S. G. Greenberg
Y. Bard

An Improved Method for Designing Optimal Linear Compensators

Abstract: In linear dynamical systems with white Gaussian plant and observation noise and quadratic cost criteria the well-known separation theorem of stochastic control holds. The result is that the determination of the system matrix for the compensator depends on the solution of two Riccati equations, one arising from a deterministic regulator problem and the other from a filtering problem. It is shown in this paper that if the given system is single-input, single-output and time-invariant, one can achieve substantial savings in computation time. Assuming controllability and observability and using the standard controllable representation of the system matrix, we show that at each iteration step of the solution of the algebraic matrix Riccati equation by Newton's method the number of variables to be solved for reduces from the customary $n(n+1)/2$ to n. Moreover, the number of operations to determine these n variables is on the order of $n^3/16$ as opposed to $n^3/3$ in ordinary matrix inversion.

The observability matrix is used as a similarity transformation so that the Riccati equation for the filtering problem is placed in the same format and the above procedure may be used again. The results obtained are easily extended to the case in which the system is either single-input or single-output. The system matrix of a fifteenth-order compensator was determined using 0.92 seconds of computer time on the IBM System/360, Model 67.

1. Introduction

In stochastic control problems specified by linear dynamics, quadratic cost criteria and Gaussian white plant and observation noise, the separation theorem[1] of stochastic control may be invoked with the result that the optimal control is determined by first estimating the state of the system and then using the estimated state in a deterministic optimal control problem. In order to estimate the state in the case of a time-invariant system one must design a Kalman filter[2], a procedure that involves as a step the solution of an algebraic matrix Riccati equation for the covariance matrix of the estimation error. The resulting deterministic control problem requires the solution of an algebraic matrix Riccati equation for the optimal feedback gain matrix. The solution of these two Riccati equations comprises the bulk of the computation in this class of problems.

Several methods for computing the solution of the Riccati equation have been put forward in recent years. These generally consist of either asymptotic integration methods[3], Newton's method[4], or direct methods based on computing the eigenvalues and eigenvectors of the Hamiltonian system[5,6]. In the sequel we shall show that if the system is linear, time-invariant and single-input or single-output, then a computational algorithm exists for the solution of the Riccati equation which utilizes an order of magnitude less computation time than is required by conventional methods. This algorithm fits within the category of Newton's method, but the complexity of the computation at each iteration step is substantially reduced over that of the traditional Newton's method.

Let us consider a system that is both single-input and single-output.

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{b}[u(t) + \xi(t)],$$

$$z(t) = \mathbf{c}'\mathbf{x}(t) + \theta(t) = y(t) + \theta(t), \tag{1}$$

where $\mathbf{x}$ is the n-vector state of the system, u is the input signal, y is the output, z is the measured output, $\mathbf{A}$ is a constant $n \times n$ matrix, $\mathbf{b}$ and $\mathbf{c}$ are constant n-vectors, $\xi(t)$ is the Gaussian white plant disturbance noise (with variance Ξ), and $\theta(t)$ is the Gaussian white observation noise (with variance Θ). The prime notation indicates the transpose of a matrix here and throughout.Suppose we are given a cost criterion of the form

$$J = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} [qy^2(t) + u^2(t)]dt,$$

where q is a positive weighting constant.

The optimal control is then given by

$$u^*(t) = -\mathbf{b}'\mathbf{K}\hat{\mathbf{x}}(t),$$

where $\mathbf{K}$ is the positive definite solution of the matrix Riccati equation

$$\mathbf{KA} + \mathbf{A}'\mathbf{K} - \mathbf{Kbb}'\mathbf{K} + q\mathbf{cc}' = \mathbf{0} \tag{2}$$

and $\hat{\mathbf{x}}(t)$ is the state of the system as estimated by a Kalman filter. The filter is specified by

$$\frac{d\hat{\mathbf{x}}(t)}{dt} = \left[\mathbf{A} - \frac{1}{\Theta}\mathbf{\Sigma cc}' - \mathbf{bb}'\mathbf{K}\right]\hat{\mathbf{x}}(t) + \frac{1}{\Theta}\mathbf{\Sigma c}z(t),$$
$$u(t) = -\mathbf{b}'\mathbf{K}\hat{\mathbf{x}}(t), \tag{3}$$

where $\mathbf{\Sigma}$, the $n \times n$ error covariance matrix is the solution of the matrix Riccati equation

$$\mathbf{\Sigma A}' + \mathbf{A\Sigma} - \frac{1}{\Theta}\mathbf{\Sigma cc}'\mathbf{\Sigma} + \Xi\mathbf{bb}' = \mathbf{0}. \tag{4}$$

