[9] Wang, L., Xiang, N., Wang, S., and Huang, M. (1989)
Parallel reduced gradient optimal power flow solution.
Electric Power Systems Research 17 (1989), 229–237.

[10] Taoka, H., Iyoda, I., Noguchi, H., Sato, N., and Nakazawa, T. (1992)
Real-time digital simulation for power system analysis on a hypercube computer.
IEEE Transactions On Power Systems, 7, 1 (Feb. 1992), 1–7.

[11] Chen, H. M., and Berry, F. C.
Parallel load-flow algorithm using a decomposition method for space based power systems.
IEEE Transactions on Aerospace and Electronic Systems, To be published.

Inversion of all Principal Submatrices of a Matrix

Let $\mathbf{A}_m$ be an $m \times m$ principal submatrix of an infinite-dimensional matrix A. We give a simple formula which expresses $\mathbf{A}_{m+1}^{-1}$ in terms of $\mathbf{A}_m^{-1}$, and based on this formula, an algorithm which computes the inverses of $\mathbf{A}_m$ for $m = 1,2,3,\ldots,n$ using only $2n^3 - 2n^2 + n$ arithmetic operations. This is an improvement over the naive method of computing the inverses separately which would require $\sum_{m=1}^{n} m^3 = O(n^4)$ arithmetic operations.

I. THE MOTIVATION

The following problem is frequently encountered in many aerospace engineering calculations, such as the Kalman filtering algorithm, as well as in signal processing and systems control theory [2, 3, 1, 4]. Given a tolerance $\epsilon > 0$ and an infinite-dimensional linear system of equations of the form

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \end{bmatrix}$$

find a finite-dimensional solution vector $\mathbf{x}_n = [x_1, x_2, \ldots, x_n]^T$ such that

$$f(\mathbf{x}_n) \le \epsilon$$

for some nonnegative objective functional f. This can be achieved by solving the following sequence of (increasing) finite-dimensional problems:

$$[a_{11}][x_1] = [b_1]$$

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\vdots$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

and checking if the above tolerance criterion is satisfied. Solving this sequence of finite-dimensional linear systems is equivalent to computing the following sequence of matrix inversions:

$$\mathbf{A}_1^{-1}, \mathbf{A}_2^{-1}, \mathbf{A}_3^{-1}, \ldots, \mathbf{A}_n^{-1}$$

where $\mathbf{A}_m$ for $1 \le m \le n$ is an $m \times m$ principal submatrix of the infinite-dimensional system matrix $\mathbf{A} = [a_{ij}]_{i,j=1}^{\infty}$.

II. THE FORMULA

We partition the $(m+1) \times (m+1)$ matrix $\mathbf{A}_{m+1}$ as follows

$$\mathbf{A}_{m+1} = \left[\begin{array}{ccc|c} a_{11} & \cdots & a_{1m} & a_{1,m+1} \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mm} & a_{m,m+1} \\ \hline a_{m+1,1} & \cdots & a_{m+1,m} & a_{m+1,m+1} \end{array}\right] : = \begin{bmatrix} \mathbf{A}_m & \mathbf{b} \\ \mathbf{c}^T & a \end{bmatrix}.$$

We assume that $\mathbf{A}_m^{-1}$ exists and

$$d := a - \mathbf{c}^T \mathbf{A}_m^{-1} \mathbf{b} \neq 0.$$

Let $\alpha := 1/d$. We claim that the inverse of $\mathbf{A}_{m+1}$ is given by

$$\mathbf{A}_{m+1}^{-1} = \begin{bmatrix} \mathbf{A}_m^{-1} + \alpha \mathbf{A}_m^{-1} \mathbf{b} \mathbf{c}^T \mathbf{A}_m^{-1} & -\alpha \mathbf{A}_m^{-1} \mathbf{b} \\ -\alpha \mathbf{c}^T \mathbf{A}_m^{-1} & \alpha \end{bmatrix}.$$

This claim is easily verified as follows:

$$\begin{aligned} \mathbf{A}_{m+1}\mathbf{A}_{m+1}^{-1} &= \begin{bmatrix} \mathbf{A}_m & \mathbf{b} \\ \mathbf{c}^T & a \end{bmatrix} \begin{bmatrix} \mathbf{A}_m^{-1} + \alpha \mathbf{A}_m^{-1} \mathbf{b} \mathbf{c}^T \mathbf{A}_m^{-1} & -\alpha \mathbf{A}_m^{-1} \mathbf{b} \\ -\alpha \mathbf{c}^T \mathbf{A}_m^{-1} & \alpha \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_m + \alpha \mathbf{b} \mathbf{c}^T \mathbf{A}_m^{-1} - \alpha \mathbf{b} \mathbf{c}^T \mathbf{A}_m^{-1} & -\alpha \mathbf{b} + \alpha \mathbf{b} \\ \mathbf{c}^T \mathbf{A}_m^{-1} + \alpha \mathbf{c}^T \mathbf{A}_m^{-1} \mathbf{b} \mathbf{c}^T \mathbf{A}_m^{-1} - \alpha a \mathbf{c}^T \mathbf{A}_m^{-1} & -\alpha \mathbf{c}^T \mathbf{A}_m^{-1} \mathbf{b} + \alpha a \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} = \mathbf{I}_{m+1}. \end{aligned}$$

Manuscript received March 11, 1993.

