UNIVERSITY OF CONNECTICUT

SCHOOL OF ENGINEERING

STORRS, CONNECTICUT

AN IMPROVED ALGORITHM FOR LEARNING SYSTEMS

R. N. Scott

Department of Electrical Engineering

August 1969

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

An Improved Algorithm for Learning Systems

R. N. Scott

Introduction

An algorithm is developed for implementing a learning controller based on the concept of a subgoal. The method differs from previous work in that the state space need not be quantized, resulting in vastly reduced memory requirements and shorter learning times. The method is applicable to linear stationary systems with unconstrained control variable wherein the system parameters are specified with uncertainty.

One of the very few known simulations of a fully automated learning control system has been reported by Waltz and Fu. [1,2]. Since these reports are the basis for the research discussed in this thesis, it is appropriate to discuss the general aspects of the Waltz and Fu learning algorithm.

The method proposed by Waltz and Fu utilizes the concepts of sample set construction and linear reinforcement. Sample set construction involves the partitioning of the state space into control situations which describe the state dependence of the control law. Linear reinforcement is the method by which the control law is adjusted to minimize a given functional, gauging overall system performance. This particular learning scheme is applicable to nonlinear, nonstationary systems. The only a-priori information required by the design is the order of the plant to be controlled. The parameters and the exact form of the plant may be unknown. As usual, one pays a penalty for such wide-ranging freedom. In this case, the penalty is realized in the initial phase of learning when the system must test each available control choice to determine the most appropriate choice for each possible situation. During this initial training period, immediate improvement is never realized, in fact, the system may be forced into an unstable situation. However, if sufficient information relating to the plant dynamics is available, then the excitation of unstable modes of the system can be avoided. In the case of a linear plant whose parameters are nominally known and a quadratic performance functional, L. E. Jones [3] demonstrated that it is more beneficial to implement a learning controller on the basis of the nominal parameters than to accept a control law determined by these nominal values alone. The details in the development of these two related learning systems follow.

The method discussed by Waltz and Fu is applicable to the wide range of systems whose dynamics can be described by a general differential relationship

$$\dot{x} = f(x,u,t), \tag{1.1}$$

where $x' = (x_1, x_2, \ldots, x_n)$ is the state vector defined on the state space Ω_x, and u the control signal chosen from the control space Ω_u. The principal objective is to determine a control policy (not necessarily linear)

$$u(t) = g(x(t)) \tag{1.2}$$

which minimizes a given performance index.

The actual learning process is accomplished with the aid of a digital

computer. The inclusion of the computer imposes a number of practical restrictions on the design of the learning controller. First, the control law cannot be defined as a explicit functional or analog relationship but rather it must be described by a relationship between the individual elements of the state space Ω_x and the control space Ω_u. Thus for each point in Ω_x there must correspond a point in Ω_u such that the total correspondence defining the law minimizes the given performance index. If Ω_x and Ω_u are both spaces with a countably infinite number of points or elements, the computer used in the learning process must have an infinite storage capability. Thus it becomes necessary to quantize the state space Ω_x into a finite number of subsets and restrict the choice of a control signal from a finite control space Ω_u. In this case, the functional correspondence defining the control law would be established between the subsets of Ω_x and each of the finite number of elements of Ω_u. Thus a single control choice would apply to the aggregate of points defining each subset or sample set in Ω_x such that the total relationship is optimal with respect to the given performance index.

The second consideration resulting from the use of the digital computer is due to the finite time required by the computer to execute an algebraic operation. Each choice of a control signal from the space Ω_u must be evaluated according to its effect on the given performance index. Since each evaluation requires a finite number of mathematical operations, the information related to the effectiveness of a particular control choice will not be available at the same instant the measurement of the performance index is made. Thus it is necessary to discretize time to allow time for evaluating each decision. In the discrete case, the control is held constant during the interval of time required to evaluate the last decision. Hence, the plant is described by a vector difference equation

$$x(i+1) = F(x(i),u(i),i). \tag{1.3}$$

Thus the actual implementation of a learning system utilizing a digital computer requires the following operation:

1) Discretize the independent variable to allow time for making and reinforcing decisions.
2) Quantize the control input into a finite set of allowable control actions, and
3) Partition the state space into subsets called sample sets to reduce the computer storage requirement.
4) Choose a reinforcement algorithm to supervise the learning process.

The above reinforcement algorithm must be related to the system performance index in the sense that the control law learned by the system will eventually approach the optimal control policy. The index of per-

formance of the system is of the form

$$IP = \sum_{r=1}^{n} r(|x_1(rT)|)^a, \tag{1.4}$$

where a is an arbitrary constant selected by the designer. Again, the primary objective of the reinforcement-type learning control system is to recognize what control signal should be applied when the plant output is observed within a certain region (sample set) of the state space. The decision to apply positive or negative reinforcement as well as the degree of reinforcement applied to a particular control situation is based on previous observations of the state vector and the resulting behavior of the system when the control choice under consideration was applied to the system. The objective of the system is to learn the control law which minimizes the given index of performance, called the primary goal, evaluating system behavior over an extended period of time. Thus, if the reinforcement algoritym is based on this primary goal, the decision to reinforce a particular control situation at a given instant of time must be withheld until a performance measure is available at some later time. It is, therefore, necessary to introduce a secondary index of performance, referred to as the subgoal, which, when minimized at each decision, results in near-optimum performance.

The difficulty encountered in using the main goal rather than a subgoal has been compared to playing a game of chess and evaluating each move on the basis of win or lose information. The chess player must consider some intermediate goal in making the separate moves. Thus an effective subgoal must fulfill two conditions:

1) It must evaluate each decision separately,
2) It must be related to the main goal so that fulfilling the subgoal is essentially equivalent to satisfying the main goal.

In the paper by Waltz and Fu, the subgoal was to choose u(i) so as to maximize

$$\frac{IPS(i) - IPS(i+1) - \lambda u^2(i)}{\max[IPS(i),\ IPS(i+1) + \lambda(u_{max})^2]} \tag{1.5}$$

where

(i) $\lambda > 0$,
(ii) $IPS(i) = x'(i)Gx(i)$ and G is a positive-definite diagonal matrix, and
(iii) $u_{max} = \max_i (u_i : u_i \epsilon \Omega_u)$.

The elements of G are determined by a secondary learning loop whose objective is to find the diagnal matrix G such that the corresponding control law yields the lowest numerical value of the primary or main goal. This secondary learning loop consists of a multidimensional search scheme.

The selection of an appropriate subgoal in the above argument is rather

heuristic. However, if the plant to be controlled is linear and time-invariant, the performance index (main goal) to be minimized is quadratic, and information regarding the nominal values of the plant parameters is available, then a less ambiguous selection of a subgoal can be made as follows.

Given a linear, time-invariant system defined by the vector difference equation

$$x(i+1) = \phi\ (i) + hu(i), \tag{1.6}$$

where

(i) x(i) is an n dimensional vector representing the sampled value of the state of a continuous process at time t = iT,
(ii) ϕ is an n x n matrix whose elements are nominally known,
(iii) h is an n x 1 column vector of nominal parameters, and
(iv) u(i) is the scalar-valued control variable.

Let the n x n positive semi-definite matrix Q and the positive scalar α define the quadratic performance index

$$I(u,x_0) = \sum_{i=0}^{n-1} (x'(i+1)Qx(i+1) + \alpha u^2(i)). \tag{1.7}$$

If no constraints are imposed on the control variable u(i) and with N fixed, x(N) free, the optimal control law is known to be a linear function of the system state, i.e.,

$$u^*(i) = k'(i+1)x(i) \tag{1.8}$$

and the minimum value of the index of performance is

$$I^*(x_0) = I(u^*,x_0) = x_0'P(0)x_0 \tag{1.9}$$

where k(i) and P(i) are determined by iterating the following set ot equations backwards in time starting with P(N) = 0.

$$R(i) = P(i) + Q \tag{1.10}$$

$$k'(i+1) = -\frac{h'R(i)\ \phi}{h'R(i)h + \alpha}$$

$$\Phi\ (i) = \phi + hk'(i)$$

$$P(i-1) = \Phi'(i)P(i)\ \Phi(i) + \alpha k(i)k'(i).$$

Consider a subgoal of the form

$$SG(i) = x'(i+1)G(i)x(i+1) + \lambda u^2(i). \tag{1.11}$$

Substituting the state equations(1.6) in (1.11) and minimizing with respect to u(i) yields a control law of the form

$$u(i) = - \frac{h'G(i)\,\phi}{h'G(i)h + \lambda}\; x\,(i). \tag{1.12}$$

In the case where the exact values of the elements of ϕ and h are known, the best choice of G(i) and λ would be

$$G(i) = R(i) = P(i) + Q \tag{1.13}$$

$$\lambda = \alpha .$$

Then, minimizing SG(i) at each point t = iT with respect to u(i) is equivalent to minimizing $I(u,x_0)$ over the entire interval (0,NT) in the sense that the the two approaches yield equivalent control policies and therefore identical minimum values for the performance index $I(u^*,x_0)$.