It should be noted that the system given by Eqs. (3) is truly a feedback compensator for the system given by Eqs. (1) in that the input to the filter is the output (corrupted by noise) of the plant and the output of the filter is the input to the plant. Another point to note in Eqs. (3) is that we do not need the complete matrices $\mathbf{K}$ and $\mathbf{\Sigma}$, but only the n-dimensional vectors $\mathbf{b}'\mathbf{K}$ and $\mathbf{\Sigma c}$.

An algorithm using Newton's method for solving Riccati equations corresponding to single-input systems is discussed in Section 2. The use of this algorithm in solving for optimal compensators is presented in Section 3.

2. Numerical solution of the Riccati equation

Let us assume that the system described by Eqs. (1) is in standard controllable form; that is, the system matrix $\mathbf{A}$ is of the form

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ a_1 & a_2 & a_3 & a_4 & \cdots & a_{n-1} & a_n \end{bmatrix}$$

and the transpose of the $\mathbf{b}$-vector is

$$\mathbf{b}' = [0\ 0 \cdots 0\ 1].$$

This is not a rigid assumption, since single-input, single-output systems are customarily specified by transfer functions (a transfer function vector in the case of multiple outputs) whose denominator coefficients form the last row of $\mathbf{A}$ and whose numerator coefficients form the output vector (or matrix in the case of multiple outputs).

Let us consider solving the following matrix Riccati equation

$$\mathbf{KA} + \mathbf{A}'\mathbf{K} - \mathbf{Kbb}'\mathbf{K} + \mathbf{Q} = \mathbf{0}, \tag{5}$$

where $\mathbf{Q}$ is any non-negative definite matrix. If we apply Newton's method to the solution of this equation, the ith iterate $\mathbf{K}_i$ must satisfy

$$\mathbf{K}_i\mathbf{A}_i + \mathbf{A}'_i\mathbf{K}_i = -\mathbf{Q}_i, \tag{6}$$

where $\mathbf{A}_i = \mathbf{A} - \mathbf{bb}'\mathbf{K}_{i-1}$ and $\mathbf{Q}_i = \mathbf{Q} + \mathbf{K}_{i-1}\mathbf{bb}'\mathbf{K}_{i-1}$.

Since $\mathbf{K}$ is symmetric we must determine $n(n+1)/2$ values. However, at each iteration step only n variables, the elements of the last row of $\mathbf{K}$, play a role. To see this, note that

$$\mathbf{A}_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ \theta_1{}^i & \theta_2{}^i & \theta_3{}^i & \cdots & \theta_{n-1}{}^i & \theta_n{}^i \end{bmatrix},$$

where $\theta_k{}^i = a_k - [\mathbf{K}_{i-1}]_{n,k}$.

That is, only the last row changes from iteration to iteration, with the kth element of the last row depending only on the kth element of the last row of the previous iteration matrix. Similarly, the matrix $\mathbf{Q}_i$ is simply the sum of the matrix $\mathbf{Q}$ and the outer product of the last row of $\mathbf{K}_{i-1}$ with itself. Thus, it is easily seen that if a simple manner of determining *only* the last row of the matrix at each iteration can be devised, then substantial savings in computer time may be achieved.

To show that one can use this reduced algorithm, we must examine the set of $n(n+1)/2$ linear equations in the $n(n+1)/2$ unknown elements of $\mathbf{K}_i$ obtained from Eq. (6). Obviously, by brute force one could invert the $n(n+1)/2 \times n(n+1)/2$ matrix to solve for the $n(n+1)/2$ unknown variables, but we shall see that we need only consider an $n \times n$ matrix at each iteration step. The tableau for this matrix and its right-hand side is given in Fig. 1 for a fifth-order example. The equations are numbered according to the corresponding indices of the matrix Eq. (6). The variables are numbered according to their true indices, that is, variable kl is $[\mathbf{K}_i]_{kl}$.

