Solving the Algebraic Riccati Equation with the Matrix Sign Function

Ralph Byers
Department of Mathematics
North Carolina State University at Raleigh
Raleigh, North Carolina 27695-8205

Submitted by Hans Schneider

ABSTRACT

This paper presents some improvements to the matrix-sign-function algorithm for the algebraic Riccati equation. A simple reorganization changes nonsymmetric matrix inversions into symmetric matrix inversions. Scaling accelerates convergence of the basic iteration and yields a new quadratic formula for certain 2-by-2 algebraic Riccati equations. Numerical experience suggests the algorithm be supplemented with a refinement strategy similar to iterative refinement for systems of linear equations. Refinement also produces an error estimate. The resulting procedure is numerically stable. It compares favorably with current Schur vector-based algorithms.

1. INTRODUCTION

This paper presents a modification of matrix sign-function based algorithms for solving algebraic Riccati equations [1, 5, 7, 25]. Our algorithm exploits Hamiltonian structure to change full matrix inversions into symmetric matrix inversions. Symmetric matrix inversions require half the work and storage of full matrix inversions. Scaling by the geometric mean of the eigenvalues accelerates convergence of the sign iteration. The new algorithm reduces to the (scalar) quadratic formula for 1-by-1 Riccati equations. It gives a new quadratic formula for certain 2-by-2 Riccati equations. We also propose that iterative refinement be used to improve numerical stability. Refinement also produces an error estimate. With respect to work, storage, and accuracy, the algorithm compares favorably with the Schur vector-based algorithm of [17].

52 Vanderbilt Ave., New York, NY 10017 0024-3795/87/$3.50

Roberts [24, 25] and Beavers and Denman [7] introduced the matrix sign function as a means of solving algebraic Riccati equations and Lyapunov equations. The matrix sign function has since attracted the attention of control engineers and some applied mathematicians. Yoo and Denman [30] and Lupas and Popeea [21] extended it to differential Riccati equations. Barraud [6] and Bierman [8] designed algorithms to solve discrete Riccati equations. Balzer [4] and Barraud [5] have described strategies for accelerating convergence. Denman and Beavers [10] and Denman and Layva-Ramos [11] extended matrix-sign-function algorithms to a list of several invariant-subspace-related calculations. Matheys [22] used it for stability analysis. Howland [14] used the matrix sign function to count eigenvalues in boxes in the complex plane. Some of the algorithms have been refined and extended by Attarzadeh [3], Bierman [8], and Byers [9]. Recently, Higham [12] has used matrix-sign-function techniques to calculate polar decompositions.

2 THE MATRIX SIGN FUNCTION AND ACCELERATION BY SCALING

Let the n-by-n matrix K have Jordan canonical form

$$K = MJM^{-1} = M(D + N)M^{-1},$$

where M is a matrix of eigenvectors and principal vectors, J is a matrix of Jordan blocks, and $D = \operatorname{diag}(d_1, d_2, d_3, \ldots, d_n)$ is diagonal. N is nilpotent and commutes with D. Let $\operatorname{Re}(z)$ denote the real part of the complex number z. The matrix sign function of K is defined by

$$\operatorname{Sign}(K) = MSM^{-1}$$

where $S = \operatorname{diag}(s_1, s_2, s_3, \ldots, s_n)$ is a diagonal matrix whose diagonal entries are given by

$$s_i = \begin{cases} 1 & \text{if } \operatorname{Re}(d_i) > 0, \\ -1 & \text{if } \operatorname{Re}(d_i) < 0. \end{cases}$$

If one of the eigenvalues of K lies on the imaginary axis, then $\operatorname{Sign}(K)$ is undefined.

If Sign(K) is defined, then the Newton iteration for the square root of the identity matrix,

$$\begin{aligned} W^{(0)} &:= K, \\ W^{(k+1)} &:= W^{(k)} - \frac{W^{(k)} - W^{(k)^{-1}}}{2}, \end{aligned} \tag{1}$$

converges to Sign(K) [12, 25]. We prefer the above formulation to $W^{(k+1)} = (W^{(k)} + W^{(k)^{-1}})/2$, because it confines rounding errors to the eventually small correction $(W^{(k)} - W^{(k)^{-1}})/2$. If Sign($K$) is undefined, then either the iteration does not converge or one of the $W^{(k)}$'s is singular. The iteration (1) is a practical procedure for calculating Sign(K).

