length m from 4 elements to 64 elements. The number of columns in the data matrix n can vary from 4 coefficients to 64 coefficients. The chip is designed to be arrayed in two dimensions and therefore the maximum size of both the input vector and the matrix can be extended. All input data are 8-b, two's complement numbers.

It is important to note there are no assumptions made of the matrix contents, such as its symmetry, that often lead to a reduced or more efficient implementation [6]. The 64 × 64 matrix size is dictated by die size limitations given currently available process technologies (see Section II-D).

B. System Organization and Block Diagram

The system is designed for an external input and output signal processing rate of 10 MHz. The 1.6-μm CMOS technology that

Manuscript received June 17, 1990; revised November 5, 1990. This work was supported by Sandia National Laboratories, Albuquerque, NM, under Contract 55-5633.

The authors are with the Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131.

IEEE Log Number 9100779.