It is clear how to generalize the fifth-order tableau given in Fig. 1 to any dimension. A detailed derivation of the form of this tableau as well as the results in the remainder of this section may be found in Ref.[7]. Clearly, the first $n(n-1)/2$ columns remain unchanged from iteration to iteration. This suggests that the corresponding variables can be eliminated symbolically once and for all, and this is done by means of Gauss-Jordan pivots on the bold-face elements in Fig. 1. After performing this pivoting, an $n \times n$ tableau of the form given in Fig. 2 (this time for a seventh-order example) is obtained for the unknown last row of $\mathbf{K}_i$. This system of n equations is readily solved by means of Gaussian elimi-

Figure 1 Initial tableau.

Indices for equations	11	21	22	31	32	33	41	42	43	44	51	52	53	54	55	*Right-hand side*
	Indices for variables															
11											$2\theta_1$					Q_{11}
21	1										θ_2	θ_1				Q_{21}
22		2										$2\theta_2$				Q_{22}
31		1									θ_3		θ_1			Q_{31}
32			1	1								θ_3	θ_2			Q_{32}
33					2								$2\theta_3$			Q_{33}
41				1							θ_4			θ_1		Q_{41}
42					1		1					θ_4		θ_2		Q_{42}
43						1		1					θ_4	θ_3		Q_{43}
44									2					$2\theta_4$		Q_{44}
51							1				θ_5				θ_1	Q_{51}
52								1			1	θ_5			θ_2	Q_{52}
53									1			1	θ_5		θ_3	Q_{53}
54										1			1	θ_5	θ_4	Q_{54}
55														2	$2\theta_5$	Q_{55}

Figure 2 Final reduced tableau after pivoting.

K_{71}	K_{72}	K_{73}	K_{74}	K_{75}	K_{76}	K_{77}	*Right-hand side*
θ_1							$\frac{1}{2} Q_{11}$
$-\theta_3$	θ_2	$-\theta_1$					$\frac{1}{2} Q_{22} - Q_{31}$
θ_5	$-\theta_4$	θ_3	$-\theta_2$	θ_1			$\frac{1}{2} Q_{33} - Q_{42} + Q_{51}$
$-\theta_7$	θ_6	$-\theta_5$	θ_4	$-\theta_3$	θ_2	$-\theta_1$	$\frac{1}{2} Q_{44} - Q_{53} + Q_{62} - Q_{71}$
	1	θ_1	$-\theta_6$	θ_5	$-\theta_4$	θ_3	$\frac{1}{2} Q_{55} - Q_{64} + Q_{73}$
			-1	$-\theta_7$	θ_6	$-\theta_5$	$\frac{1}{2} Q_{66} - Q_{75}$
					1	θ_7	$\frac{1}{2} Q_{77}$

nation. It can be shown that the number of multiplications performed in this procedure for the special type of matrix in Fig. 2 is of the order $n^3/16$ as opposed to $n^3/3$ in the case of a general matrix.

Thus, we see that at each iteration step the computation has been reduced in complexity. The test for convergence is based on the maximum absolute difference between the elements of the present iterate and those of the previous iterate. After convergence has been achieved, the remaining elements of the matrix $\mathbf{K}$ may be determined by simple substitution, the expressions for which will not be given, since, as will be seen in the next section, only the last row is needed to determine the optimal compensator.

Convergence in Newton's method is guaranteed only if certain conditions are satisfied by the initial guess. These conditions are that the initial guess $\mathbf{K}_0$ be positive definite and that the matrix $\mathbf{A} - \mathbf{bb}'\mathbf{K}_0$ have all its eigenvalues in the left half-plane. Since, in the above method, we need only supply the initial guess of a vector, "positive definiteness" is guaranteed by choosing the nnth element positive. It is also a relatively easy matter to choose the initial vector to satisfy the stability requirement. If we denote the initial guess as $\mathbf{k}_0$ and let

$$\theta_i = a_i - k_{0i}, \qquad i = 1, \cdots, n,$$

then the θ_i's are the coefficients of the closed-loop system matrix characteristic function

$$p(x) = -\theta_1 - \theta_2 x \cdots - \theta_n x^{n-1} + x^n.$$

Let us choose to have $n-1$ roots at $-\Psi$ and one root at $-\phi$, where Ψ and ϕ are positive real numbers. Then clearly

$$\theta_1 = -\phi\Psi^{n-1},$$

$$\theta_i = -\left[\phi\Psi^{n-i}\binom{n-1}{i-1} + \binom{n-1}{i-2}\Psi^{n-i+1}\right], \quad i = 2, \cdots, n.$$

To satisfy the requirement that k_{0n} be positive, we require that

$$a_n + \phi + \Psi(n-1) > 0.$$

In practice we have used $\Psi = 1$ and $\phi = 1 + \max\,[0, (1-n) - a_n]$ with good results.

