UNIVERSITY OF CONNECTICUT

SCHOOL OF ENGINEERING

STORRS, CONNECTICUT

FACTORS INFLUENCING CONVERGENCE OF A LEARNING SYSTEM

G. E. Zajac

Department of Electrical Engineering

May 1970

This work has been sponsored by the
National Aeronautics and Space Administration
Research Grant NGL 07-002-002

ACKNOWLEDGEMENT

The author wishes to acknowledge the valuable assistance and guidance given by David P. Lindorff, Charles H. Knapp, and Ralph J. Kochenburger during the research conducted for this thesis.

This work was supported under Research Grant NSG 07-002-002 of the National Aeronautics and Space Administration.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

ABSTRACT

This thesis will investigate several factors affecting convergence of a learning control system proposed by Scott [1]. As a result of these investigations an improvement to the algorithm is developed which is capable of reducing convergence time by a factor of 1/(N+1) where N is the order of the system.

INTRODUCTION

When Scott [1] proposed his algorithm for a learning control system, the choice of measurement rays was left open for investigation. This thesis will investigate the effect of the choice of these rays upon convergence of the algorithm. Then a modification to the algorithm will be developed which will provide more rapid and reliable convergence for any general set of N linearly dependent rays. Finally, the modified algorithm will be applied to a third order example.

Since Scott's report forms a basis for this thesis, the theory behind his learning system should be discussed. His method differs from previous work [2,3] in that the state space need not be quantized; thus, the memory requirements and learning times are greatly reduced.

The algorithm will learn an optimum control for a linear stationary system described by a vector difference equation

$$x(i+1) = \phi x(i) + hu(i) \tag{1.1}$$

where x(i) is an N dimensional vector and u(i) is a scalar. The system performance is measured by the quadratic cost functional

$$\sum_{i=0}^{\infty} x'(i+1)\, Q\, x(i+1) + \alpha\, u^2(i) \tag{1.2}$$

defined by the N x N positive semi-definite matrix Q and the positive scalar α. Assuming no constraints on the final point $x(\infty)$, the optimal control is known to be the linear control law

$$u(i) = k_*' x(i) \tag{1.3}$$

where

$$k_*' = - \frac{h' R \phi}{h'Rh+\alpha} . \tag{1.4}$$

R is the positive definite symmetric matrix determined by iterating the discrete Ricatti equations backwards in time until a steady state solution is obtained.

Due to ignorance of h and ϕ, k_* is not known. However, since the optimal control is known to be a linear function of the states, it is reasonable to assume a linear control law

$$u(i) = k' x(i) \tag{1.5}$$

and minimize a subgoal with respect to the vector k. However, the subgoal must be an explicit function of the gain vector k, and it must yield the control vector (1.4).

For an N state system, a subgoal was developed of the form

$$SG(k) = \sum_{i=1}^{N} SG_i(k) \tag{1.6}$$

where

$$SG_i(k) = \frac{x'(t_i+T)\, Rx(t_i+T)+\alpha u^2(t_i)}{x'(t_i)Bx(t_i)} .$$

R is a positive definite symmetric matrix, and α is a positive scalar, as defined in (1.4). B is an arbitrary positive definite matrix.

Although, as will be shown, this subgoal meets the previously stated requirements, it also has the important property of being independent of the magnitude of the state vectors $x(t_i)$. This is not evident by observing expression (1.6); however, by substituting the state equation (1.1) and the linear control (1.5) into the subgoal (1.6), the expression

$$SG(k) = \sum_{i=1}^{N} \frac{x'(t_i)[\phi' R\phi+2kh' R\phi+\beta kk']x(t_i)}{x'(t_i)\, Bx(t_i)} , \tag{1.7}$$

where $\beta=h'Rh+\alpha$, is obtained. Now it is readily seen that the subgoal is independent of the magnitude of each state vector $x(t_i)$. This is a

very important property of the subgoal because without it, it would be necessary to return the system to the same point in the state space every time a given term of the subgoal was to be evaluated. Because this subgoal is independent of the state-vector magnitudes, it is only necessary to cause the state of the system to reach each of N rays in the state space when measuring the subgoal. Since the rays can be made to be eigenvectors of the system, returning the state of the system to a ray in the state space is much simpler than forcing it to a specific point.

Assuming that the system has been forced to states on the given rays $(r_1, r_2, \ldots, r_N)$ at the respective points in time $(t_1, t_2, \ldots, t_N)$, these rays can be substituted for the state vectors $[x(t_1), x(t_2), \ldots, x(t_N)]$ in equation (1.7) without affecting the value of the subgoal. With this substitution, (1.7) becomes

$$SG(k) = \sum_{i=1}^{N} \frac{r_i'[\phi'R\phi+2kh'R\phi+\beta kk']r_i}{r_i' B r_i} . \qquad (1.8)$$

Noting that $k'r_i=r_i' k$, the subgoal can be rewritten in the form:

$$SG(k) = \beta k' M(r_1,\ldots, r_N)k+2k' M(r_1,\ldots, r_N) p + S(r_1,\ldots,r_N) \qquad (1.9)$$

where

$\beta= h'Rh+\alpha$, $M(r_1,\ldots,r_N) = \sum_{i=1}^{N} \frac{r_i r_i'}{r_i'Br_i}$ is an NxN matrix,

$p=\phi'Rh$ is an Nx1 column vector, and

$S(r_1,\ldots,r_N) = \sum_{i=1}^{N} \frac{r_i'\phi'R\phi r_i}{r_i'Br_i}$ is a scalar.

Equation (1.9) clearly shows that the subgoal is a quadratic function of the gain vector k.

The extrema of the subgoal are at the points at which the gradient

$$\nabla_k \, SG(k) = 2M(r_1,\ldots, r_N) \, [\beta k+p] \tag{1.10}$$

equals zero. If the subgoal is to have an extremum at a unique point, the matrix $M(r_1,\ldots, r_N)$ must be non-singular. Furthermore, if $M(r_1,\ldots,r_N)$ is also positive-definite, then the extremum is a minimum. It was proved by Scott that, if the rays $(r_1,\ldots,r_N)$ are linearly independent, the matrix $M(r_1,\ldots,r_N)$ is non-singular and positive definite. Consequently, if the rays are linearly independent, the subgoal has a minimum at the unique point

$$k = - p/\beta = - \frac{\phi' Rh}{h' Rh+\alpha} \, . \tag{1.11}$$

Thus, minimizing the subgoal with respect to the vector k yields the desired result, (1.4). Furthermore, this point is uniquely defined by R, α, and the system parameters, independent of the choice of the linearly independent set of rays $(r_1,r_2,\ldots,r_N)$.