In order that the subgoal be optimal with respect to the main goal, it is necessary, as shown in (1.13), that G be a time variable matrix. If this requirement is to be maintained, then the required computer memory must be expanded to permit the storage of G at each sample point t = iT. Thus, in the interest of reducing the memory requirement, G will be assumed time-invariant and equal to R(0). Then the subgoal will be suboptimal, and optimal only in the case where $N = \infty$.

In the formulation of the learning control problem ignorance of parameter values is assumed, otherwise there is no need to learn. Thus, the following question is raised: what is the best learning subgoal in the case of uncertainty in the parameters? The preceeding argument is no longer applicable since the exact system parameter values are not available to determine the optimal control law and subgoal. However, nominal values of these parameters are given and can be used to determine a subgoal which can be used to supervise the learning for the actual system. Consider the special situation when the actual and nominal values are equivalent. Then the subgoal, when minimized with respect to the control variable u(i) at each decision point, will yield the optimal control law for the actual (nominal) plant. In fact, it has been demonstrated by Jones that the control policy determined by this subgoal is superior in the following sense. Let $u_F(i)$ be the control law derived by solving the optimal control problem for the nominal plant and let $I_F(x_0)$ be the resulting value of the performance index when this control is applied to the actual system. Let $u_L(i)$ be the control law determined by the learning algorithm using the subgoal based on the nominal plant parameters. $I_L(x_0)$ is the corresponding value of the performance functional. Then it has been shown that

$$I_0(x_0) \leq I_L(x_0) \leq I_F(x_0) \tag{1.14}$$

where $I_0(x_0)$ is the minimum cost for the actual system. Thus, it is more

beneficial to learn the control law using a subgoal based on the nominal parameter values than to accept the control policy determined by the optimal solution for the nominal plant.

In the foregoing discussion, the control variable was assumed to be unconstrained. Therefore, recalling that it is necessary to restrict the class of allowable control sets to include only countable sets in order to reduce the computer memory requirement, the above results are not directly ameanable to the reinforcement learning algorithm as proposed by Waltz and Fu. However, the same subgoal can be utilized to implement the reinforcement-type learning system. Consider the simple example where either positive, negative, or zero forcing is available, i.e.,

$$u(i) \epsilon \Omega_u = (-1, 0, +1). \tag{1.15}$$

Then, using the subgoal determined by the optimal solution for the nominal plant assuming no control constraints, the learning system will converge to the nonlinear control law defined by the two switching lines

$$k_L' x = \pm 1/2 \tag{1.16}$$

where

$$k_L' = - \frac{h' G \phi}{h' G h + \alpha} \tag{1.17}$$

and $x(i+1) = \phi x(i) + h u(i)$ is the actual system equation. In the general case, the switching lines are determined by

$$k_L' x = -1/2 \ (u_j + u_{j+1}) \tag{1.18}$$

where u_j and u_{j+1} are elements of the ordered control set

$$\Omega_u = \{ u_1, \ldots, u_j, u_{j+1}, \ldots, u_p \}. \tag{1.19}$$

In the preceeding discussion, the entire development was related to the determination of a subgoal for the reinforcement-type learning control system. In this type of learning situation, a large memory block is required to store all the learning parameters. For example, given a second order system with the state space subdivided into 5 x 5 rectangular sets covering $|x_1| \leq 50$ and $|x_2| \leq 50$, at least 2000 parameters are generated in the process of learning a relay control law, i.e., $u = \pm 1$. Higher order systems, finer quantizations of the state space, and wider ranges on the choice of the control variable all increase the memory requirement. Furthermore, the increase in the required storage capability is accompanied by an increase in the time required to complete the learning process.

In the succeeding discussion, a different algorithm is developed for a linear system with unconstrained control. Less severe memory requirements and shorter learning times are realized by this method.

Development of Learning Algorithm

Assume for simplicity in the subsequent discussion that the values of the plant parameters are identical to the actual parameter values. Hence, the problem becomes one of forcing the system to converge to a given limit point defined by the optimal solution. Although this problem has no practical relevance, it does serve to illustrate the procedure for obtaining a sub-optimal solution when the nominal and actual plant parameters are distinctly different. Suppose a convergent procedure has been found which satisfies the conditions of the problem defined above. Then this same procedure will yield a solution to the problem when there is some degree of ignorance of the parameters.

The object of this paper is to develop an algorithm which will learn an optimal control for a linear stationary system described by a vector difference equation

$$x(i+1) = \phi x(i) + hu(i) \tag{2.1}$$

where x(i) is an n-dimensional vector and u(i) is a scalar. The performance of this system is to be measured by the quadratic cost functional

$$\sum_{i=0}^{\infty} x'(i+1)Qx(i+1) + \alpha u^2(i) \tag{2.2}$$

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 linear

$$u(i) = k_*'x(i), \tag{2.3}$$

where k_*' is defined by

$$k_*' = -\frac{h'R\phi}{h'Rh + \alpha}. \tag{2.4}$$

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

Following the procedure proposed by Jones [3], an appropriate subgoal is proposed. Consider a subgoal of the form

$$SG(i) = x'(i+1)Gx(i+1) + \lambda u^2(i). \tag{2.5}$$

The problem is then to determine G and λ such that the subsequent statement applies. If the subgoal is minimized with respect to the constant vector k defining the linear control law

$$u(i) = k'x(i), \tag{2.6}$$

then, starting from some arbitrary point k_0, the system will converge to the optimal control k_* defined by (2.3) and (2.4).

The dependence of the subgoal on the vector k becomes evident if the state equations (2.1) and the linear control law (2.3) are both substituted in the expression for the subgoal. This substitution yields the following expression for the subgoal in terms of the state at time iT, x(i), and the gain vector k,

$$SG(i) = x'(i)[\phi'G\phi + 2kh'R\phi + \beta kk']x(i), \tag{2.7}$$

where $\beta = h'Gh + \lambda$.

Although (2.5) and (2.7) are equivalent expressions for the subgoal, it should be stressed that in the actual minimization routine the subgoal will be evaluated by means of (2.5). However, for analytical purposes, the latter expression for the subgoal will be used.

In order to illustrate the procedure for minimizing the subgoal with respect to the vector gain k, consider a simple discrete system defined by the first-order vector difference equation

$$x(i+1) = x(i) + u(i). \tag{2.8}$$

Suppose that a subgoal has been determined which will force the system to converge to a given control law, k = -1/2. Let that subgoal be defined by

$$SG(i) = x^2(i+1) + u^2(i). \tag{2.9}$$

The equivalent expression for the subgoal is found to be

$$SG(i) = x^2(i)(1 + k)^2 + k^2x^2(i). \tag{2.10}$$

In this simple one-dimensional problem where the subgoal is easily observed to be a strictly convex function of the argument k, an elementary one-dimensional search routine is the most appropriate procedure for minimizing the subgoal with respect to k. An outline of this procedure follows.

a) Starting at some arbitrary initial state x(0), allow the system to run for one sample period under the influence of the control $u(0) = k_0x(0)$. Measure the state at i = 1 and determine the value of the subgoal corresponding to k_0 by substituting in (2.9).

b) Repeat step (a) for $k_1 = k_0 + \delta k$ to determine whether k_0 lies to the right or left of the minimum. If the value of the subgoal corresponding to $k = k_1$ is less than the value corresponding to $k = k_0$, proceed in the direction defined by k, otherwise, reverse the direction of search.

c) Continue this trial and error procedure until a minimum is located.

However, a false conclusion might be obtained in step (b) unless the state observed one sample period prior to the measurement of the subgoal is identical to the initial state x(0) in step (a). Thus, the system state must be held constant throughout the minimization. The difficulty that may arise if this restriction is not observed is easily illustrated in a graph of SG(i) vs. k with x(i) as a parameter (Fig. 2.1). Suppose x(0) = 1, k_0 = 1. The value of the subgoal corresponding to these values is five. In step (b), let $\delta k = -1/2$ and suppose that the state one sample period prior to the measurement of the subgoal is equal to two. Then the value of the subgoal corresponding to these values is ten, indicating that the minimum lies to the right of k_0 when, in fact, the minimum lies to the left of k_0. Thus, one additional condition should be added to both steps b and c:

> Measure the subgoal according to (2.9) only after the system returns or is forced to return to the state x(i) = x(0).

Then the system will be forced to converge to the desired value k = -1/2.

In the problems where the dimensions of x and k are greater than one, the same difficulty prevails. Similarly, measurements of the subgoal are made only after the system returns or is forced to an arbitrary but fixed state. This restriction is particularly severe, since the probability that a particular state will be repeated a finite number of times is practically zero. A less severe but equally valid restriction would be:

> Measure the subgoal according to (2.9) after the system returns or is forced to return to a surface in the state space along which the value of the subgoal varies only with respect to the vector k.

The problem of finding such a surface is illustrated by considering the following second order system:

$$x_1(i+1) = x_1(i) + u(i) \tag{2.11}$$

$$x_2(i+1) = x_2(i) + u(i).$$

Again suppose that a subgoal has been determined which will force the system to converge to a given control law (not the optimal control law in this case).* Let

$$SG(i) = x'(i+1)x(i+1) + u^2(i), \tag{2.12}$$

or equivalently

$$SG(i) = x'(i)A(k_1,k_2)x(i) \tag{2.13}$$

where $A(k_1,k_2)$ is the n x n positive definite symmetric matrix defined by

---

* Since (2.11) is uncontrollable, the optimal solution can be shown not to exist. However, the learning algorithm can still be applied.