Although (1) ultimately converges quadratically, initially progress may be slow [9, 12]. For example, let I_n be the n-by-n identity matrix, and suppose $W^{(0)} := K = 5000I_n$. The first twelve iterations of (1) differ insignificantly from $W^{(k+1)} := W^{(k)}/2$. At $O(n^3)$ floating-point operations per iteration, this is an expensive way to divide by 2.

If $c \in R$ is positive, then Sign(K) = Sign(cK). Convergence can be accelerated by scaling $W^{(k)}$ at each iteration to have eigenvalues as close to ± 1 as possible. Let $\lambda(W)$ denote the eigenvalues of W. Define the distance function $D(W)$ by

$$D(W) = \sum_{\lambda \in \lambda(W)} (\ln|\lambda|)^2.$$

This is a measure of the distance of $\lambda(W)$ from the unit circle, so it is a partial measure of how much $\lambda(W)$ differs from $\{\pm 1\}$. $D(W) = 0$ iff $\lambda(W)$ lies on the unit circle in the complex plane. It is an increasing function of $|1 - |\lambda||$ for each $\lambda \in \lambda(W)$. As the iteration (1) converges, $D(W^{(k)})$ goes to zero.

Some elementary calculus shows $D(cW)$ is minimized by $c = |\det(W)|^{-1/n}$, i.e., the reciprocal of the geometric mean of the eigenvalues of W. Incorporating this scaling into (1) gives

$$\begin{aligned} W^{(0)} &:= K, \\ Z^{(k)} &:= W^{(k)}|\det(W^{(k)})|^{1/n}, \\ W^{(k+1)} &:= Z^{(k)} - \frac{Z^{(k)} - Z^{(k)^{-1}}}{2} \end{aligned} \tag{2}$$

It is convenient to calculate the determinant from the same triangular factors

of W that are used to calculate $W^{(k)^{-1}}$. See the LINPACK matrix inversion subroutines DGEDI and DSIDI [20] for details.

Trivial cases are handled trivially by (2). If K is a real 1-by-1 matrix, then $W^{(0)}/|\det(W)| = W^{(1)} = \text{Sign}(K)$. So (2) produces $\text{Sign}(W)$ in one iteration. If $W^{(0)}$ is 2-by-2 with real eigenvalues, then $W^{(1)}$ has two eigenvalues of equal magnitude, and $W^{(2)} = \text{Sign}(K)$. If K is 3-by-3 or 4-by-4 with real eigenvalues, it appears that $\text{Sign}(K)$ is produced in a finite number of iterations, but we have not been able to prove anything.

The above iteration is scale invariant in the sense that multiplying $W^{(0)}$ by a positive constant does not change the subsequent $W^{(k)}$'s. For the example $W^{(0)} := K = 5000I_n$, $W^{(1)} = \text{Sign}(K)$.

Balzer [4] gives other acceleration strategies that are not scale invariant. Barraud [5] suggests scaling by the geometric mean of the largest and smallest eigenvalue, but usually these eigenvalues are not available.

Recently, in independent work, Higham [12] has suggested using iteration (2) scaling by $[\|W^{(k)^{-1}}\|_1 \|W^{(k)^{-1}}\|_\infty / (\|W^{(k)}\|_1 \|W^{(k)}\|_\infty)]^{1/4}$ instead of $|\det(W^{(k)})|^{-1/n}$. Here $\|\cdot\|_1$ and $\|\cdot\|_\infty$ represent the and l_1 and l_∞ operator norms, respectively. Higham's iteration is also scale invariant, and initially it also accelerates convergence. However, it destroys the ultimate quadratic convergence. Higham wisely suggests switching to the unscaled iteration (1) once $W^{(k)}$ gets "close" to convergence. The iteration (2) does not slow quadratic convergence. Also, Higham's iteration does not give two-step convergence for 2-by-2 matrixes with real eigenvalues as (2) does.