Earlier in this development, an assumption was made that the system could be forced to attain a state lying along any given ray in the state space. Thus, it is necessary to implement a controller that can force the system to any given ray a finite number of times to permit measurement of the subgoal (1.6) during the learning process. State feedback can be employed to force the system to behave in this manner. For the system described by equation (1.1), the closed-loop system matrix is the NxN matrix

$$A = \phi+hk' \tag{1.12}$$

Since this is a linear stationary system, its response is known to be a linear combination of the eigenvectors of the form

$$x(nT) = \sum_{i=1}^{N} \alpha_i \lambda_i^n e_i \tag{1.13}$$

where

(i) λ_i is an eigenvalue of A,

(ii) e_i is an eigenvector corresponding to λ_i, and

(iii) α_i is determined by the initial state x(0).

From the preceding development it is easily seen that, if the magnitudes of all but one of the eigenvalues, say λ_1, are forced to be much less than one, then the system response will eventually approach the vector $\lambda_1^n e_1$. The rate of convergence will depend on the relative magnitudes of the other N-1 eigenvalue. Since ϕ and h are fixed, the magnitudes of the eigenvalues of matrix A can be set to any values by the appropriate choice of the gain vector k. Since the system must be forced to each of N given rays in the state space during evaluation of the subgoal (1.6), N feedback gain vectors must be computed. To force the system to the ith ray, the gain vector k of the linear control law (1.5) would be replaced by k_i. Thus, to measure the subgoal starting from any point x(0) in the state space, a controller must be implemented which will follow this sequence:

1) Apply a gain vector k_1 that will force the system to converge on ray r_1.
2) When it is ascertained that the system is on the ray r_1 at time t_1, measure the state $x(t_1)$ and restore the linear control law (1.5).
3) Allow the system to run one time period, T, under the linear control law.
4) At the end of the period at time $t_1 + T$, measure the state $x(t_1 + T)$; and with the value of the state $x(t_1)$ obtained in step 2, compute the partial subgoal $SG_1(k)$ using the form given in (1.6)

Repeating the above steps another N-1 times and then summing the N partial subgoals will yield the value of the subgoal.

Having reviewed the method for measuring the subgoal, it remains to be shown how minimization of the subgoal can be accomplished so as to minimize the convergence time.

EFFECT OF RAY SELECTION UPON CONVERGENCE

It is shown that the rate of convergence of the subgoal is dependent on the choice of the measurement rays, and that an orthogonal set of rays leads to the most favorable convergence time. To determine the effect of these rays, a second order system with various sets of rays were simulated. To eliminate the necessity of computing feedback gain vectors for several rays and also to eliminate any errors due to the system not having settled on a given ray, the system was set precisely upon the respective ray, one time period prior to evaluation of the associated term in the subgoal.

Also to simplify matters, the matrices R and B of the subgoal (1.6) were chosen to be identity matrices,* and the scalar α was chosen to be 1.0; thus, the subgoal simplifies to

$$SG(k) = \sum_{i=1}^{N} SG(k) = \sum_{i=1}^{N} \frac{x'(t_i+T)x(t_i+T)+u^2(t_i)}{x'(t_i)x(t_i)}$$

A method proposed by Powell [4] was chosen first as the technique to minimize the subgoal since it not only requires no derivatives, but also guarantees convergence upon the minimum of a quadratic function, such as the subgoal, within a specified number of iterations if the minimum in each search direction can be computed with sufficient accuracy. Appendix A covers the iterative steps of this minimization technique.

---

* It has been shown that minimizing the subgoal (1.6) yields a solution (1.4) for k. For this to be a solution to the minimization of (1.2) it is required that R be a solution to the Ricatti equations. However, for the purposes of studying convergence, R may be any positice definite symmetric matrix, and in particular the identity matrix as used here.

The minimum of the function along any search direction in the parameter space k, i.e.,

$$\min_{\lambda} f(k_0 + \lambda k_\Delta), \tag{2.2}$$

was determined by computing the subgoal at three points and solving for the minimum on the basis of a quadratic fit through these points. The value of λ for which the function is a minimum along the line is given by

$$\lambda = (\alpha/2)\ \frac{(FN-FP)}{(FN+FP)}\ , \tag{2.3}$$

where

$$FN = f(k_0 - \alpha k_\Delta) - f(k_0)$$

$$FP = f(k_0 + \alpha k_\Delta) - f(k_0).$$

Since the subgoal is a quadratic in k, the minimum computed from (2.3) is assumed to be the actual minimum, although it may differ very slightly from the actual minimum due to round off errors in the calculations. Usually, these small errors are of no consequence; however, it will be shown later that there are instances when these usually insignificant errors can affect convergence.

The above minimization method, with initial search directions along the coordinate axes of the k space, was applied to the second-order system described by the equations

$$x(i+1) = \phi x(i) + hu(i) \tag{2.4}$$

where

$$\phi = \begin{vmatrix} 0.1 & 1 \\ 0 & 0 \end{vmatrix}\ ,\ h = \begin{vmatrix} 1 \\ 1 \end{vmatrix}\ .$$

The results of the optimization procedure with several sets of rays are summarized in Table 1. The initial gain vector in each case was [1,1].

The desired gain vector can be computed from (1.11) to be [-0.1/3, -1/3]. As can be seen from Table 1, an orthogonal set of rays, regardless of the orientation, always results in the most rapid convergence. These rays required only six measurements of the subgoal - three in each of two search directions - to locate the minimum of the function. In fact, a search in the ith coordinate direction of the k space was always sufficient to locate the ith component of the optimum gain vector. This indicates that the ith partial derivative of the subgoal is a function of only the ith gain, i.e., $\partial SG/\partial k_1=F(k_1)$, $\partial SG/\partial k_2=F(k_2)$, etc. For this to be true, the M matrix in the gradient of the subgoal (1.10) must be diagonal. From equation (1.9), it is recalled that

$$M(r_1,\ldots, r_N)=\sum_{i=1}^{N} \frac{r_i r_i'}{r_i' B r_i} . \tag{2.5}$$

In Appendix B, it is proved that M is not only diagonal, but an identity matrix, if the rays r_1, r_2, ..., r_N are orthogonal and B is an identity matrix.

The geometric interpretation of the contours of $SG(k)=c_1$ for a second-order system with orthogonal rays can be readily observed if the identity matrix, I, is substituted for matrix M in the subgoal equation (1.9). The equation reduces to the form

$$SG(k)=k_1^2+k_2^2+ak_1+bk_2+c=c_1 \tag{2.6}$$

which is recognized as an equation describing concentric circles, for various values of c_1.

As shown in Table 1, the convergence time in the case of non-orthogonal rays is greater than that for orthogonal rays, but is constant for most non-orthogonal rays, an exception being for the cases in which the angular separation of rays is less than 13 degrees. The constant

convergence time is directly attributable to the method of minimization since, as stated earlier, Powell's method [4] will locate the minimum of a quadratic function within a fixed number of iterations. For a second order equation, this means at most six searches and consequently, 18 subgoal measurements.

Nevertheless, there are two examples in Table 1 which did not converge within the predicted number of iterations. These are the examples with pairs of rays (1,1), (1,0.7), and (1,1), (1,0.8). Judging from the angular separation between the rays, it would seem that convergence should have been obtained since the example with rays (1,0) and (40,0) converged to the minimum although the angular separation of 1.4 degrees in this case is much smaller. The only apparent difference in these two cases is that one set of rays lies near one of the coordinate axes, whereas the sets causing difficulties lie approximately in the center of the first quadrant of the parameter space k.