$x^2(1 + 2k + 2k^2)$

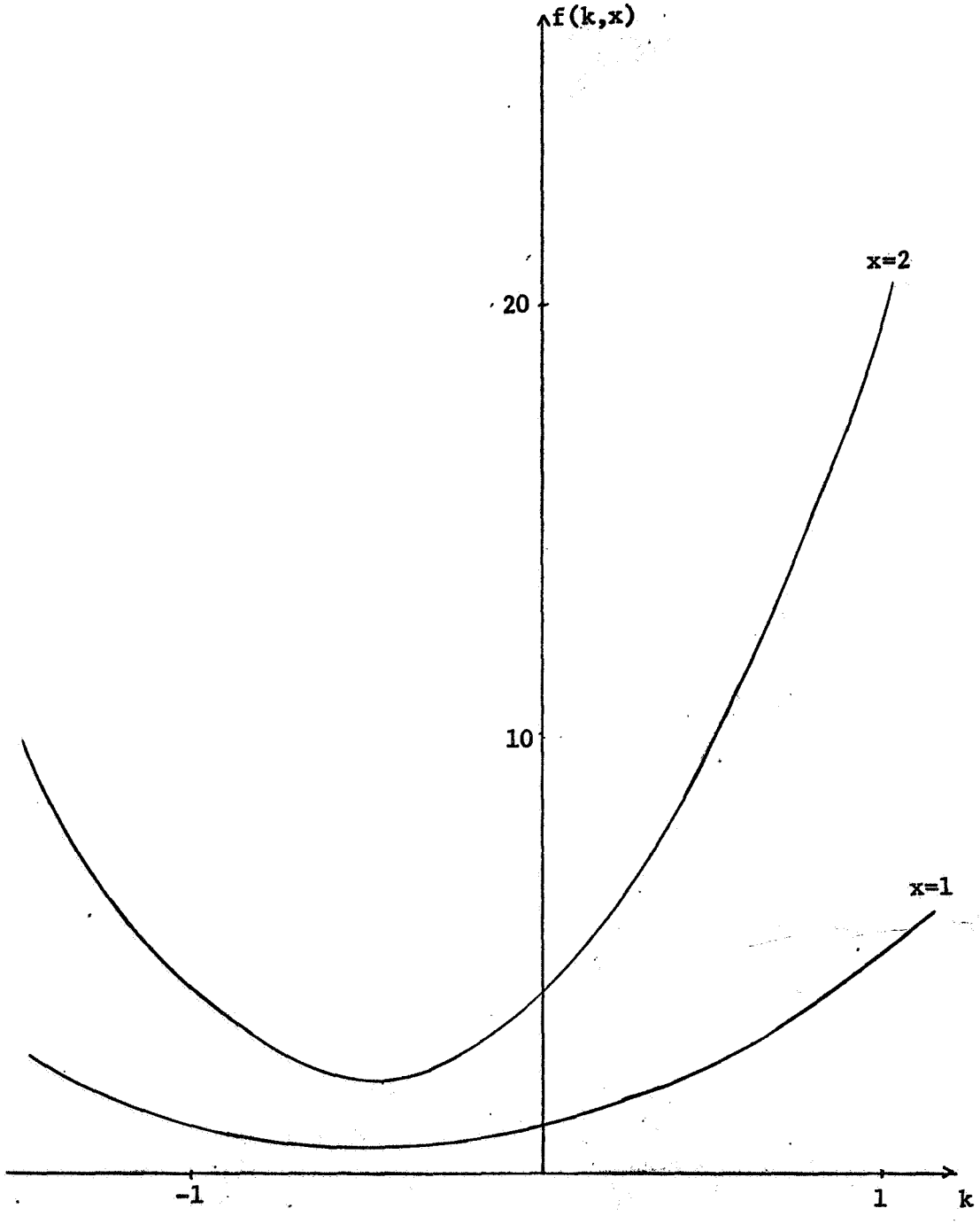

Figure 2.1

$$(2.14)\quad A(k_1,k_2) = \begin{bmatrix} (1+k_1)^2+2k_1^2 & k_1+k_2+3k_1k_2 \\ k_1+k_2+3k_1k_2 & (1+k_2)^2+2k_2^2 \end{bmatrix}$$

Since this example will be used repeatedly to illustrate certain aspects of the learning algorithm, the system equations (2.11) and the subgoal (2.12) will be referred to collectively as System A.

The obvious choice of a surface satisfying the above condition is a constant-value contour in the phase plane. In particular let

$$(2.15)\quad C_1 = \{x(i)\epsilon E_2 : x'(i)A(1,1)x(i) = 24\}$$

be one such contour in the two-dimensional phase plane E_2. If C_1 is to qualify as a valid measurement surface along which the subgoal SG(i) varies only with respect to k, then every other contour corresponding to different values of k_1 and k_2 must not intersect C_1. Yet, as seen in (Figure 2.2), the contour

$$(2.16)\quad C_2 = \{x(i)\epsilon E_2 : x'(i)A(1,-1)x(i)\} = 24$$

intersects C_1, thus invalidating the choice of C_1 as an appropriate measurement surface. In fact, no such contour in the phase plane is an appropriate surface.

This difficulty, however, can be circumvented by redefining the subgoal. Let

$$(2.17)\quad SG(i) = \frac{x'(i)A(k)x(i)}{x'(i)Bx(i)},$$

where B is an n x n positive-definite symmetric matrix whose elements are independent of k. It is observed that (2.17) is constant along rays in the phase plane passing through the origin. Thus, it is necessary to return the system only to some point on the ray prior to measuring the subgoal. Let

$$(2.18)\quad r_1 = \{x(i)\epsilon E_2 : x_1(i) - x_2(i) = 0\}$$

be one such ray. Then along r_1 with B = I, the subgoal as defined by (2.17) has the value

$$(2.19)\quad SG(i) = 1 + 2(k_1+k_2) + 3(k_1k_2) + (3/2)(k_1^2+k_2^2).$$

This expression indicates that the redefined subgoal varies only with respect to k along the ray r_1. In fact, the same is true along any arbitrary ray in the phase plane which passes through the origin. Thus, when x is constrained

$S_1 = \{x \epsilon X : SG(i) = 24, k_1 = 1, k_2 = 1\}$

$S_2 = \{x \epsilon X : SG(i) = 24, k_1 = 1, k_2 = -1\}$

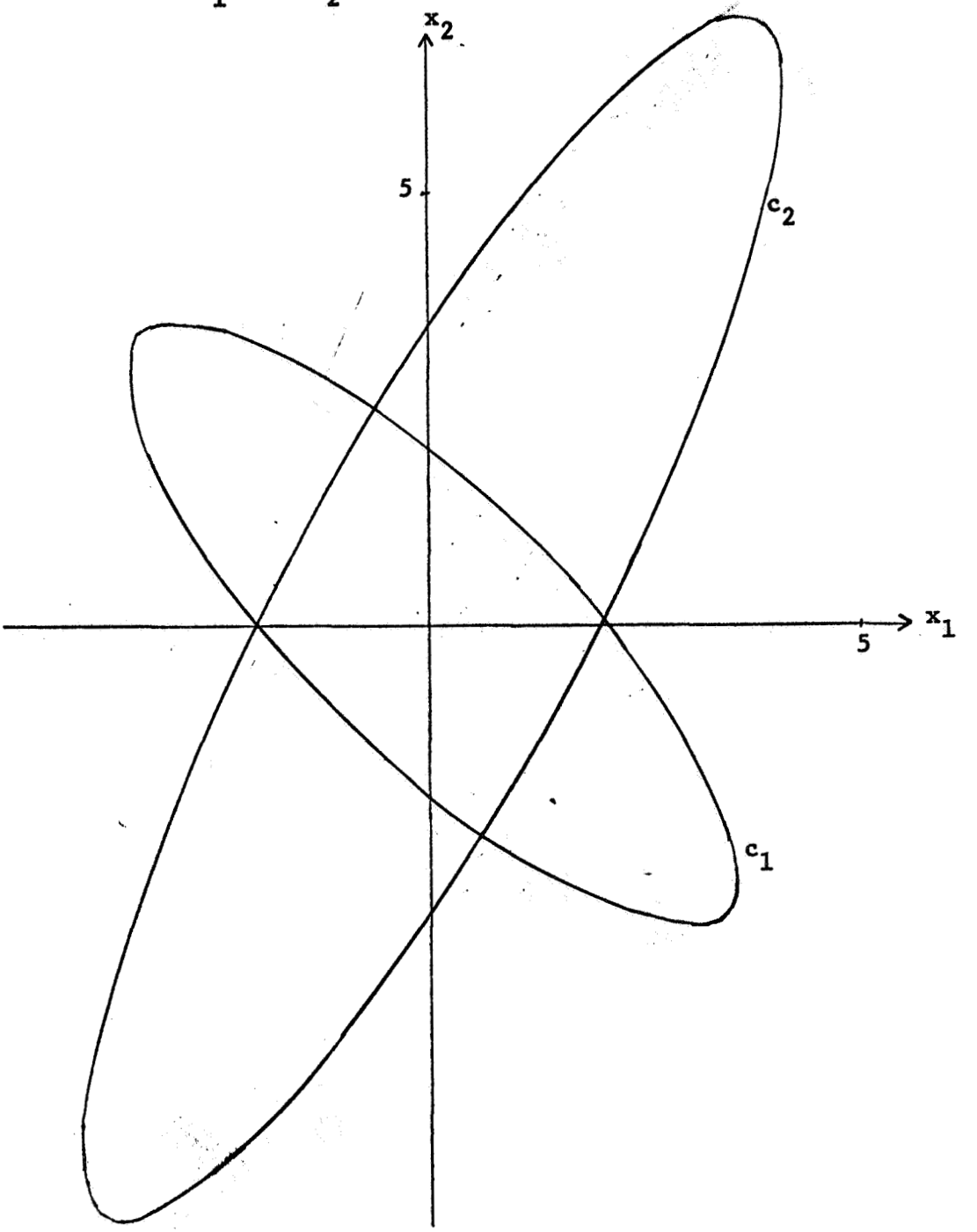

Figure 2.2

to lie along any such ray, the problem is reduced to one of minimizing a function of two variables.