3. SOLVING THE ALGEBRAIC RICCATI EQUATION WITH THE MATRIX SIGN FUNCTION

The algebraic Riccati equation

$$G + A^T X + XA - XFX = 0 \tag{3}$$

arises in stochastic and optimal control. A, G, and F are known n-by-n matrices. G and F are symmetric and positive semidefinite. The desired solution, X, is symmetric positive semidefinite and stabilizing in the sense that all eigenvalues of $A - FX$ have negative real part. Under mild conditions, such a solution exists and is unique. We will assume the desired solution exists and is unique. A discussion of the algebraic Riccati equation and its role in control theory can be found in many textbooks. See for example [16] or [29].

The algebraic Riccati equation (3) is equivalent to the $2n$-dimensional matrix equation [23]

$$K = \begin{bmatrix} A^T & G \\ F & -A \end{bmatrix}$$

$$= \begin{bmatrix} X & -I_n \\ I_n & 0_n \end{bmatrix} \begin{bmatrix} -(A-FX) & -F \\ 0_n & (A-FX)^T \end{bmatrix} \begin{bmatrix} X & -I_n \\ I_n & 0_n \end{bmatrix}^{-1}. \quad (4)$$

I_n indicates the n-by-n identity matrix, and 0_n indicates an n-by-n matrix of zeros. Since the eigenvalues of $A - FX$ have negative real part, the matrix sign function is defined. Applying the matrix sign function to (4) gives

$$\operatorname{Sign}(K) = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

$$= \begin{bmatrix} X & -I_n \\ I_n & 0_n \end{bmatrix} \begin{bmatrix} I_n & Z \\ 0_n & -I_n \end{bmatrix} \begin{bmatrix} X & -I_n \\ I_n & 0_n \end{bmatrix}^{-1}. \quad (5)$$

Since K and $\operatorname{Sign}(K)$ commute, Z satisfies the Lyapunov equation

$$(A - FX)Z + Z(A - FX)^T = 2F.$$

Let $M \in R^{2n \times n}$ be the first n columns and let $N \in R^{2n \times n}$ be the last n columns of $W - I_{2n}$. Equation (5) determines X by

$$MX = -N. \quad (6)$$

Since

$$\begin{bmatrix} -I_n & X \end{bmatrix} M = \begin{bmatrix} -I_n & X \end{bmatrix} \begin{bmatrix} -2I - XZ \\ -Z \end{bmatrix} = 2I_n,$$

(6) is a full-rank, consistent system of $2n^2$ equations in the n^2 unknown entries of X. Using the normal equations to solve for X squares the condition number of M; it is safer to use a QR factorization. See Chapter 11 of [18].

The scaled sign iteration (2) also handles trivial Riccati equations trivially. For real scalar quadratic equations, K in (5) is a 2-by-2 matrix with real eigenvalues of equal magnitude and opposite sign. So $K/(|\det(K)|^{1/2}) =$

Sign(K). Thus (2) produces Sign(K) in one iteration. Substituting this expression for W in (5) and simplifying turns (6) into the usual scalar quadratic formula. The discriminant of the scalar quadratic equation is $\det(K)$. For 2-by-2 algebraic Riccati equations, K in (5) is a 4-by-4 matrix with two pairs of eigenvalues of equal magnitude and opposite sign. If these are real eigenvalues, then in (2) $W^{(1)}$ has two double eigenvalues of equal magnitude and opposite sign, and $W^{(2)} = \text{Sign}(K)$. So (2), (5), and (6) form a quadratic formula for 2-by-2 Riccati equations with real eigenvalues.

This relationship with the quadratic formula is not shared by the acceleration of [4] nor by the scaling strategy of [12].

Define $J \in R^{2n \times 2n}$ to be

$$J = \begin{bmatrix} 0_n & I_n \\ -I_n & 0_n \end{bmatrix}.$$

A matrix $H \in R^{2n \times 2n}$ is said to be Hamiltonian if JH is symmetric. The matrix K in (5) is Hamiltonian. Matrix inversion, scalar multiplication, and matrix addition preserve Hamiltonian matrices, so throughout the iteration (2), $W^{(k)}$ is Hamiltonian. The full $2n$-by-$2n$ matrix inversions can be changed to symmetric $2n$-by-$2n$ matrix inversions by organizing $Z^{(k)^{-1}}$ as $(JZ^{(k)})^{-1}J$ [8, 9]. Incorporating this change into (2) gives

$$\begin{aligned} W^{(0)} &:= K \\ Z^{(k)} &:= W^{(k)} \left| \det(W^{(k)}) \right|^{-1/(2n)} \\ W^{(k+1)} &:= Z^{(k)} - \frac{Z^{(k)} - (JZ^{(k)})^{-1} J}{2} \end{aligned} \tag{7}$$