To determine why very poor convergence, or lack of convergence, occurs when the rays are near the center of the quadrant, two examples will be given that have nearly equal angular separations. For the example with lack of convergence, the rays (1,1) and 1,0.8) will be used; and the example with good convergence, the rays (1,0) and (10,1) will be used. The respective angular separations are 6.3 degrees and 5.7 degrees. Since the M matrix will be useful in the following analysis, it will be computed for each case from equation (1.9). For the rays (1,1) and (1,0.8),

$$M = \begin{vmatrix} 1.110 & 0.988 \\ 0.988 & 0.890 \end{vmatrix}; \qquad (2.7)$$

and for the rays (1,0) and 10,1),

$$M = \begin{vmatrix} 1.990 & 0.099 \\ 0.099 & 0.0099 \end{vmatrix} . \qquad (2.8)$$

Remembering that no solution exists for gradient equation (1.10) when M is singular, it appears reasonable that (2.7) might be nearly singular. Computing the determinant of (2.7) yields a value of 0.0117, however, the determinant of (2.8) has a somewhat smaller value of 0.0099. Thus, near singularity of M does not appear to be the cause of no convergence with the rays (1,1) and (1,0.8).

A remaining factor capable of affecting convergence is the shape and skewing of the contours of the subgoal function. Since M is positive definite for any set of N linearly independent rays, all contours of the subgoal (1.9) will in general be ellipsoids. Although the angle of skewing between the coordinate axes and the axes of the elliptical contours can be readily computed for a second-order system, it will be more informative actually to plot one contour for each pair of rays. First, the vector equation of the subgoal (1.9) must be reduced to scalar form to facilitate solution of contour points. Substituting in the subgoal equation (1.9) the values for ϕ and h given in (2.4) and the values of R, B and α given in (2.1), the rays (1,1) and 1,0.8) yield the quadratic equation

$$SG(k)=3.33\ k_1^2 + 5.928k_1k_2 + 2.67\ k_2^2+2.198\ k_1+1.9776k_2 +1.099. \qquad (2.9)$$

Similarly the rays (1,0) and (10,1) yield the equation

$$SG(k)=6.0k_1^2+0.6k_1k_2+0.03k_2^2+0.6k_1+0.04k_2+0.0496. \qquad (2.10)$$

A plot of the contours for each of these equations with SG(k) = 5.0 yields the ellipses shown in Fog. 1. It is seen that both ellipses are highly eccentric; however, the ellipse of equation(2.10) is also

skewed about 45 degrees from the coordinate axes. The reason for the lack of convergence, when rays (1,1) and (1,0.8) were selected, is now evident. It is well known that a long narrow valley with pronounced skewing causes minimization difficulties.

Although Powell's method theoretically guarantees convergence upon the minimum of a quadratic function when the 6 searches given in Table 1 are completed, it is required that the minimum in each search direction be determined accurately; thus, it appears that there is much sensitivity to small errors in the calculations when the ellipses are highly eccentric and have pronounced skewing with respect to the initial search directions. These small errors are probably due to errors in the quadratic curve fitting calculations for finding the minimum in the respective search directions.

The validity of this argument is borne out by noting that good convergence was obtained with the rays (1,0) and (10,1), even though the resulting contours were highly eccentric. By computing the amount of skewing in this case, it is found that the axes of the ellipse are skewed by only 2.85 degrees from the coordinate axes; thus, the initial search directions along the coordinate axes are nearly coincident with the axes of the elliptical contours.

Intuitively, it appears that setting the initial search directions coincident with the axes of the elliptical contours should greatly improve convergence. It can in fact be shown that these directions will locate the minimum after only one search along each of them. However, because of the theoretical development in the following section which tackles the problem of selecting search directions from a slightly different foundation, a proof of the above statement will not be given.

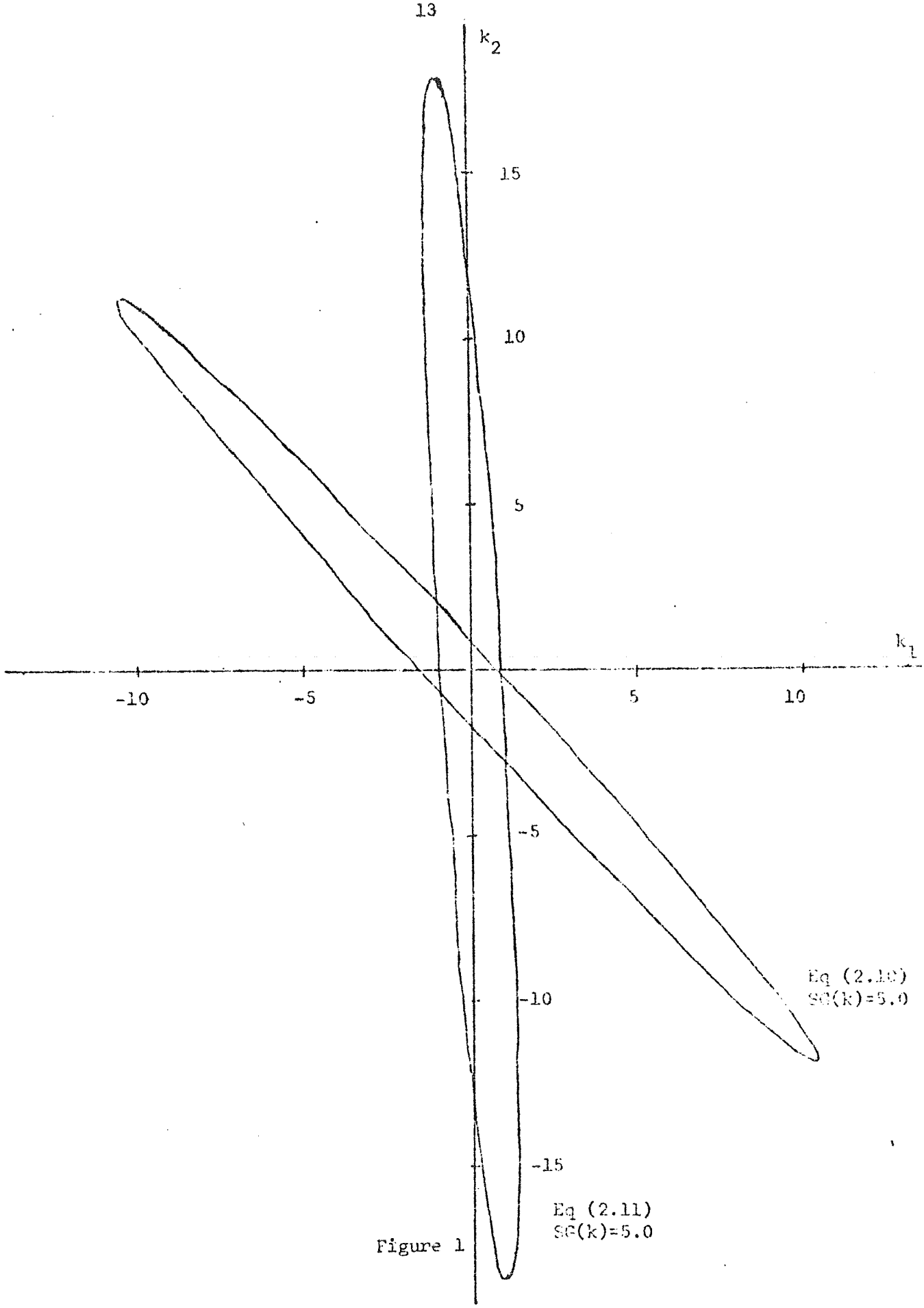


Figure 1

Although search directions coincident with the axes of the contours can be easily computed for a second-order system by the equation