The existence of an absolute minimum of the subgoal in the gain parameter space K will now be examined. The subgoal as defined by (2.17) can be written as

$$SG(i) = k'M(x)k + 2k'M(x)v + s(x), \tag{2.20}$$

where

$$M(x) = \frac{x(i)x'(i)}{x'(i)Bx(i)} \quad \text{is an n x n matrix,}$$

$$v = \phi'Gh \quad \text{is an n x 1 column vector, and}$$

$$s(x) = \frac{x'(i)\phi'G\phi x(i)}{x'(i)Bx(i)} \quad \text{is a scalar.}$$

Equation (2.18) reflects the fact that the subgoal is a quadratic function of the parameter k. The locus of points at which the gradient of the subgoal

$$\nabla_k SG = 2M(x)[\beta k + v] \tag{2.21}$$

equals zero defines the extreme points of SG(i). However, since M(x) is a singular matrix of rank one, the solutions of $\nabla_k SG(i) = 0$ lie on a one-dimensional hypersurface in the space k. Thus, there is no unique minimum with respect to k.

The consequence resulting from the fact that xx' is a singular matrix can be demonstrated in the case of System A, where minimizing the subgoal produces a minimum that lies along the line

$$3k_1 + 3k_2 + 2 = 0. \tag{2.22}$$

Thus, the limit point of the minimization procedure would depend upon both the search method and the point at which the search was initiated. Consider, for example, that the first step of the method employed to find a minimum involved a search in the k_1 direction. Then, starting at the point (1,1), the procedure would converge to the point (-5/3,1) in one step. On the other hand, suppose the initial search was made in a direction parallel to the k_2 axis. Then, with the same initial condition, the point (1,-5/3) would be reached in a single step. Since the principal objective is to converge on a unique point, regardless of the starting point and method, this condition is very undesirable. However, this undesirable result can be eliminated by redefining the subgoal.

Consider first the second order example. Let r_1 and r_2 be two distinct rays in the state space which intersect the origin and let x and y be points on r_1 and r_2 respectively. Consider now the following choice of a subgoal.

$$1 + 2(k_1+k_2) + 3(k_1k_2) + \frac{3}{2}(k_1^2+k_2^2)$$

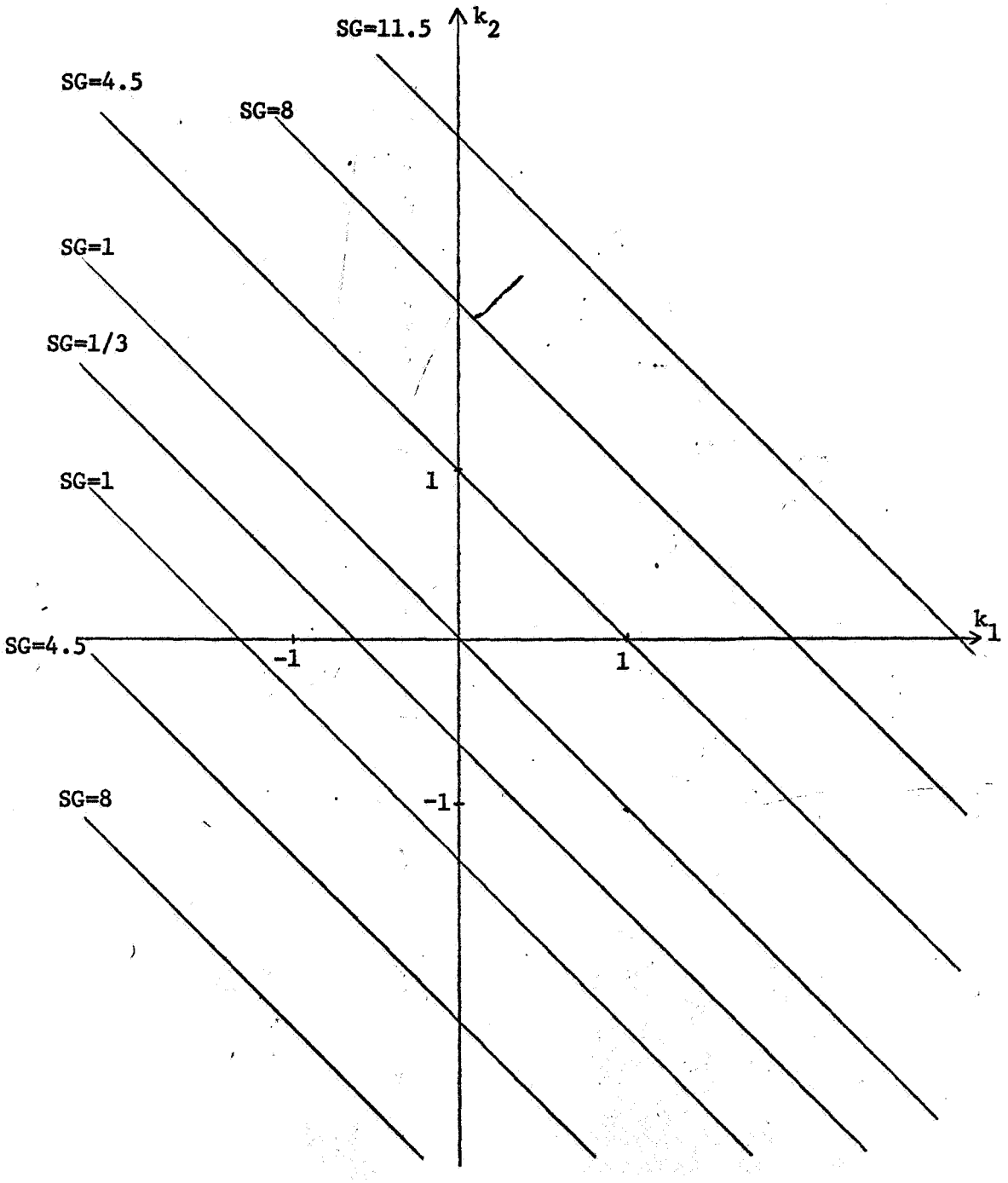

Figure 2.3

Let

$$SG(i) = \frac{x'(i)A(k)x(i)}{x'(i)Bx(i)} + \frac{y'(i)A(k)y(i)}{y'(i)By(i)} \tag{2.23}$$

or equivalently in the form of (2.18).

$$SG\ (i) = k'M(x,y)k + 2k'M(x,y)v + s(x,y) \tag{2.24}$$

where

$$M(x,y) = \frac{xx'}{x'Bx} + \frac{yy'}{y'By} \quad \text{is a 2 x 2 matrix,}$$

$$v = \phi'Gh \quad \text{is a 2 x 1 column vector, and}$$

$$s(x,y) = \frac{x'\phi'G\phi x}{x'Bx} + \frac{y'\phi'G\phi y}{y'By} \quad \text{is a scalar.}$$

In this case, $M(x,y)$ is a positive-definite and non-singular matrix* (see Appendix A). Thus the gradient

$$\nabla_k SG = 2M(x,y)\ [\beta k + v] \tag{2.25}$$

equals zero at the unique point

$$k = -v/\beta = -\frac{\phi'Gh}{h'Gb + \lambda}\ . \tag{2.26}$$

Then by an appropriate choice of G and λ, the system can be forced to converge to any given point in the space k. Furthermore, this point will be uniquely determined by G, λ, and the system parameters, independently of the choice of the rays r_1 and r_2.

The System (2.11) and subgoal (2.23) will be called System B. To illustrate the effect of this modified subgoal on the existence of an absolute minimum, consider the case of System B. Since the choice of r_1 and r_2 is arbitrary**, let r_1 be the x_1 axis and let r_2 be the x_2 axis. Then, with $G = B = I$ and $\lambda = 1$, the subgoal (2.23) has the value

$$SG = (1+k_1)^2 + 2k_1^2 + (1+k_2)^2 + 2k_2^2. \tag{2.27}$$

The graph of the constant-value contours (Fig. 2.4) indicates that an absolute

---

* $M(x,y)$ is required to be positive definite in order that the optimum solution be a minimum.