A version of the above algorithm has been programmed as a subroutine in Fortran IV for the single-input, single-output case. The call to this subroutine requires four arguments: 1) the last row of the system matrix $\mathbf{A}$ (assumed to be in standard controllable form), 2) the output vector $\mathbf{c}$, 3) the dimension n, and 4) the last row of the optimal feedback matrix, which is returned by the subroutine. (It should be noted that the full feedback matrix may be obtained by a simple procedure of back substitution.) In the next section we shall see how this subroutine can be used effectively in the optimal compensator problem.

3. Computation of the optimal compensator

To determine the system matrix, control vector and output vector of the optimal compensator given by Eqs. (3), we must determine the vectors $\mathbf{b}'\mathbf{K}$ and $\mathbf{\Sigma c}$. It is a straightforward matter to compute $\mathbf{b}'\mathbf{K}$ in the standard controllable case. Clearly, $\mathbf{b}'\mathbf{K}$ is the last row of the matrix solution of Eq. (2). This vector is obtained directly by calling the computational subroutine described in the previous section with an output vector $\mathbf{c}_1 = q^{1/2}\mathbf{c}$ as an argument.

It is not as straightforward a matter to compute $\mathbf{\Sigma c}$. It should be noted that Eq. (4) is not in a form such that the algorithm of Section 2 may be applied. Let us define a transformation matrix $\mathbf{T}$ with the properties

$$\mathbf{TA}' = \mathbf{AT}, \tag{7}$$

$$\frac{1}{\sqrt{\Theta}}\mathbf{T}'\mathbf{c} = \mathbf{b}. \tag{8}$$

Now, let $\mathbf{\Sigma}_1$ be some matrix such that

$$\mathbf{\Sigma} = \mathbf{T\Sigma}_1\mathbf{T}'. \tag{9}$$

Substituting Eq. (9) into Eq. (4) and using the symmetry of $\mathbf{T}$ and Eqs. (7) and (8), we find that $\mathbf{\Sigma}_1$ satisfies

$$\mathbf{\Sigma}_1\mathbf{A} + \mathbf{A}'\mathbf{\Sigma}_1 - \mathbf{\Sigma}_1\mathbf{bb}'\mathbf{\Sigma}_1 + \mathbf{c}_2\mathbf{c}'_2 = \mathbf{0}, \tag{10}$$

where

$$\mathbf{c}_2 = (\Xi/\Theta)^{1/2}\mathbf{c}.$$

This is precisely the form the subroutine requires. Moreover, since we really desire the vector $\mathbf{\Sigma}_1\mathbf{c}$, we have the beneficial result that

$$\begin{aligned}\mathbf{\Sigma c} &= \mathbf{T\Sigma}_1\mathbf{T}'\mathbf{c} = \sqrt{\Theta}\mathbf{T\Sigma}_1\mathbf{b} \\ &= \sqrt{\Theta}\mathbf{T}\boldsymbol{\sigma}_1,\end{aligned} \tag{11}$$

where $\boldsymbol{\sigma}_1$ denotes the last row of $\mathbf{\Sigma}_1$, the vector that is returned by the subroutine.