Of course it is unnecessary to actually perform matrix multiplication by J. It suffices to rearrange the components of W, changing signs where necessary. Using symmetry cuts work and storage requirements almost in half. See the LINPACK symmetric inversion subroutines DSICO and DSIDI [18].

4. COMPUTATIONAL DETAILS

Ordinarily, rounding errors and stopping (7) after a finite number of iterations corrupt the results. It is difficult to analyze the effect these errors have on the computed solution $\hat{X}$. It is prudent to refine $\hat{X}$ and estimate the error $X - \hat{X}$.

It is known that the desired solution is symmetric. So some small improvement can be made by replacing the calculated solution $\hat{X}$ by $(\hat{X} + \hat{X}^T)/2$, the Frobenius projection of $\hat{X}$ onto the space of symmetric matrices.

Let $R = R(\hat{X}) = G + A^T\hat{X} + \hat{X}A - \hat{X}F\hat{X}$. If $P \in R^{n\times n}$, $P = P^T$, and P satisfies the algebraic Riccati equation [8]

$$R + (A - F\hat{X})^T P + P(A - F\hat{X}) - PFP = 0, \tag{8}$$

then $X = \hat{X} + P$ satisfies the original algebraic Riccati equation (3). So an approximate solution $\hat{X}$ can be refined by solving (8) for P and replacing $\hat{X}$ by $\hat{X} + P$. Since $\hat{X}$ is an approximate solution of the original algebraic Riccati equation, the correction P is small. The refinement step is well suited to Newton's method starting with the initial guess $P = 0$ [13, 15]. The matrix sign function can be regarded simply as a way to obtain a good initial guess for Newton's method. The eigenvalues of $A - FX$ are a by-product of some implementations of Newton's method [13]. Bierman has observed that almost any algebraic Riccati solver can be used to solve (8)—even the matrix sign function itself [8].

In practice, rounding errors corrupt the calculation of P to give a matrix $\hat{P}$. The refinement step may need to be repeated iteratively. If the first few significant binary digits of the entries of $\hat{P}$ are correct, then $\hat{X} + \hat{P}$ is more accurate than $\hat{X}$. Thus, if the underlying algebraic Riccati equation is not too ill conditioned, then the accuracy attainable is limited only by the accuracy of the arithmetic, the condition of the Riccati equation, and the accuracy to which $R = R(\hat{X})$ is calculated [9]. Iterative refinement makes the algorithm numerically stable.

Note that $\hat{P}$ gives the error estimate

$$X - \hat{X} = P \approx \hat{P}. \tag{9}$$

Often limiting accuracy is reached after one or two refinements. Even when there is no improvement, the error estimate (9) is of the correct magnitude.

Some of the benefits of refining with Newton's method have also been observed in [2].

Admittedly some economy of work and storage of the matrix sign function is lost in the refining process. In our numerical experiments using Newton's method the refinements accounted for 25% to 50% of the work.

The iteration (7) needs a stopping criterion. Inverting the $JZ^{(k)}$'s accounts for the most significant rounding errors. Using t-digit base-b arithmetic, the relative error of the calculated inverse tends to be about

$cb^{1-t}[\|JZ^{(k)}\|\,\|(JZ^{(k)})^{-1}\|]$. Here $\|\cdot\|$ may be any reasonably well-balanced matrix norm. The number c is a low-degree polynomial in n that depends on the norm and on the details of the arithmetic [28]. This suggests that the iteration be stopped when

$$\frac{\left\|Z^{(k)}-(JZ^{(k)})^{-1}J\right\|}{\left\|(JZ^{(k)})^{-1}\right\|}\leqslant nb^{1-t}\|JZ^{(k)}\|\left\|(JZ^{(k)})^{-1}\right\|. \tag{10}$$

Another possible stopping criterion is to stop when the iterates $W^{(k)}$ no longer change significantly, i.e. when

$$\left\|Z^{(k)}-(JZ^{(k)})^{-1}J\right\|\leqslant b^{1-t}\|Z^{(k)}\|. \tag{11}$$

In practice the criterion (10) sometimes stops the iteration too early, and the criterion (11) sometimes stops it too late. Rounding errors may prevent (11) from ever being satisfied. We use the observations of [5] to choose a compromise between them.