$$\theta = 0.5 \, \mathrm{Tan}^{-1} 2m_{12}/(m_{11}-m_{22}), \tag{2.11}$$

this method of computing search directions cannot be readily extended to higher order systems.

As a consequence of the foregoing discussion, the following section will develop a method for determining conjugate search directions for a system of any order after only one evaluation of the subgoal.

COMPUTING CONJUGATE SEARCH DIRECTIONS FOR MINIMIZING THE SUBGOAL

This section will show that the eigenvectors of matrix M form conjugate directions with respect to M, and that the minimum of the subgoal can be found with only one search along each of these eigenvectors. M is a real symmetric matrix appearing in the subgoal equation (1.9).

Since the material that follows assumes that M is known, it will be stated now that M can be computed while the subgoal (1.6) is being evaluated for the first time. Recalling that

$$M = \sum_{i=1}^{N} \frac{r_i r_i'}{r_i' B r_i}, \tag{3.1}$$

it is only necessary to substitute the value of the state vector $x(t_i)$ for r_i to compute M; thus,

$$M = \sum_{i=1}^{N} \frac{x(t_i)x'(t_i)}{x'(t_i)Bx(t_i)} . \tag{3.2}$$

B is defined for purposes of computing the subgoal; consequently, it is also known. The accuracy to which M can be computed is limited only by the accuracy of the measuring devices, or system noise, or both.

If a quadratic function of the form given by (1.9)

$$SG(k) = \beta k'Mk + 2k'Mp + S \tag{1.9}$$

is to be minimized, then the directions q and r are defined to be conjugate with respect to M if

$$q'Mr=0 \tag{3.3}$$

To prove that the eigenvectors of the real symmetric matrix, M, are conjugate directions, we will use a theorem which states that the eigenvectors of a real symmetric matrix are orthogonal if all the eigenvalues of the matrix are distinct [5]. Assuming the eigenvalues of M are distinct, let

$$Me_i = \lambda_i e_i \tag{3.4}$$

where λ_i is an eigenvalue of the matrix M, and e_i is the corresponding eigenvector. If we let e_j be another eigenvector of M and premultiply both sides of equation (3.4) by $e_j^!$, we have

$$e_j^! Me_j = \lambda_i e_j^! e_j \ . \tag{3.5}$$

Since e_i and e_j are orthogonal,

$$e_j^! Me_j = 0 \text{ for } i \neq j; \tag{3.6}$$

thus, it is proved that the eigenvectors of the real symmetric matrix, M, are conjugate directions.

Powell [4] has proved a theorem which states that a quadratic function in the form of (1.9) can be minimized by only one search in each conjugate direction; consequently if the eigenvectors of M are the search directions, it is possible to find the minimum of the subgoal of an N order system after only N searches. This means that any non-orthogonal set of linearly independent rays can yield convergence times equivalent to those obtained with orthogonal rays. Furthermore, since Powell's method [4] with initial search directions along the coordinate axes requires N(N+1) searches to locate the minimum of a quadratic function when the conjugate directions are not coincident with the coordinate axes, choosing the search directions along the eigenvectors of M has the potential of reducing convergence time by as much as 1/(N+1). As shown by examples to be presented later, it

was found that N or N+1 searches were adequate to very accurately determine the minimum of the subgoal.

At this point it may be argued by some, that eigenvalue solutions are so time consuming and complicated that there is little or no advantage to obtaining them. It is true that this would be the case if we were minimizing an explicit function; but we are minimizing a function that requires from 5N to 10N sampling intervals just to evaluate the function at one point. That is, it takes about 5 to 10 periods to drive the system to a ray in the state space, and there are N rays for an Nth order system. Thus, for a third-order system with a sampling interval of 0.1 second, it requires about 1.5 to 3.0 seconds just to evaluate the subgoal equation (1.6) once. A flow diagram of the operations involved in measuring the subgoal is given in Appendix D.

If the computer is idle during most of the time between sampling periods, it can easily compute a third or fourth order M matrix and its eigenvectors within one or two seconds if it is an efficiently programmed modern high speed digital computer. Thus, for the few seconds taken to compute the eigenvectors of M, the resulting reduction in convergence time is considerable. However, if the computer has tasks other than the learning algorithm, somewhat more time will of course be required to obtain the eigenvector solution.

Since these eigenvectors have conjugate directions, it is also possible to replace Powell's method by a simpler minimization technique in which conjugate search directions need not be computed during the minimization process. Thus, the relaxation method proves to be the most efficient technique if it is modified to search in the directions of the eigenvectors of M rather than the coordinate axes.

This relaxation method was applied to the example of the previous section to demonstrate the improvement in convergence time. Thus, for this demonstration, the subgoal is (2.1) and the system is (2.4). The elements of matrix M are computed by equation (3.2) during the first subgoal evaluation as explained earlier in this section. The normalized eigenvectors of M were computed by the method described in Appendix C and used for the search directions. As in the examples of the previous section, the system was again set precisely upon a given ray one time period prior to measurement of the corresponding partial subgoal.

The results for several pairs of rays are shown in Table 2. As can be seen by comparing Tables 1 and 2, the convergence time for equivalent accuracy when the rays are non-orthogonal, is now one third or one half of what it was when the initial search directions were coincident with the coordinate axes of the parameter space k. Thus, it is seen, for a second order system where the state of the system is located precisely upon a respective ray one time period prior to evaluation of the partial subgoal, that the convergence time is reduced by a factor of 1/(N+1) or 1/N.

This factor is dependent upon the angular separation between the rays. For rays with separations greater than ten degrees, N searches were sufficient to find the minimum to within 0.5 percent of the true minimum; on the other hand, when the rays had separations of about ten degrees or less, N+1 searches were required to find the minimum to the same accuracy.

The next section will apply the method of computing conjugate search directions described in this section to a more realistic, third-order system where the system is driven toward the rays using appropriate

computed feedback gains rather than being set precisely upon the respective rays.