The transformation matrix may be determined by standard techniques for transforming to standard controllable form. If the matrix $\mathbf{T}$ is given by

$$\mathbf{T}' = [\mathbf{t}_1, \cdots, \mathbf{t}_n],$$

we have the recursion relation

$$\mathbf{t}_i = \mathbf{A}t_{i-1}, \qquad i = 2, 3, \cdots, n$$

and $\mathbf{t}_1$ is given by

$$\mathbf{t}_1 = \mathbf{L}^{-1}\,\mathbf{b},$$

where $\mathbf{L}$ is the observability matrix for the system (c.f. Ref.[8], ch. 4),

$$\mathbf{L}' = [\mathbf{c}, \mathbf{A}'\mathbf{c}, \mathbf{A}'^2\mathbf{c}, \cdots, \mathbf{A}'^{n-1}\mathbf{c}].$$

However, in the special case that we consider, the matrix is to be transformed into its own transpose and computational simplification can be gained. The matrices $\mathbf{AT}$ and $\mathbf{TA}'$ have the forms

$$\mathbf{AT} = \begin{bmatrix} \mathbf{t}'_2 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{t}'_n \\ \hline \boldsymbol{\alpha}'\mathbf{T} \end{bmatrix} \qquad \text{and} \qquad \mathbf{TA}' = [\mathbf{t}_2 \cdots \mathbf{t}_n \mid \mathbf{T}'\boldsymbol{\alpha}],$$

where $\boldsymbol{\alpha}' = (a_1, a_2, \cdots, a_n)$. From the upper left $(n-1) \times (n-1)$ block we obtain the recursion formula

$$T_{ij} = T_{i-1,j+1}, \qquad i = 2, \cdots, n-1; j = i, \cdots, n-1. \tag{12}$$

From Eq. (12) we may deduce that $\mathbf{T}$ is symmetric. This follows from the fact that

$$\begin{aligned} T_{i+1,\,j-1} &= T_{ij} \\ T_{i+2,\,j-2} &= T_{ij} \\ &\ \,\vdots \\ T_{i+k,j-k} &= T_{ij}. \end{aligned}$$

Indeed, if we choose $k = j - i$, then

$$T_{ji} = T_{ij}$$

and symmetry is proved.

From the first $n-1$ elements of the last row (or column) we have

$$T_{in} = \boldsymbol{\alpha}'\mathbf{t}_{i-1}. \tag{13}$$

Given recursion formulas (12) and (13), all elements may be evaluated once $\mathbf{t}_1$, the first row of $\mathbf{T}$ is determined

```
OPTIMAL COMPENSATOR OF DIMENSION  6  WITH STATE WEIGHTING FACTOR    1.0000
MEASUREMENT NOISE VARIANCE    1.0000    AND PLANT NOISE VARIANCE    1.0000
THE LAST ROW OF THE SYSTEM MATRIX IS
   0.100000D 01    0.200000D 01    0.300000D 01    0.400000D 01    0.400000D 01    0.200000D 01
THE OUTPUT VECTOR OF THE SYSTEM IS
   0.400000D 01    0.600000D 01    0.200000D 01    0.800000D 01    0.450000D 01    0.720000D 01
    SYSTEM MATRIX OF COMPENSATOR
   0.248758D 01    0.473138D 01    0.124379D 01    0.497517D 01    0.279853D 01    0.447765D 01
   0.162126D 01    0.243189D 01    0.181063D 01    0.324252D 01    0.182392D 01    0.291827D 01
  -0.509856D 00   -0.764784D 00   -0.254928D 00   -0.197115D-01   -0.573588D 00   -0.917740D 00
  -0.179076D 01   -0.268614D 01   -0.895380D 00   -0.358152D 01   -0.101460D 01   -0.322337D 01
  -0.194838D 01   -0.292257D 01   -0.974191D 00   -0.389676D 01   -0.219193D 01   -0.250709D 01
  -0.105071D 02   -0.258462D 02   -0.331113D 02   -0.481154D 02   -0.314427D 02   -0.220912D 02
    COMPENSATOR INPUT VECTOR
  -0.621896D 00   -0.405316D 00    0.127464D 00    0.447690D 00    0.487096D 00    0.159600D 01
    COMPENSATOR OUTPUT VECTOR
  -0.512311D 01   -0.182702D 02   -0.329193D 02   -0.393474D 02   -0.282607D 02   -0.126001D 02
```

Figure 3 Printout for sixth-order example.

by inverting the observability matrix as shown above. As might be expected, in this special case the observability matrix can be calculated more simply than in the general case. Indeed, the first row of the observability matrix is the vector **c** and succeeding rows are given by

$$L_{i1} = a_1 L_{i-1,\,n}, \qquad i = 2, \cdots, n, \tag{14}$$
$$L_{ij} = a_j L_{i-1,\,n} + L_{i-1,\,j-1}, \qquad j = 2, \cdots, n.$$

After **L** has been computed, the first row of the transformation matrix is determined by

$$\mathbf{t}_1 = \sqrt{\theta}\mathbf{L}^{-1}\mathbf{b} \tag{15}$$

and remaining elements are found by using Eqs. (12) and (13) and the symmetry relation.