The following algorithm summarizes the preceding discussion.

ALGORITHM SGNREF

INPUT: $A,G,F\in R^{n\times n}$; $G=G^T$; $F=F^T$

OUTPUT: $X\in R^{n\times n}$ approximately satisfying (3) and such that all eigenvalues of $A-FX$ have negative real part; error estimate $P\in R^{n\times n}$

1. $W:=\begin{bmatrix}A^T & G\\ F & -A\end{bmatrix}$; DONE := FALSE
2. FOR $j=1,2,3,\ldots$ UNTIL (DONE = TRUE)
   2.1. Use LINPACK subroutines DSICO and DSIDI to calculate

$$k:=\|JW\|\left\|(JW)^{-1}\right\|$$

$$d:=|\det(JW)|^{1/2n}$$

$$Y:=d(JW)^{-1}$$

   2.2. $Z=W/d$
   2.3. $s=\|Z-YJ\|$
   2.4. $W:=Z-(Z-YJ)/2$
   2.5. IF($s\leqslant b^{1-t}\|Z\|$) THEN DONE := TRUE
   2.6. IF($8\leqslant j$ AND $s\leqslant nb^{1-t}k\|Y\|$) THEN DONE := TRUE

3. Partition $W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$

4. Use LINPACK routines DQRDC and DQRSL to solve for X in

$$\begin{bmatrix} W_{11} - I_n \\ W_{21} \end{bmatrix} X = \begin{bmatrix} -W_{12} \\ I_n - W_{22} \end{bmatrix}$$

5. $X := (X + X^T)/2$

6. Solve (8) for P by Newton's method [13] with initial guess $P = 0$ or some other means [8].

7. $X := X + P$

The solution can be further refined by iterating steps 6 and 7.

It is unnecessary to perform the matrix multiplications by J in step 2. It suffices to rearrange the components of W and Z, changing signs where necessary. W is conveniently represented as the lower triangle of the symmetric matrix JW. It can be stored in the lower triangle of a $2n$-by-$2n$ array. The upper triangle can be used by DSICO and DSIDI to calculate $\det(JW)$ and $(JW)^{-1}$.

The above algorithm uses approximately $6n^2$ storage locations. The full matrix A requires n^2 locations. The symmetric matrices G and F require $n(n+1)/2$ locations each. A $2n$-by-$2n$ workspace array is used in step 2 to calculate $\mathrm{Sign}(W)$. The same workspace may be used to form X in steps 4 and 5 and to set up and solve (8) for P. Our implementation of Newton's method [9] also requires $6n^2$ storage locations. The Schur vector-based algorithm [17] uses $8n^2$ storage locations.

5. EXAMPLES

Algorithm SGNREF was programmed in FORTRAN with Newton's method for the refining step. The Schur vector-based algorithm of [17], SCHVEC, was also programmed using subroutines modified from EISPACK [26]. The two programs were tested on several algebraic Riccati equations (3). Where the exact solutions were not known, errors were estimated by (8) and (9). All computations were performed on Northern Illinois University's DEC VAX 11/750 with floating-point accelerator UNIX f77 compiler. Timings were done while no one else was using the computer.