THIRD ORDER EXAMPLE

This section will be concerned with applying the learning algorithm to a simulated third-order discrete system. Rather than setting the system precisely upon the respective rays as in the previous examples of this thesis, the system will now be driven to the rays by the appropriate feedback gain vectors. In addition to applying the method of the previous section for computing conjugate search direcitons to a more realistic third-order system, a test will also be made to determine how small deviations from the actual rays affect convergence of the algorithm.

The system is described by the following equation.

$$x(t_i+T)=\phi x(t_i)+hu(t_i) \tag{4.1}$$

where

$$\phi = \begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.6 & 0.7 & 0.1 \end{vmatrix} \quad \text{and } h = \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix}$$

For this example, the nominal parameters will be equal to the actual values; however, this does not result in any loss of generality.

To compute the feedback gain vectors required to drive an Nth order system to any one of N arbitrary but linearly independent rays, N sets of eigenvalues must be chosen. Each set must consist of N eigenvalues chosen so that one value will dominate the other N-1 values as explained in the introduction. To determine a feedback gain vector such that a respective set of chosen eigenvalues are the eigenvalues of the closed loop system matrix

$$A = \phi + hk' \tag{1.12}$$

the set of N equations

$$\begin{aligned} &\det [A(k)-\lambda_1 I]=0 \\ &\det [A(k)-\lambda_2 I]=0 \\ &\cdots \\ &\det [A(k)-\lambda_N I]=0 \end{aligned} \tag{4.2}$$

must be solved for the vector k.

For the particular system described by equation (4.1), the following three sets of eigenvalues were chosen.

$$\begin{aligned} \underline{\lambda}_1&=(1.0,0.1,-0.1) \\ \underline{\lambda}_2&=(-1.0,\ 0.1,\ -0.1) \\ \underline{\lambda}_3&=(1.5,\ 0.1,\ -0.1) \end{aligned} \tag{4.3}$$

Solving the set of equations (4.2) for each of these sets of eigenvalues the corresponding feedback gain vectors are obtained.

$$k_1=\begin{vmatrix} -0.61 \\ -0.69 \\ 0.9 \end{vmatrix},\quad k_2=\begin{vmatrix} -0.59 \\ -0.69 \\ -1.1 \end{vmatrix},\quad k_3=\begin{vmatrix} -0.615 \\ -0.69 \\ 1.4 \end{vmatrix} \tag{4.4}$$

It should be noted that the gain vector, k, to be adjusted to minimize the subgoal is distinguished from the feedback vectors by not having a subscript.

The resulting <u>dominant</u> eigenvectors corresponding to each of the above feedback gain vectors are

$$e_1=\begin{vmatrix} 1.0 \\ 1.0 \\ 1.0 \end{vmatrix}\qquad e_2=\begin{vmatrix} 1.0 \\ -1.0 \\ 1.0 \end{vmatrix}\qquad e_3=\begin{vmatrix} 1.0 \\ 1.5 \\ 2.25 \end{vmatrix}$$

These are the rays of the system, and it can be seen that they are linearly independent. However, if it had been found that they were not independent, one or more sets of eigenvalues in (4.3) would have

had to be changed, and new vectors computed.

To drive the system to the jth ray whose direction is represented by e_j in (4.5), the jth feedback gain vector of (4.4) is applied to the system such that the control law becomes

$$u(t_i) = k_j x(t_i). \tag{4.6}$$

To determine if the system has converged upon a ray, the angle is measured between the present state vector at time t_i and the previous state vector one sample period earlier, at time t_i-T. The angle can be computed by the equation

$$\cos\ \theta = \frac{x'(t_i)x(t_i-T)}{||x(t_i)||\cdot||x(t_i-T)||} \tag{4.7}$$

Letting $x=x(t_i)$ and $y=x(t_i-T)$, and applying the trigonometric identity $\sin\theta=(1-\cos^2\theta)^{1/2}$, equation (4.7) becomes

$$\sin\ \theta = [1-(x'y)^2/x'xy'y]^{1/2} \tag{4.8}$$

However, since the angle for determining convergence will be small, we can let $\theta=\sin\ \theta$; thus, the criterion for determining that the system has essentially converged on a ray will be to acertain that

$$[1-(x'y)^2/x'xy'y]^{1/2}<\theta \tag{4.9}$$

for some suitable small number θ.

The subgoal used for this example is the same one used in the examples of the previous sections. That is,

$$SG(k)=\sum_{i=1}^{N} SG_i(k)=\sum_{i=1}^{N} \frac{x'(t_i+T)x(t_i+T)+u^2(t_i)}{x'(t_i)x(t_i)} \tag{2.1}$$

This is the subgoal equation (1.6) with R=B=I and α=1.0. A flow diagram is given in Appendix D to show diagrammatically the operations

involved in computing the subgoal. Also shown in this diagram are the computations for the matrix M which are made only once when the subgoal is evaluated for the first time. After the first evaluation of the subgoal is complete, the normalized eigenvectors of M are computed and used for the search directions. These search directions are used throughout the minimization process. That is, repetitive searches are conducted in these direcitons until the minimum is found. The minimum in a respective search direction is determined from (2.2) and (2.3) by computing the value of the subgoal at three points, and computing the minimum based on a quadratic curve fitted through the points.

For various values of θ in equation (4.9), and an initial gain vector of $k_0'=(-1, -1, -1)$, the results are summarized in Table 3. The number of searches and the number of sampling periods listed are those that were required to compute the corresponding gain vectors given in the table.

The desired gain vector, as obtained from (1.11), is found to be $k_*'=(-0.30, -0.35, -0.05)$. The last column of Table 3 gives a measure of the deviation of the learned gain vector from the desired value. Since the gain component, k_3, though smaller than the other components of the gain vector by almost an order of magnitude, usually deviated from its true value by a greater percentage than any of the other components, the components of the gain vector were normalized before the mean square error was computed. Thus the normalized error given in Table 3 is computed by the equation

$$\overline{e_n^2} = \frac{1}{N} \sum_{i=1}^{N} ((k_i^* - k_i)/k_i^*)^2. \qquad (4.10)$$

where k_j^* is desired value of the ith component of the gain, and k_i is the learned value.

Examining Table 3, it can be seen that convergence on the optimal gain is most rapid and accurate if the ray convergence criterion, θ, of (4.9) is 0.3 degree or less. That is, only one search is required in each of the conjugate search directions. If the sampling periods are 0.1 seconds, it should be noted that the convergence time, when θ=0.3 degree, is only slightly longer than the convergence time of 13 seconds that Scott [1] obtained with his second order example.

When θ is in the range of 0.4 to 0.7 degree, the accuracy is still very good, but convergence time has increased by about 50 percent. And when θ=1.0 degree convergence time has doubled although the accuracy is still good; However, increasing θ to 2.0 degrees yielded rather poor results. Thus, the upper limit of θ, that is consistent with accurate results in a reasonable amount of time, lies somewhere between 1.0 degree and 2.0 degrees. Based upon this knowledge, it is reasonable to estimate that reliable convergence can be obtained if the state measurements are corrupted with 1 to 2 percent noise.