** It should be noted that the choice of rays for higher order systems, as noted in Appendix A, is not arbitrary, in that the rays must be linearly independent.

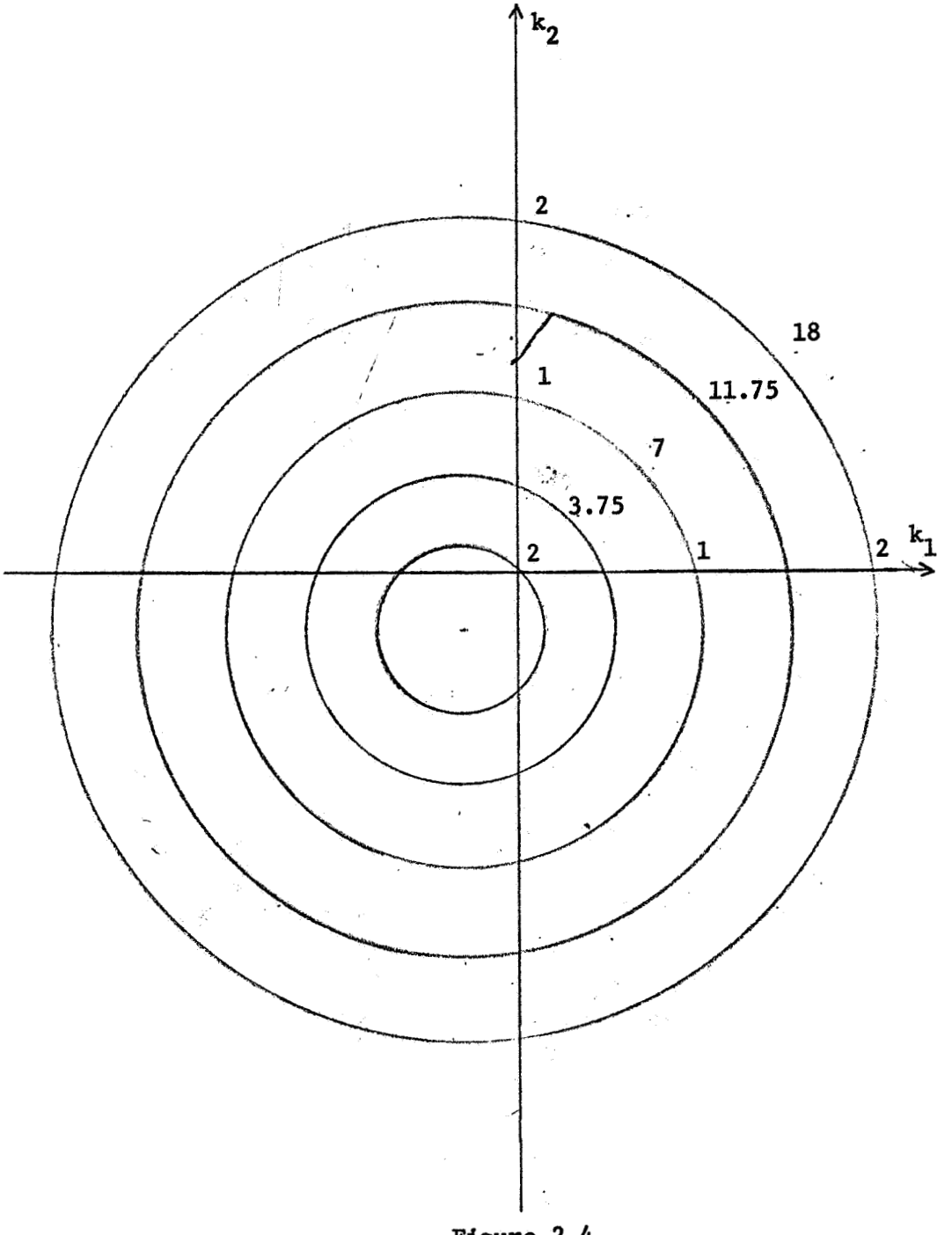

Figure 2.4

minimum is attained at the unique point (-1/3,-1/3).

Using the subgoal defined by (2.27), the procedure for finding the minimum point is outlined below.

a) Starting at some arbitrary initial state x(0) on a given ray r_1, allow the system to continue under the influence of the control $u(i) = k_0'x(0)$. Measure the state at t = T and compute the value of

(2.28) $$\frac{x'(1)Gx(1) + \lambda u^2(0)}{x'(0)Bx(0)} .$$

b) After the system has attained or is forced to attain a state on a second given ray r_2 at time t = nT, allow the system to continue under the influence of the same control law as was applied in step (a). Measure the state at t = nT + T and compute the value of

(2.29) $$\frac{x'(n+1)Gx(n+1) + \lambda u^2(n)}{x'(n)Bx(n)} .$$

c) Add (2.28) and (2.29) to obtain the value of the subgoal corresponding to $k = k_0$.

d) After the system state returns or is forced to return to a point on the ray r_1, repeat steps a, b, and c for a different value of k to obtain the corresponding value of the subgoal.

e) Using this information, adjust k in an appropriate manner and repeat the procedure until a minimum is located. The convergence of this method is assured by the fact that the subgoal is a strictly convex function of the vector k.

For systems of order greater than two, two rays are not sufficient to guarantee that the resultant quadratic will be positive definite. It can be shown, however, that linearly independent vectors $v_1, v_2, \ldots, v_n$ are sufficient to generate a positive definite matrix of the form

(2.30) $$M(v_1, \ldots, v_n) = \sum_{i=1}^{n} \frac{v_i v_i'}{v_i' B v_i} .$$

The proof of this is given in Appendix A. The schedule of operations involved in the learning process must also be amended to include measurements at these additional rays. For an n^{th} order system, step (b) of the preceeding five-step outline of the minimization routine should be repeated n - 2 times before a complete measure of the subgoal is obtained.

In the above procedure, the probability that the system will encounter a particular ray in the state space more than once is practically zero.

Hence, it becomes necessary to implement a controller which will force the system to return to a given ray a finite number of times to allow the learning controller to make the appropriate measurements. State feedback will be employed to force the system to behave in this prescribed manner. In order to demonstrate this procedure, let A represent the system matrix of the closed loop system, i.e.,

$$A = \phi + hk' . \tag{2.31}$$

Then N x N matrix A can be expanded in dyadic form as

$$A = \sum_{i=1}^{N} e_i > \lambda_i < e_i^+ , \tag{2.32}$$

where

i) λ_i is an eigenvalue of A,

ii) e_i is the eigenvector corresponding to λ_i,

and

iii) e_i^+ is the dual vector associated with the eigenvector e_i.

Using this notation, the system response to an initial condition x(0) can be written as

$$x(nT) = \sum_{i=1}^{N} e_i > \lambda_i^n < e_i^+, x(0) > . \tag{2.33}$$

Since the eigenvectors constitute a bisis for the state space, the initial state x(0) can be expressed as a linear combination of these eigenvectors.

$$x(0) = \sum_{j=1}^{N} \alpha_j e_j . \tag{2.34}$$

Substituting (2.34) in (2.33), and utilizing the fact that

$$< e_i^+, e_j > = \delta_{ij} , \tag{2.35}$$

the response can be written as

$$x(nT) = \sum_{i=1}^{N} \alpha_i \lambda_i^n e_i . \tag{2.36}$$

Thus it is easily seen that if magnitudes of all but one of the eigenvalues, say λ_1, are forced to have values 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 eigenvalues.

Consider again the system described by (2.11). The eigenvalues of the system matrix are both one and the eigenvectors of the open loop system are $e_1' = (1,0)$, and $e_2' = (0,1)$. With $u = -0.9x_2$, the eigenvalues of the closed loop system are $\lambda_1 = 1.0$, $\lambda_2 = 0.1$. The corresponding eigenvectors are $e_1' = (1,0)$, $e_2' = (1,1)$. Thus, the system response will approach a point on the x_1 axis. Likewise, with $u = -0.9x_1$, the state vector will tend to align itself with the x_2 axis.

The entire strategy is now applied to System B. A method proposed by M.J.D. Powell (see Appendix B) is used to determine the point in the K space at which the subgoal is a minimum. The advantage of this particular method is that no derivative measurements are required. The search is begun at the point K = (1,1) and converges to the point K = (-1/3,1/3) as desired (Fig. 2.5). The response of the system for one complete measurement cycle is shown in Fig. 2.6. The letters a,b, and c adjacent to each segment of the trajectory correspond to the steps a, b, and c in the outline on page

This completes the development of the learning algorithm. The three principal results of this section are summarized below:

1) Since the subgoal must be held constant with respect to x while it is minimized with respect to k, the originally defined subgoal was altered to allow measurements to be made along a line in the state space rather than at a single point.

2) To guarantee the existence of a unique limit point, the subgoal was again altered. If the subgoal was evaluated on the basis of measurements on n distinct rays in the state space, it was shown that the resulting function was strictly convex in the vector k thus assuring the existence of an absolute minimum.

3) It was shown that state-feedback can be used to force the system to return to the two distinct rays in the state space that are required in the evaluation of the subgoal.