EXAMPLE 1. The first example comes from the position and velocity control of a string of high-speed vehicles [19]. A string of k vehicles gives rise to order-$(2k-1)$ coefficient matrices $A=[a_{ij}]$, G, and F of the form

$$G=\operatorname{diag}(0,10,0,10,\ldots,0),$$

$$F=\operatorname{diag}(1,0,1,0,\ldots,1),$$

and

$$a_{ij}=\begin{cases}-1 & \text{if } i=j \text{ and } i \text{ is odd},\\ -1 & \text{if } i=j-1 \text{ and } i \text{ is even},\\ 1 & \text{if } i=j+1 \text{ and } i \text{ is even},\\ 0 & \text{otherwise.}\end{cases}$$

Laub used the cases of five, ten, and twenty vehicles as an example in [17]. We tested the five-vehicle and twenty-vehicle cases. SCHVEC solved the five-vehicle problem in 5.2 seconds and the twenty-vehicle problem in 240 seconds. SGNREF solved the five-vehicle problem in 3.4 seconds and the twenty-vehicle problem in 120 seconds. Only one Newton step was required for the refinement. In both cases both algorithms produced solutions accurate to about fifteen significant decimal digits. Machine epsilon was about 10^{-17}. SGNREF had no difficulty solving this well-conditioned Riccati equation as accurately and somewhat less expensively than SCHVEC.

EXAMPLE 2. This more ill-conditioned algebraic Riccati equation is Example 6 in [17]. Define n-by-n matrices $A=[a_{ij}]$, G, and F by

$$G=\operatorname{diag}(1,0,0,0,\ldots,0),$$

$$F=\operatorname{diag}(0,0,0,0,\ldots,1),$$

and

$$a_{ij}=\begin{cases}1 & \text{if } i=j-1,\\ 0 & \text{otherwise.}\end{cases}$$

Like [17], we used $n=21$. SCHVEC finished this problem in 47 seconds. SGNREF finished it in 28 seconds. Only one Newton step was required by the refinement step. Although the arithmetic was accurate to approximately 17 significant decimal digits, both algorithms produced solutions accurate to

only nine significant decimal digits. The less accurate solutions of this problem are due to the ill-conditioning of the underlying Riccati equation [2, 9, 17]. They are not the fault of the methods themselves. The ill-conditioning did not slow convergence.

EXAMPLE 3. The last example is artificially constructed to demonstrate the improvement in accuracy that can be obtained from the refining step in SGNREF. The coefficient matrices were the 20-by-20 matrices $A = VBV$, $G = VCV$, and $F = VDV$, where V is the symmetric, orthogonal matrix $V = I - (2/n)uu^T$, $u \in R^{20}$ is the vector of 1's, and B, C, and F are 20-by-20 matrices following the pattern of the 3-by-3 example

$$B = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & -3 \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1.01 \times 10^{-7} \end{bmatrix},$$

$$D = \begin{bmatrix} 10^7 & 10^7 & 10^7 \\ 10^7 & 10^7 & 10^7 \\ 10^7 & 10^7 & 10^7 \end{bmatrix}.$$

SCHVEC and SGNREF both used about 28 seconds. SGNREF took longer than usual, because it needed two Newton steps to refine the solution. This time they did not give solutions of the same accuracy. A, G, and F have different magnitudes. Rounding errors in algorithms that work with the Hamiltonian matrix (4) as a whole tend to perturb G by amounts proportional to the magnitude of F times the precision of the arithmetic. This caused SCHVEC to produce a solution correct to about four significant decimal digits. After refining with two steps of Newton's method, SGNREF produced a solution correct to about 16 significant decimal digits.

Of course, solutions produced by SCHVEC can also be refined with a few steps of Newton's method.

Arnold [2] has also observed the advantages of refining solutions with Newton's method.

CONCLUSIONS

The matrix sign function with iterative refinement is an efficient numerically stable method for solving algebraic Riccati equations. Scaling by the inverse of the geometric mean of the eigenvalues accelerates convergence of

the iteration. The resulting algorithm reduces to the scalar quadratic formula in the 1-by-1 case. Taking advantage of Hamiltonian structure by turning unsymmetric matrix inversions into symmetric inversions cuts work and storage requirements in half. Iterative refinement gives numerical stability and an error estimate. With respect to work, storage, and accuracy, the matrix sign function followed by iterative refinement compares favorably with the Schur vector method of [17].

REFERENCES

1 B. D. O. Anderson, Second order convergent algorithms for the steady state Riccati equation, *Internat. J. Control* 28 295–306 (1978).

2 W. F. Arnold, Numerical solution of algebraic matrix Riccati equations, Report NWC TP 6521, Naval Weapons Center, China Lake, Calif. 93555, 1984.