As the dimension becomes higher, the matrix **L** becomes more ill conditioned. It has been found that when the dimension is 16 or higher, there is an abrupt loss in accuracy in inverting **L,** even when double-precision arithmetic is used. This corresponds physically to the fact that the higher the dimension of a system, the more difficult it is to control (or observe) it by a single input (or output).

The computational steps involved in determining the optimal compensator may be summarized as follows:

1.) Compute the compensator output vector $\mathbf{b}'\mathbf{K}$ by entering the Riccati equation solution algorithm with $\mathbf{c}_1 = q^{1/2}\mathbf{c}$,
2.) compute the transformation **T** which satisfies Eqs. (7) and (8) by using Eqs. (12), (13), (14) and (15),
3.) compute the last row, $\boldsymbol{\sigma}_1$, of the matrix $\boldsymbol{\Sigma}_1$ by entering the Riccati equation solution algorithm with $\mathbf{c}_2 = (\Xi/\Theta)^{1/2}\mathbf{c}$,
4.) compute the compensator input vector $(1/\Theta)\,\boldsymbol{\Sigma}\,\mathbf{c} = (1/\sqrt{\Theta})\mathbf{T}\boldsymbol{\sigma}_1$, and
5.) compute the compensator system matrix as indicated in Eq. (3).

A Fortran IV program[9] has been written to perform these computations and the results of a sixth-order problem are given in Fig. 3. The execution time for this example was 0.27 seconds on an IBM System/360 Model 67. A fifteenth-order example was run with an execution time of 0.92 seconds, a decrease by a factor of more than ten in comparison with the execution time of conventional methods such as the eigenvalue method and the generalized Newton's method.

4. Conclusions

The computational procedure described above demonstrates the savings that can be achieved over standard Riccati equation solution techniques for solving optimal single-input, single-output problems. Computational complexity is reduced from order $n(n + 1)/2$ to order n. It is shown that the Riccati equation associated with the deterministic optimal control problem is in the appropriate form for application of the simplified solution algorithm. In the case of the Riccati equation associated with the filtering problem, a transformation must be made before the algorithm is applied. The determination of this transformation has been shown to be relatively simple and straightforward.

The results of Section 2 can be extended to single-output, multi-input systems quite readily[7] by transforming the system equation to the standard observable form and solving a transformed Riccati equation for the inverse of the **K** matrix so that the results of Section 3, after obvious modifications, can be extended to include systems that are either single-input or single-output.

References

1. W. M. Wonham, "Random Differential Equations in Control Theory," *Probabilistic Methods in Applied Mathematics*, Vol. II, A. T. Bharucha-Reid, Ed., Academic Press, New York, 1969.
2. R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," *Trans. ASME, J. Basic Eng.* **83,** 95 (1961).
3. R. E. Kalman and T. S. Englar, *A User's Manual for ASP-C*, NASA-CR-475, Ames Research Center, 1965.
4. D. L. Kleinman, "On the Linear Regulator Problem and the Matrix Riccati Equation," M.I.T. Electronics System Laboratory Report No. ESL-R-271, June, 1966.
5. A. G. MacFarlane, "An Eigenvector Solution of the Optimal Linear Regulator Problem," *J. Elect. Control* **14,** 643 (1963).
6. J. E. Potter, "Matrix Quadratic Solutions," *J. SIAM Appl. Math.* **14,** 496 (1966).
7. Y. Bard and S. G. Greenberg, "Solution of the Algebraic Matrix Riccati Equation for Single-input Systems in Standard Controllable Form," IBM Cambridge Scientific Center Report, 1970.
8. M. Athans and P. L. Falb, *Optimal Control: An Introduction to the Theory and its Applications*, McGraw-Hill Book Co., New York, 1966.
9. The Fortran IV program and its subroutines are for IBM internal use only.

Received February 16, 1971

The authors are located at the Data Processing Division Scientific Center, Cambridge, Massachusetts. This paper was presented at the International IEEE Conference, Oaxtepec, Mexico, January 1971.