The conjugate search directions that were computed, based upon measurement of the rays obtained during the first evaluation of the subgoal, are not shown in Table 3; however, they vary only slightly from one case to another. Consequently, since the directions obtained when θ=0.1 degree are most accurate, these will be given.

When θ=0.1 degree, the rays converged upon, to six significant digits, were

$$r_1 = \begin{vmatrix} -0.999801 \\ 1.000000 \\ -0.999999 \end{vmatrix} \qquad r_2 = \begin{vmatrix} 0.999717 \\ 0.999915 \\ 0.999913 \end{vmatrix} \qquad r_3 = \begin{vmatrix} -5.45707 \\ -8.18592 \\ -12.2789 \end{vmatrix}$$

These rays are the state vectors as measured at time t_i when the condition of inequality (4.9) is satisfied.

The M matrix computed from these rays by equation (3.2) was

$$M = \begin{vmatrix} 0.786783 & 0.180446 & 0.937291 \\ 0.180446 & 0.937435 & 0.406019 \\ 0.937291 & 0.406019 & 1.275780 \end{vmatrix} \tag{4.12}$$

The computed eigenvalues of this matrix were 0.797618, 0.048613, and 2.15377; consequently, the normalized eigenvectors of M were computed to be

$$e_1 = \begin{vmatrix} 0.310600 \\ -0.932728 \\ 0.183158 \end{vmatrix} \quad e_2 = \begin{vmatrix} -0.766291 \\ -0.131692 \\ 0.628852 \end{vmatrix} \quad e_3 = \begin{vmatrix} 0.562427 \\ 0.335675 \\ 0.755644 \end{vmatrix} \tag{4.13}$$

Thus, these were the search directions that were used for minimizing the subgoal when θ was specified as 0.1 degree. Because of the slight difference in the rays obtained with other values of θ given in Table 3, the search directions also differed slightly from (4.13) when θ differed from 0.1 degree.

From this example it is observed that, if the method of computing conjugate search directions as outlined in the previous section is incorporated in the learning algorithm, convergence time of a third-order system can be reduced by a factor of as much as 1/4 with no impairment in accuracy.

SUMMARY AND CONCLUSION

As a result of the investigation of the effect of ray orientation upon convergence of a second-order system, it was found that if arbitrary orthogonal directions such as the coordinate axes are chosen as search directions for minimizing the subgoal, orthorgonal rays yield the most rapid convergence time. If a minimization technique which locates the minimum of a quadratic function within a fixed number of iterations is used to minimize the subgoal, non-orthogonal rays generally required longer but uniform convergence times, except in certain cases where small angular separations between them resulted in contours with pronounced eccentricity and skewing.

The resolution of the difficulties encountered with small angular separations lead to development of a method of computing conjugate search directions such that the minimum of the subgoal can be found after only one search in each direction. By using these search directions, convergence times for most non-orthogonal rays equalled those previously obtained with orthogonal rays. In addition, for those cases in which convergence was very poor because of pronounced eccentricity and skewing of the elliptical contours, the new approach yield excellent results.

The learning algorithm with this improvement, when applied to a third-order system, was found to converge to desired gain vector after only one search in each direction if the system was driven sufficiently close to the respective rays prior to making measurements of the subgoal.

Based upon results of computer runs it was estimated that convergence is possible if the state measurements are corrupted with less than 1 to 2 percent noise; however, the effect of noise upon convergence

should be more thoroughly investigated. An attempt should also be made to apply the algorithm to an actual system.

REFERENCES

1. Scott, R. E., "An Improved Algorithm for Learning Systems", Ph. D. Thesis, University of Connecticut, Storrs, Connecticut, June 1970.

2. Waltz, M. D., Fu, K. S., "A Learning Control System", Reprints of the JACC, 1964.

3. Waltz, M. D., Fu, K. S., "A Heuristic Approach to Reinforcement Learning Control Systems", IEEE Transactions on Automatic Control, Vol. AC-10, No. 4, October, 1965, pp. 390-398.

4. Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives", Computer Journal, Vol. 7, 1964, pp 155-162.

5. Finkbeiner, D. T., Introduction to Matrices and Linear Transformations, W. H. Freeman and Company, San Francisco, 1960, pp. 201-209.

6. Randolph, J. F., Calculus and Analytic Geometry, Wadsworth Publishing Company, Belmont, California, 1962, pp 278-281.

7. DeRusso, P. M., Roy, R. J., Close, C. M., State Variables for Engineers, John Wiley & Sons, New York, 1967, pp 233-234.

8. McCalla, T. R., Introduction to Numerical Methods and FORTRAN Programming, John Wiley & Sons, New York, 1967.

APPENDIX A

Summary of Powell's Method

Powell [4] has proposed a method of minimizing a function of several variables. Although a variation of the method where one parameter is varied at a time, it computes conjugate search directions which enable it to converge upon the minimum of a quadratic function within a fixed number of iterations. The quadratic has the form

$$f(x) = x'Ax+b'x+c, \tag{A1}$$

and the directions p and q are defined as being conjugate with respect to A if

$$p'Aq=0 \tag{A2}$$

Each iteration of the algorithm starts with a search in n linearly independent directions $\xi_1, \xi_2, \ldots, \xi_n$, starting from the best approximation of the minimum x_0. Conjugate directions are generated by computing a new search direction ξ at the finish of each iteration such that the n linearly independent search directions of the next iteration are $\xi_2, \ldots, \xi_n, \xi$. Because of the way in which ξ is defined, the last k of the n directions chosen for the (k+1)th iteration are mutually conjugate if the function being minimized is quadratic. Thus, after n iterations, all n of the directions and the minimum of the quadratic is found.

Given n linearly independent search directions $(\xi_1, \xi_2, \ldots, \xi_n)$ and an approximation of the minimum p_0, an iteration consists of the following steps:

1) For r = 1,2,...,n calculate λ, so that $f(p_{r-1}+\lambda_r\xi_r)$ is a minimum and define $p_r=p_{r-1}+\lambda_r\xi_r$.

2) For $r = 1, 2, \ldots, n-1$ replace ξ_r by ξ_{r+1}.

3) Replace ξ_n by $(p_n - p_0)$.

4) Choose λ so that $f(p_n + \lambda (p_n - p_0))$ is a minimum and replace p_0 by $p_n + \lambda(p_n - p_0)$.

APPENDIX B

Conditions Under Which M is a Diagonal Matrix

Given n orthogonal rays (vectors) $r_1, r_2, \ldots, r_n$ and

$$M(r_1, r_2, \ldots, r_n) = \sum_{i-1}^{n} \frac{r_i r_i'}{r_i' B r_i} \tag{B1}$$

then M is a diagonal matrix if B = I, i.e., B is an identity matrix.