3 F. Attarzadeh, Block decomposition algorithm for time-invariant systems using the generalized matrix sign function, *Internat. J. Systems Sci.* 14:1075–1085 (1983).

4 L. A. Balzer, Accelerated convergence of the matrix sign function, *Internat. J. Control* 32:1057–1078 (1980).

5 A. Y. Barraud, Investigations autour de la fonction signe d'une matrice. Application a l'equation de Riccati, *RAIRO Automat./Systems Anal. and Control* 13:335–368 (1979).

6 A. Y. Barraud, Produit étoile et fonction signe de matrice. Application à l'equation de Riccati dans le cas discret, *RAIRO Automat./Systems Anal. and Control* 14:55–85 (1980).

7 A. Beavers and E. Denman, A new solution method for quadratic matrix equations, *Math. Biosci.* 20:135–143 (1974).

8 G. J. Bierman, Computational aspects of the matrix sign function to the ARE, Report, Factorized Estimation Applications, Inc., 7017 Deveron Ridge Rd., Canoga Park, Calif. 91301, 1984.

9 R. Byers, Hamiltonian and Symplectic Algorithms for the Algebraic Riccati Equation, Ph.D. Thesis, Cornell Univ., Ithaca, N.Y., 1983.

10 E. Denman and A. Beavers, The matrix sign function and computations in systems, *Appl. Math. Comput.* 2:63–94 (1976).

11 E. Denman and J. Layva-Ramos, Spectral decomposition of a matrix using the generalized sign matrix, *Appl. Math. Comput.* 8:237–250 (1981).

12 N. Higham, Computing the polar decomposition—with applications, Numerical Analysis Report No. 94, Dept. of Mathematics, Univ. of Manchester, Manchester M13 9PL, England, Nov. 1984.

13 S. Hammarling, Newton's method for solving the algebraic Riccati equation, Technical Report DICT 12/82, National Physics Lab., Teddington, Middlesex, 1982.

14 J. L. Howland, The sign matrix and the separation of matrix eigenvalues, *Linear Algebra Appl.* 49:221–232 (1980).

15 D. Kleinman, On an iterative technique for Riccati equation computations, *IEEE Trans. Automat. Control* 13:114–115 (1968).
16 H. Kwaadernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, New York, 1972.
17 A. Laub, A Schur method for solving algebraic Riccati equations, *IEEE Trans. Automat. Control* 24:913–925 (1979).
18 C. Lawson and R. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, N.J., 1974.
19 W. Levine and M. Athans, On the optimal error regulation of a string of moving vehicles, *IEEE Trans. Automat. Control* 11:355–361 (1966).
20 J. Dongarra, C. Moler, J. Bunch, and G. Stewart, LINPACK *Users' Guide*, SIAM, Philadelphia, 1979.
21 L. Lupas and C. Popeea, Solution of differential matrix equations by the matrix sign function, *Rev. Roumaine Sci. Techn. Sér. Électrotechn. Énergét.* 22:89–97 (1976).
22 R. Matheys, Stability analysis via the extended matrix sign function, *Proc. Inst. Elec. Engrs.* 125:241–243 (1978).
23 J. Potter, Matrix quadratic solutions, *SIAM J. Appl. Math.* 14:496–501 (1966).
24 J. Roberts, Linear model reduction and solution of algebraic Riccati equations by use of the sign function, Engineering Report, CUED/B-Control, Tr-13, Cambridge Univ., Cambridge, England, 1971.
25 J. Roberts, Linear model reduction and solution of the algebraic Riccati equation by the use of the sign function, *Internat. J. Control* 32:677–687 (1980).
26 T. Smith, J. Boyle, B. Garbow, Y. Ikebe, V. Kema, and C. Moler, EISPACK *Guide*, Springer, New York, 1974.
27 G. W. Stewart, Error and perturbation bounds for subspaces associated with certain eigenvalue problems, *SIAM Rev.* 15:727–764 (1973).
28 J. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon, Oxford, 1965.
29 W. Wonham, *Linear Multivariable Control: A Geometric Approach*, Springer, New York, 1979.
30 R. Yoo and E. Denman, Uncoupling of constant coefficient canonical differential equations of optimal control, Report DEE, Univ. of Houston, Houston, Tex., 1974.

Received 18 December 1984; revised 26 December 1985