IEEE Log No. T-AES/30/1/13055.

This work was supported in part by the US Army Research Office under Grant DAAL03-91-G-0106 and by the Institute of Space Systems Operations, University of Houston.

0018-9251/94/$4.00

TABLE I
Computation of A_{m+1}^{-1} Given A_m^{-1}

Step	Operation	Complexity
1.	$(A_m^{-1})(b)$	$m(2m-1)$
2.	$(c^T)(A_m^{-1})$	$m(2m-1)$
3.	$(c^T A_m^{-1})(b)$	$2m-1$
4.	$(a)-(c^T A_m^{-1} b)$	1
5.	$\alpha := 1/(a - c^T A_m^{-1} b)$	1
6.	$(-\alpha)(A_m^{-1} b)$	m
7.	$(-\alpha)(c^T A_m^{-1})$	m
8.	$(\alpha A_m^{-1} b)(c^T A_m^{-1})$	m^2
9.	$(A_m^{-1}) + (\alpha A_m^{-1} b c^T A_m^{-1})$	m^2

It is also straightforward to show that $A_{m+1}^{-1} A_{m+1} = I_{m+1}$.

III. THE ALGORITHM

The algorithm computes the inverses in the following order: $A_1^{-1}, A_2^{-1}, A_3^{-1}, \ldots, A_n^{-1}$. Once A_m^{-1} is obtained, we compute A_{m+1}^{-1} using some additional vector-vector, vector-matrix, and matrix-matrix operations. We give the steps of the algorithm in Table I, where the number of arithmetic operations required at each step is also indicated.

Let $T(m)$ be the number of arithmetic operations required to compute the inverse of A_m. Assuming that A_m^{-1} is already computed using $T(m)$ arithmetic operations, we proceed to compute A_{m+1}^{-1} using additional $6m^2 + 2m + 1$ arithmetic operations, as can be seen from Table I. Note that A_1 is only a scalar, thus $T(1) = 1$. As m runs from 2 to n, we obtain the inverses of all A_m for $2 \le m \le n$ using

$$T(n) = T(n-1) + 6(n-1)^2 + 2(n-1) + 1$$

arithmetic operations. Thus, the number of arithmetic operations required for computing A_m^{-1} for $1 \le m \le n$ is found as

$$\begin{aligned} T(n) &= T(n-1) + 6(n-1)^2 + 2(n-1) + 1 \\ &= T(n-2) + 6(n-2)^2 + 6(n-1)^2 \\ &\quad + 2(n-2) + 2(n-1) + 1 + 1 \\ &\vdots \\ &= T(1) + 6\sum_{m=1}^{n-1} m^2 + 2\sum_{m=1}^{n-1} m + \sum_{m=1}^{n-1} 1 \\ &= 1 + (n-1)n(2n-1) + (n-1)n + (n-1) \\ &= 2n^3 - 2n^2 + n. \end{aligned}$$

ÇETIN KAYA KOÇ
Department of Electrical & Computer Engineering
Oregon State University
Corvallis, OR 97331

GUANRONG CHEN
Department of Electrical Engineering
University of Houston
Houston, TX 77204

REFERENCES

[1] Chui, C. K., and Chen, G. (1992)
Signal Processing and System Theory: Selected Topics.
New York: Springer-Verlag, 1992.

[2] Gohberg, I. (1986)
I. Schur Methods in Operator Theory and Signal Processing.
Boston, MA: Birkhäuser, 1986.

[3] Gohberg, I. (1988)
Topics in Interpolation Theory of Rational Matrix-Valued Functions.
Boston, MA: Birkhäuser, 1988.

[4] Lu, M., Qiao, X. Z., and Chen, G. (1992)
A parallel square-root algorithm for modified Kalman filter.
IEEE Transactions on Aerospace and Electronic Systems, **28**, 1 (1992), 153–163.

Application of Neural Networks in Target Tracking Data Fusion

Kalman filtering is a fundamental building block of most multiple-target tracking (MTT) algorithms. The other building block usually involves some type of data association schemes. Here it is proposed to incorporate a neural network into the normal Kalman filter configuration such that the neural network provides the adaptive capability the filter needs. As such, the estimation error of the Kalman filter would be reduced, hence improving the MTT solution. Simulation results have shown that this claim is valid.

I. INTRODUCTION

Target tracking is an important issue in military surveillance systems, especially when such systems employ multiple sensors to interpret the environment. Typical sensors such as radar, infrared, electronic support measure, sonar, etc. report measurements from various sources including target kinematics, attributes, clutter and background noise. The objective of target tracking, or in general, multiple-target tracking (MTT) is to partition sensor data into sets of observations, or tracks produced by the same source. Once tracks are formed and confirmed, the number of targets can be estimated and parameters such as position and velocity can be obtained from each track. When multiple sensors are used, the

Manuscript received March 1, 1993; revised May 6, 1993.

IEEE Log No. T-AES/30/1/13056.

0018-9251/94/$4.00 © 1994 IEEE