Proof:

M can be factored into

$$u_i = \frac{r_i}{(r_i' B r_i)^{1/2}} \quad \text{and} \quad u_i' = \frac{r_i'}{(r_i' B r_i)^{1/2}} \tag{B2}$$

Since the r_i vectors are orthogonal, so are the u_i vector. Now let P be an n x n matrix with the columns composed of the vectors $u_1, u_2, \ldots, u_n$, i.e.,

$$P = [u_1, u_2, \ldots, u_n]. \tag{B3}$$

Then P' is an n x n matrix with rows composed of the row vectors $u_1', u_2', \ldots u_n'$, i.e.,

$$P' = \begin{vmatrix} u_1' \\ u_2' \\ \cdot \\ \cdot \\ \cdot \\ u_n' \end{vmatrix} \tag{B4}$$

and $M = PP' = \sum_{i=1}^{n} u_i u_i'$.

If PP' = I, then M is a diagonal matrix; however, this requires that P be an orthogonal matrix. But for P to be orthogonal, the u_i vectors

must not only be orthogonal but orthonormal, i.e., $u_i' u_j = \delta_{ij}$. The requirement that they also be unit vectors can be easily satisfied if $B = I$ in (B2). Thus, if the rays $r_1, r_2, \ldots, r_n$ are chosen as orthogonal and $B = I$, then M is an identity matrix, and therefore a diagonal matrix.

APPENDIX C

Method of Computing Eigenvectors and Eigenvalues

Although there are some specialized techniques for finding the eigenvalues and eigenvectors of a real symmetric matrix; the method to be outlined here is composed of several basic numerical analysis methods and has the advantage of being rather straight forward.

The first objective is to compute the coefficients of the characteristic equation of the matrix M. For a characteristic equation of the form

$$\lambda^n + a_1 \lambda^{n-1} + a_2 \lambda^{n-2} + \ldots + a_{n-1} \lambda + a_n = 0, \quad (C1)$$

DeRusso, et al, [7] has presented the following recursive formula whereby the coefficients of the equation can be computed.

$$a_1 = -T_1$$

$$a_2 = -\frac{1}{2}(a_1 T_1 + T_2)$$

$$a_3 = -\frac{1}{3}(a_2 T_1 + a_1 T_2 + T_3)$$

$$\ldots \quad \ldots$$

$$a_n = -1/n\ (a_{n-1} T_1 + a_{n-2} T_2 + \ldots + a_1 T_{n-1} + T_n)$$

T_k is defined as the trace of M^k. The trace is the sum of the diagonal elements of the matrix, and M^k is the matrix M multiplied by itself k times.

The next step is to find the eigenvalues of (C1). Since M is real symmetric, all the roots of the characteristic equation (C1) will be positive real [5]; thus, the Birge Vieta method [8] of finding real roots of a polynomial can be employed. From an initial estimate, this method converges upon a real root by the Newton Raphson [8] iterative

method. The polynomial and its derivative are evaluated by synthetic division. As each root is found the order of the polynomial is reduced.

After the eigenvalues are determined, the eigenvectors can be solved by the equation

$$[M - \lambda I]e=0 \qquad C3)$$

Since the eigenvector e is not unique, the element e_n will be set equal to unity; thus, equation (C3) becomes

$$\begin{aligned}
&(m_{11}-\lambda)e_1+m_{12}e_2+\ldots+m_{1,n-1}e_{n-1}+m_{1n}=0\\
&m_{21}e_1+(m_{22}-\lambda)e_2+\ldots+m_{2,n-1}e_{n-1}+m_{2n}=0\\
&\quad\ldots\qquad\quad\ldots\qquad\quad\ldots\qquad\quad\ldots\\
&m_{n-1,1}e_1+\ldots+(m_{n-1,n-1}-\lambda)e_{n-1}+m_{n-1,n}=0\\
&m_{n1}e_1+m_{n2}e_2+\ldots+m_{n,n-1}e_{n-1}+(m_{nn}-\lambda)=0.
\end{aligned} \qquad (C4)$$

Thus we now have n equations in n-1 unknowns.

For the eigenvalue, λ, the corresponding eigenvectors can be obtained by the following steps:

1) Subtract λ from the diagonal elements of matrix M.
2) Reverse the sign of the elements in the nth column.
3) Choose n-1 rows of the matrix.
4) Perform a Gauss-Jordan [8] reduction on the (n-1)xn resulting matrix.

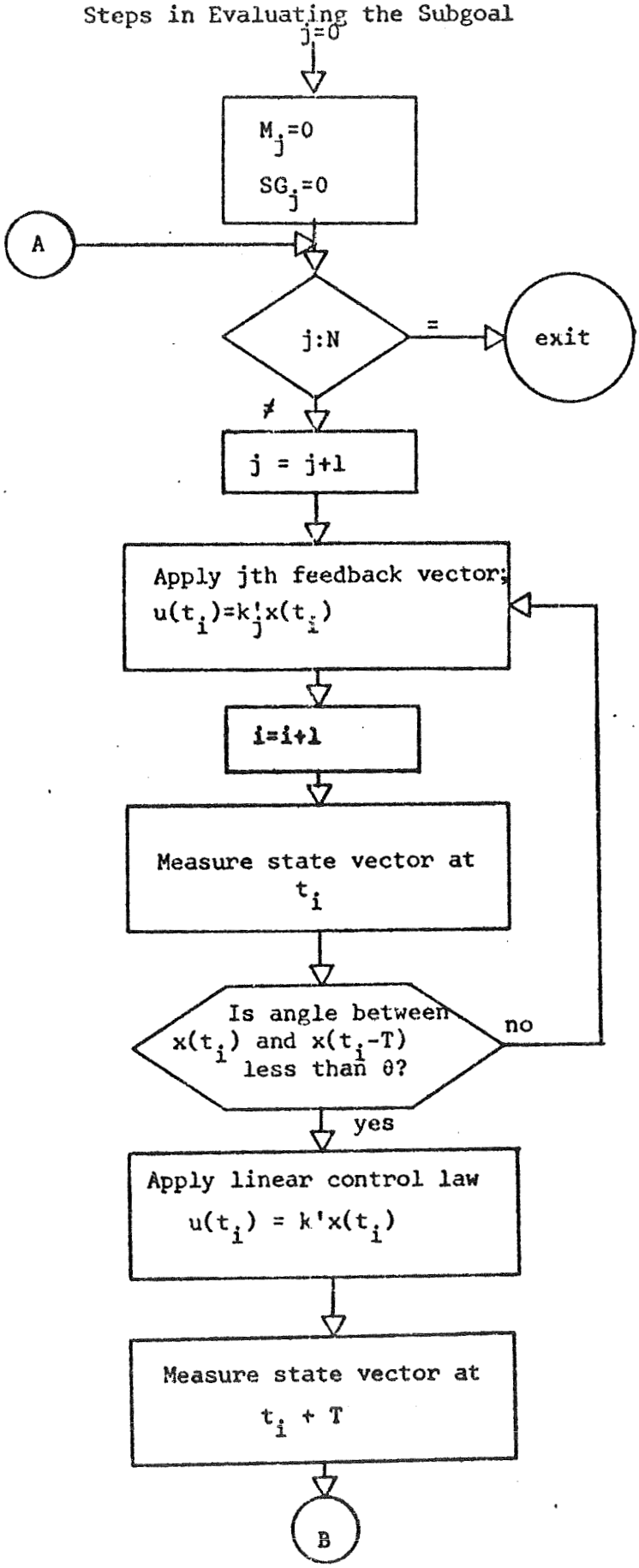
The solution for the first n-1 elements of the eigenvector will be the n-1 elements of the nth column of the reduced matrix. Hence, if the elements of the reduced matrix are denoted as m^*_{ij}, the eigenvector solution will be

$$e = \begin{vmatrix} m^*_{1n} \\ m^*_{2n} \\ \cdot \\ \cdot \\ \cdot \\ m^*_{n-1,n} \\ 1.0 \end{vmatrix} . \qquad (C5)$$