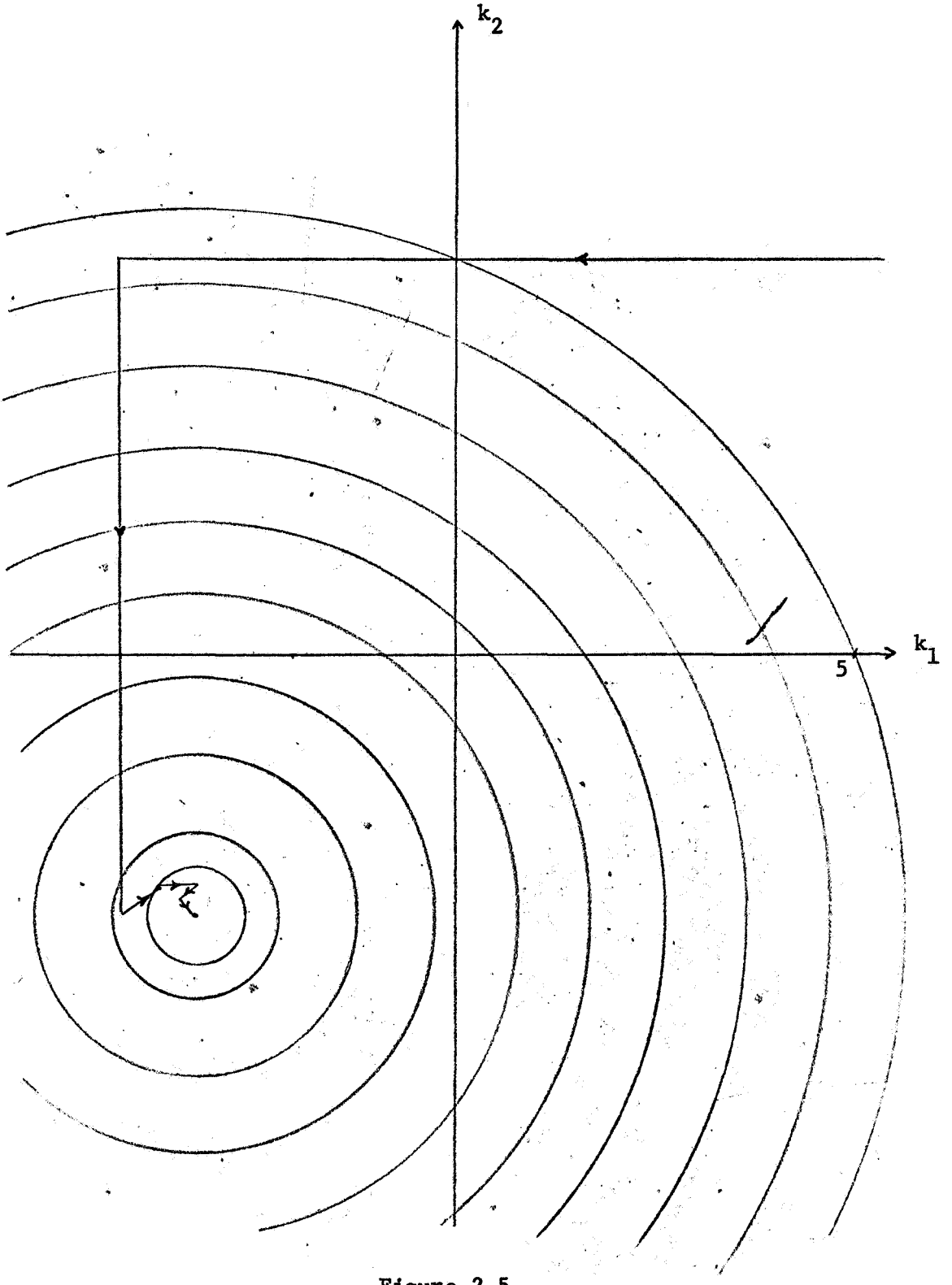

Figure 2.5

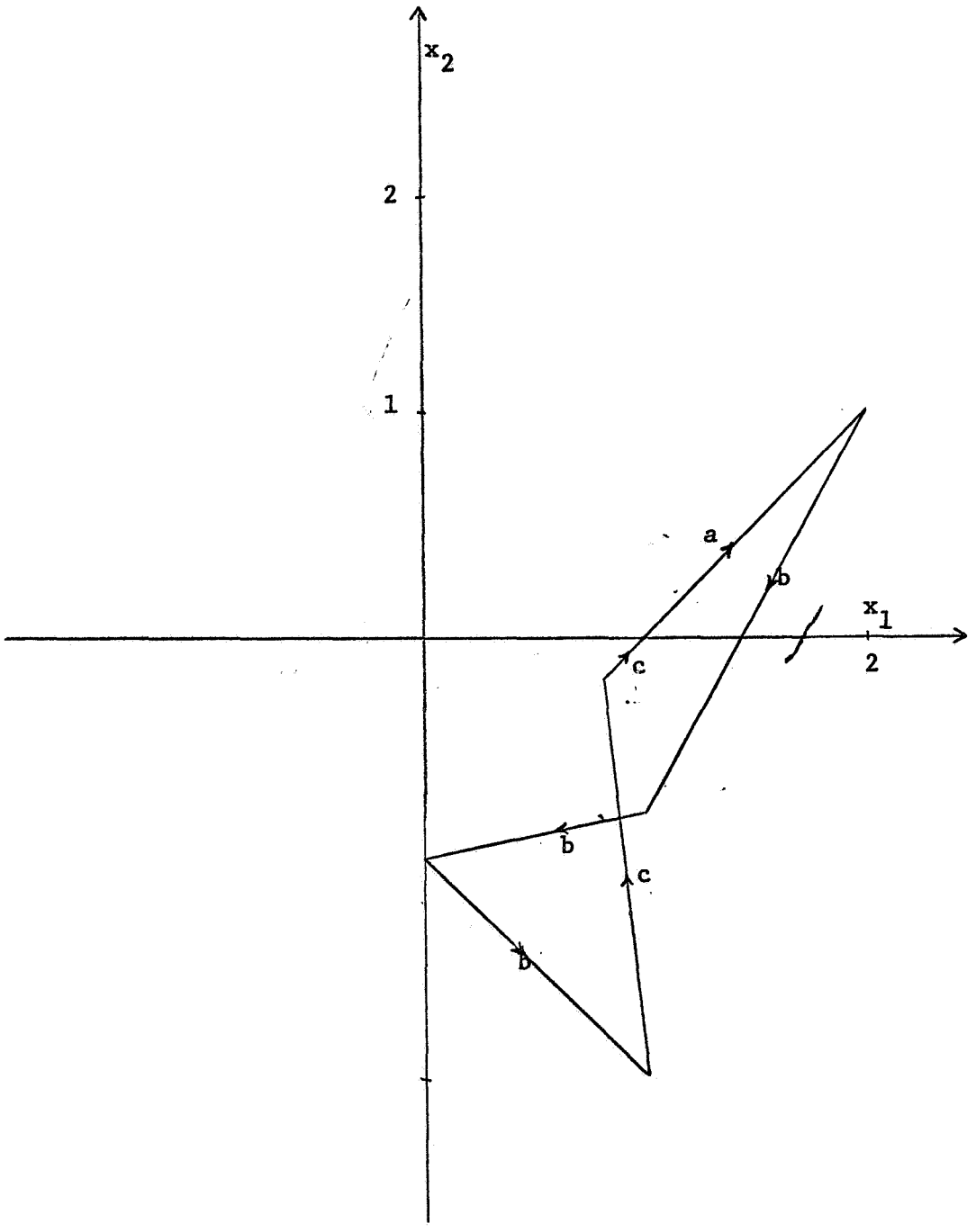

Figure 2.6

Example

Consider the second-order system described by the vector differential equation

$$\dot{x}(t) = Ax(t) + bu(t), \tag{3.1}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix}, \text{ and } b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.2}$$

If the states are sampled uniformly every T seconds and the continuous plant is preceded by a zero-order hold, then the equivalent vector difference equation is

$$x(i+1) = \phi x(i) + hu(i), \tag{3.3}$$

where

$$\phi = \begin{bmatrix} 1 & a^{-1}[1-\exp(-aT)] \\ 0 & \exp(-aT) \end{bmatrix} \tag{3.4}$$

and,

$$h = \begin{bmatrix} a^{-2}[aT-1\exp(-aT)] \\ a^{-1}[1-\exp(-aT)] \end{bmatrix} \tag{3.5}$$

The principal objective is to control this plant optimally with respect to the infinite time performance index

$$I(u,x_0) = \sum_{i=0}^{\infty} x'(i+1)Qx(i+1) + u^2(i) \tag{3.6}$$

where, in this example, Q is chosen to be positive definite, and is given the value

$$Q = \begin{bmatrix} 20 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.7}$$

A priori, it is known that the actual a = 1.0 $\pm$ 20%. Thus, the "best" subgoal is determined by solving the discrete optimal control problem for the nominal system, i.e., when a = 1.0. With T = 0.1 sec. the optimal

control law is found to be

$$u(i) = -3.9853x_1(i) - 2.1533x_2(i), \tag{3.8}$$

while the associated minimum value of the cost function is $x'(0)Px(0)$ and

$$P = \begin{bmatrix} 138.585 & 44.6894 \\ 44.6894 & 22.4974 \end{bmatrix}. \tag{3.9}$$

The corresponding subgoal is quadratic

$$SG(i) = x'(i+1)Gx(i+1) + u^2(i), \tag{3.10}$$

with

$$G = P + Q = \begin{bmatrix} 158.585 & 44.6894 \\ 44.6894 & 23.4974 \end{bmatrix} \tag{3.11}$$

The next step is the selection of appropriate feedback gains so that the resulting closed-loop systems exhibit: 1) an unstable mode and 2) a stable mode whose influence dies out rapidly. The eigenvalues of the nominal system are $\tilde{\lambda}_1 = 1.0$ and $\tilde{\lambda}_2 = \exp(-0.1) = 0.9048$, indicating that no state feedback is required to generate the necessary unstable mode. It is only necessary to decrease the magnitude of $\tilde{\lambda}_2$ in order to effect the domination by the unstable mode. With

$$u(i) = -8.46\, x_2(i), \tag{3.12}$$

$\tilde{\lambda}_1$ remains unchanged while $\tilde{\lambda}_2 = 0.1$. The corresponding nominal eigenvectors are

$$\tilde{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } \tilde{e}_2 = \begin{bmatrix} -0.06 \\ 1 \end{bmatrix}. \tag{3.13}$$

A second unstable mode is generated by applying

$$u(i) = x_1(i) - 8.46\, x_2(i). \tag{3.14}$$

In this case, the eigenvalues of the resulting closed-loop nominal system are $\pi_1 = 1.79$ and $\pi_2 = -0.21$, and the corresponding nominal eigenvectors are

$$\tilde{f}_1 = \begin{bmatrix} 1 \\ 0.951 \end{bmatrix}, \quad \tilde{f}_2 = \begin{bmatrix} -0.054 \\ 1 \end{bmatrix}. \tag{3.15}$$

The vectors $\tilde{e}_1$ and $\tilde{f}_1$ are then the approximate, non-collinear vectors used to measure the value of the subgoal. Each evaluation of the subgoal at various points in the parameter space K consists of the following steps:

1) Starting at some initial state x(0), apply the control option $u(i) = -8.46\, x_2(i)$. Maintain this control function until t = nT when the following condition is satisfied:

$$\frac{\langle x(n), x(n-1)\rangle}{\|x(n)\|\ \|x(n-1)\|} > 1 - \varepsilon \quad . \tag{3.16}$$

In the above expression, ε is a small positive number and $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote respectively the inner product and norm relations defined on the two dimensional Euclidean space. At t = nT the vector output is assumed to be coincident with the vector $e_1 = \tilde{e}_1 + \Delta e_1$. The perturbation in e_1, Δe_1, is due to the inexact knowledge of the parameter a.

2) Evaluate the partial subgoal for gains k_1 and k_2 by applying the control $u(n) = k_1 x_1(n) + k_2 x_2(n)$ and evaluating the functional

$$SG(n) = \frac{x'(n+1)Gx(n+1) + u^2(n)}{x'(n)Bx(n)} \tag{3.17}$$

at time t = (n+1)T. B is an arbitrary positive definite matrix. In this example, B is chosen as the identity matrix.

3) Starting at the point x(n+1), apply the control option $u(i) = x_1(i) - 8.46\, x_2(i)$. Maintain this control function until t = mT (m > n) when the following condition is satisfied.

$$\frac{\langle x(m), x(m+1)\rangle}{\|x(m)\|\ \|x(m-1)\|} > 1 - \varepsilon \tag{3.18}$$

4) Evaluate the partial subgoal for gains k_1 and k_2 by applying the control $u(m) = k_1 x_1(m) + k_2 x_2(m)$ and evaluation of the functional

$$SG(m) = \frac{x'(m+1)Gx(m+1) + u^2(m)}{x'(m)Bx(m)} \tag{3.19}$$

at time t = (m+1)T.

5) Add the partial subgoals determined at times t = nT and t = mT (steps 2 and 4) to determine the value of the complete subgoal for the gains k_1 and k_2.

The entire procedure is repeated as many times as is necessary to complete the search for the minimum.

The minimum search routine used in this example was the method proposed by Powell (Appendix B). The maximum of the function along a line in the parameter space k, i.e.,

$$\min_{\lambda} f(k_0 + \lambda k_\Delta) \tag{3.20}$$

was determined by measuring 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

$$\lambda = (\alpha/2)\ \frac{(FN - FP)}{(FN + FP)}\ , \tag{3.21}$$

where

$$\begin{aligned} FN &= f(k_0 - \alpha k_\Delta) - f(k_0) \\ FP &= F(k_0 + \alpha k_\Delta) - f(k_0)\ . \end{aligned} \tag{3.22}$$

The entire minimum search routine including the above procedure was applied to the second-order example. First, a check of the accuracy of this routine was made by applying the learning algorithm to the nominal system. A measure of the accuracy is obtained by comparing the learned linear control law with the optimal linear control law. For this example, the maximum deviation in either component of the learned control gain vector was less than 10^{-5}. The learning algorithm was then used to determine k_L assuming the actual $a = 0.9$. The minimum search begins at the point k_F. The succession of points encountered in the search routine is shown below in tabular form.

(3.23)

k_1	k_2
-3.9853	-2.1533
-4.0043	-2.1533
-4.0043	-2.1694
-4.0013	-2.1669
-4.0013	-2.1729
-3.9994	-2.1713
-3.9961	-2.1790
-3.9961	-2.1790

Thus, $k_L' = (-3.9961, -2.1790)$. The entire learning process was completed in less than 13 seconds -- real time, not computer time.

A check of the learning system's advantage can be made by comparing the

cost associated with the learned system with the cost obtained by implementing a system with the fixed feedback gain determined by the optinal solution for the nominal plant. The cost for the learned system is given by

$$x'(0)\left[\sum_{n=0}^{\infty} x'(n+1)Qx(n+1) + \lambda u^2(n)\right]x(0) \tag{3.24}$$

$$= x'(0)P(\infty)x(0)$$

where

$$P(N) = \sum_{n=1}^{N} \Phi'^n Q \Phi^n + \lambda \sum_{n=0}^{N-1} \Phi'^n k_L k_L' \Phi^n \quad \text{and} \tag{3.25}$$

$$\Phi = \phi + h k_L'.$$

The matrix $P(\infty)$ can be approximated by computing $P(N)$ for increasing values of N until the increase in $P(N)$ with each increment of N is negligible. The cost for the learned system above was found to be

$$I_L(x(0)) = x'(0)\begin{bmatrix} 136.994 & 44.6975 \\ 44.6975 & 23.3393 \end{bmatrix}x(0). \tag{3.26}$$

The cost associate with the second alternative design was similarly determined and found to be

$$I_F(x(0)) = x'(0)\begin{bmatrix} 137.014 & 44.6971 \\ 44.6971 & 23.3966 \end{bmatrix}x(0). \tag{3.27}$$

The difference in cost for the two options is

$$I_F(x(0)) - I_L(x(0)) = x'(0)Dx(0) \tag{3.28}$$

where

$$D = \begin{bmatrix} 1.97351 \times 10^{-1} & -3.99786 \times 10^{-3} \\ -3.99786 \times 10^{-3} & 6.83201 \times 10^{-2} \end{bmatrix} \tag{3.29}$$

Since the matrix D is positive definite, the learned system is shown to be superior for all initial conditions.

An identical analysis was performed for different values of the parameter

a, within the intended range (0.8,1.2). The results are shown graphically in Figures 3.1 and 3.2. Figure 3.1 is a graph of d_{11} versus a, while Figure 3.2 is a plot of the determinant of D versus a. The fact that both graphs lie above the a axis indicates that the comparison matrix D is positive definite for all a. Thus, it is shown that learning is superior for all initial conditions and for all values of a, in the given range.

Summary and Conclusions

An algorithm has been developed for implementing a learning controller based on Jones' formulation of a learning subgoal. Jones' original form of a subgoal was appropriately modified to facilitate the evaluation of the subgoal from measurements of the system state and also to guarantee absolute convergence of the process. This procedure was demonstrated for a second order example and this learning controller was shown to yield better results than a fixed control policy based on the best available model of the plant.

The application of the method is restricted to linear, stationary systems with quadratic performance measures. In addition, the control variable is assumed to be unconstrained. The advantage of this method over a reinforcement-type learning controller, assuming that the quantization of the control and state variables is fine enough, is realized in a considerable reduction in the number of variables to be stored and the time required to complete the learning process.

The discussion of this algorithm is by no means complete. Although, the extension of this method has been theoretically justified, it would be advantageous to apply the procedure to a third or fourth order system. It is anticipated that for these higher-order systems the problem of disigning proper state feedback in order to generate the appropriate rays will be more complex and tedious. Further attention should be given to the choice of these measurement vectors and its effect on the rate of convergence of the algorithm.