In rare instances, the element e_n will be zero. In this case, setting $e_n = 1.0$ will result in no solution when the Gauss-Jordan reduction is performed. If this should happen, a different element, r_i, should be set equal to unity. Then perform the steps of the above procedure with the exception that, at step 2, the ith and nth columns of the matrix will be interchanged before the signs of the nth column are reversed. Now the solution for the eigenvector will contain a one in the ith element. However, if no solution can be obtained regardless of which element is set equal to unity, the characteristic equation has a root which has a multiplicity greater than one.

APPENDIX D

Steps in Evaluating the Subgoal



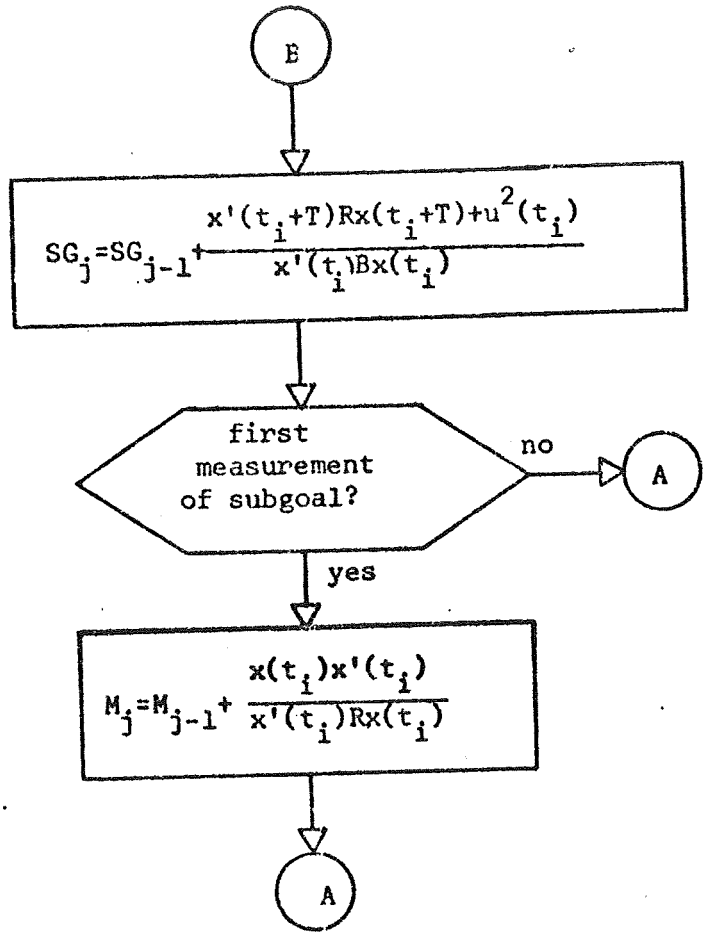

E
$SG_j = SG_{j-1} + \frac{x'(t_i+T)Rx(t_i+T)+u^2(t_i)}{x'(t_i)Bx(t_i)}$
first measurement of subgoal?
no
A
yes
$M_j = M_{j-1} + \frac{x(t_i)x'(t_i)}{x'(t_i)Rx(t_i)}$
A

TABLE 1

Powell's method [4] used to minimize subgoal with initial directions along the coordinate axes.

Rays			# subgoal		Final Gains	
r_1	r_2	Angle	measurements	# searches	k_1	k_2
1,0	0,1	90°	6	2	-0.03334	-0.3333
1,2	-2,1	90°	6	2	-0.03333	-0.3333
1,0	1,2	63.4°	18	6	-0.03333	-0.3333
1,2	2,1	36.9°	18	6	-0.03325	-0.3334
1,1	1,0.6	13.9°	18	6	-0.03209	-0.3349
1,1	1,0.7	10.0°	18	6	-0.04911	-0.3143
1,1	1,0.8	6.3°	18	6	0.12810	-0.5147
1,0	10,1	5.7°	18	6	-0.03333	-0.3333
1,0	40,1	1.4°	18	6	-0.03333	-0.3331

TABLE 2

Subgoal minimized by repetitive searches along eigenvectors of M matrix

Rays r_1	Rays r_2	angle	# subgoal measurements	# searches	Final Gains k_1	Final Gains k_2	Search Directions s_1	Search Directions s_2
1,0	1,2	63.4°	6	2	-0.03336	-0.3333	-0.5257,0.8507	0.8507,0.5257
1,2	2,1	36.9°	6	2	-0.03331	-0.3334	-0.7071,0.7071	0.7071,0.7071
1,1	1,0.6	13.9°	6	2	-0.03317	-0.3335	-0.6154,0.7882	0.7882,0.6154
1,1	1,0.7	10.0°	6	2	-0.03226	-0.3346	-0.6427,0.7661	0.7661,0.6427
			9	3	-0.03334	-0.3333		
1,1	1,0.8	6.3°	6	2	-0.03207	-0.3348	-0.6669,0.7451	0.7451,06669
			9	3	-0.03334	-0.3335		
1,0	10,1	5.7°	6	2	-0.03298	-0.3406	-0.0498,0.9988	0,9988,0.0498
			9	3	-0.03334	-0.3333		
1,1	1,0.95	1.5°	6	2	-0.01636	-0.3508	-0.6980,0.7161	0.7161,0.6980
			9	3	-0.03345	-0.3332		

TABLE 3

Third-Order Example

θ	Num. of Searches	Sampling Periods	Gain Vectors k_1	k_2	k_3	Error $\overline{e_n^2} \times 10^4$
0.1°	3	154	-0.2995	-0.3500	-0.05020	0.065
0.2°	3	145	-0.2995	-0.3500	-0.05020	0.063
0.3°	3	136	-0.2999	-0.3499	-0.05032	0.144
0.4°	5	227	-0.3023	-0.3512	-0.04961	0.452
0.5°	5	220	-0.3030	-0.3517	-0.05077	1.21
0.7°	5	215	-0.3031	-0.3515	-0.05087	1.41
1.0°	8	336	-0.3004	-0.3507	-0.04926	0.75
2.0°	11	439	-0.2974	-0.3377	-0.05975	131.0