Appendix A

Let $r_1, r_2, \ldots, r_n$ be n distinct rays in the n dimensional Euclidean space E^n. Let v_i be an arbitrary vector coincident with the ray r_i. The purpose of this section is to demonstrate that the matrix

$$M(v_1,\ldots,v_n) = \sum_{i=1}^{n} \frac{v_i v_i'}{v_i' B v_i} \tag{A1}$$

is positive definite for every set of linearly independent vectors $v_1, v_2, \ldots, v_n$. The proof is dependent on the fact that $M(v_1,\ldots,v_n)$ can be represented as the

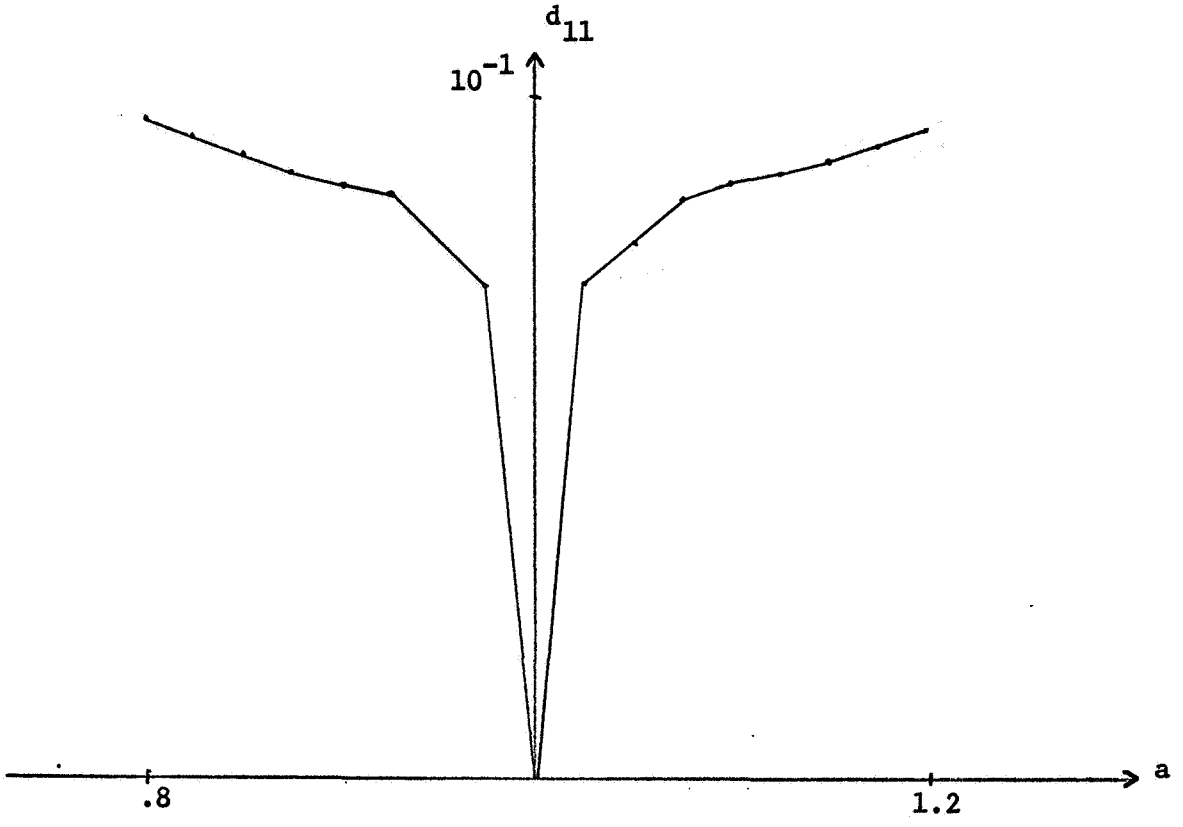

Figure 3.1

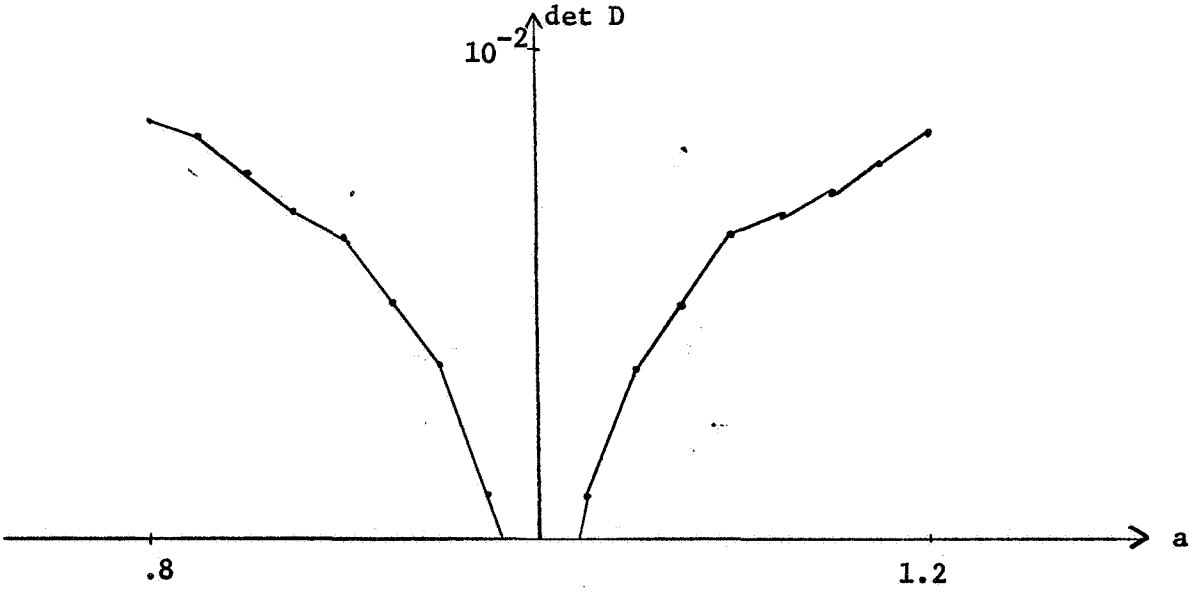

Figure 3.2

product of two non-singular matrices.

Let $u_i = v_i/(v_i'Bv_i)^{1/2}$. Then in order to prove that $M(v_1,\ldots,v_n)$ is positive definite, it is sufficient to prove that

$$M(u_1,\ldots,u_n) = \sum_{i=1}^{n} u_i u_i' \tag{A2}$$

is positive definite.

Let P be an n x n matrix with the vectors $u_1,\ldots,u_n$ as the columns of P, i.e.,

$$P = [u_1 \quad u_2 \quad \ldots. \quad u_n] . \tag{A3}$$

Then P' is an n x n matrix whose rows are the vectors $u_1', u_2', \ldots, u_n'$, i.e.,

$$P' = \begin{bmatrix} u_1' \\ u_2' \\ \vdots \\ u_n' \end{bmatrix} . \tag{A4}$$

Let M be the symmetric n x n matrix formed by the matrix product PP'. Then, observing the rules for block multiplication of matrices, the matrix M can be expanded as follows

$$M = \sum_{i=1}^{n} u_i u_i' . \tag{A5}$$

Thus, it has been shown that $M(u_1,u_2,\ldots,u_n)$ can be represented as the product of two matrices. If the vectors u_i, or equivalently the vectors v_i, are linearly independent, the matrices P and P' are both non-singular and the symmetric matrix $M(v_1,v_2,\ldots,v_n)$ is positive definite. [4]

Appendix B

Powell [5] describes a simple variation of the well-known method of minimizing a quadratic function of n variables by changing one parameter at a time. The function to be minimized is assumed to be of the form

$$f(x) = x'Ax + b'x + c \tag{B1}$$

where x and b are n dimensional vectors, A is a positive definite n x n matrix, and c is a scalar. The procedure commences with a search along each

of the co-ordinate directions. The results of this search form the basis for generating a new search direction p to replace one of the co-ordinate directions. The search is then repeated for this new set of directions and a second linearly independent search direction q is determined by the outcome of this search. The algorithm used to generate p and q is chosen so that the vectors p and q are conjugate with respect to the matrix A, i.e., $p^{t}Aq = 0$. Furthermore, after each iteration or search down the n linearly independent directions, a new search direction is generated which is conjugate to all the direction generated by the previous iterations. Then, after n iterations, all the directions are mutually conjugate and the absolute minimum of the quadratic is found. The algorithm used in the generation of the new conjugate directions is outlined below.

Let $(\xi_1, \xi_2, \ldots, \xi_n)$ be the n linearly independent directions of search and let p_0 be the point in the n dimensional parameter space at which the iteration begins. Each iteration consists of the following steps.

1) For $r = 1,2,\ldots,n$, calculate λ_r 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 $p_n + \lambda(p_n - p_0)$.

At the conclusion of the n^{th} iteration, the exact minimum of the quadratic function f(x) will have been located.

References

1. Waltz, M. D., Fu, K. S., "A Learning Control System," Reprints of the JACC, 1964.
2. 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.
3. Jones, L. E., "On the Choice of Subgoals for Learning Control Systems," IEEE Transactions on Automatic Control, Vol. AC-13, No. 6, December, 1966, pp. 613-620.
4. Finkbeiner, D. T., Introduction to Matrices and Linear Transformations, W. H. Freeman and Company, San Francisco, 1960, p. 188.
5. Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Perivates," Computer Journal, Vol. 7, 1964, pp. 155-162.