NUMERICAL COMPUTATION OF NEARLY-OPTIMAL FEEDBACK CONTROL LAWS AND OPTIMAL CONTROL PROGRAMS

by

ALAN GORDON LONGMUIR

B.A.Sc., University of British Columbia, 1964

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in the Department of

Electrical Engineering

We accept this thesis as conforming to the

required standard

Research Supervisor .....

Members of Committee .....

.....

Head of Department .....

Members of the Department

of Electrical Engineering

THE UNIVERSITY OF BRITISH COLUMBIA

March, 1968

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver 8, Canada

Date April 1, 1968

ABSTRACT

An investigation is made into the approximate synthesis of optimal feedback controllers from the maximum principle necessary conditions. The overall synthesis can be separated into two phases: the computation of optimal open-loop controls (control programs) and trajectories from the necessary conditions, and the processing of this data to obtain an approximate representation of the optimal control as a state function.

A particular technique for approximating the optimal feedback control from the optimal open-loop controls and trajectories is proposed and examined in Part I of the thesis. Parameters in a prechosen suboptimal controller structure are computed such that a sum of integral square deviations between the suboptimal and optimal feedback controls is minimized. The deviations are computed and summed over a certain set of trajectories which "cover" the system operating region. Experimentation with various controller structures is quite feasible since the controller parameters are computed by solving linear algebraic equations. Examples are given to illustrate the application of the technique and ways in which suitable controller structures may be found. If general purpose functions are to be used for this purpose, piecewise polynomial functions are recommended and techniques for their use are discussed. The synthesis method advocated is evaluated with respect to control sensitivity and instrumentation and compared to alternative procedures.

Part II is concerned with the computation of optimal

control programs, the most time consuming numerical task in the synthesis procedure. A new numerical optimization technique is presented which extends the function space Newton-Raphson method (quasilinearization) to a more general terminal condition. More significantly, a generalized Ricatti transformation is employed, and as a consequence, the integration of the unstable coupled canonical system is eliminated. Examples are given as evidence of the improved numerical qualities of the new algorithm. This method is one example of a class of algorithms, defined and developed in the thesis, called second variation methods. Some methods in this class have previously appeared in the literature but they are developed in the thesis from a unified point of view. The recognition of this class allows the relationships between the various methods to be seen more clearly as well as allowing techniques developed for use in one algorithm to be used in others.

TABLE OF CONTENTS

Page

PART II COMPUTATION OF OPTIMAL CONTROL PROGRAMS

LIST OF TABLES

LIST OF ILLUSTRATIONS

NOTATION

Let $S(x,y,\ldots)$ be a scalar function of the vectors $x,y,\ldots$ Assume that x has n components and y has m components. Then

$$\frac{\partial S}{\partial x} = S_x \triangleq (S_{x_1}, S_{x_2}, \ldots, S_{x_n})$$

and

$$\frac{\partial^2 S}{\partial x \partial y} = S_{xy} \triangleq \begin{bmatrix} S_{x_1 y_1} & \cdots & S_{x_1 y_m} \\ \vdots & & \vdots \\ S_{x_n y_1} & \cdots & S_{x_n y_m} \end{bmatrix}$$

If $f(x,y,\ldots)$ is a vector function with k components,

$$\frac{\partial f}{\partial x} = f_x \triangleq \begin{bmatrix} f_{1x_1} & \cdots & f_{1x_n} \\ \vdots & & \vdots \\ f_{kx_1} & \cdots & f_{kx_n} \end{bmatrix}$$

A superscript T denotes the transpose of a vector or matrix.

ACKNOWLEDGEMENT

I would like to thank Dr. E.V. Bohn, supervisor of this project, for his continued interest and guidance. In addition, the support which I have received from the National Research Council of Canada in the form of a Bursary in 1964-65 and Studentships in subsequent years is gratefully acknowledged.

I have had many helpful discussions about the material in this thesis with Mr. M. Masak.

The original manuscript was typed by my wife, Barbara, to whom I owe a large debt of gratitude for her patience and encouragement throughout my postgraduate work.

1. INTRODUCTION

1-1 The Optimal Control Problem.

The optimal control of dynamical systems is of interest to control system engineers for several reasons. There may be direct economic returns from the use of optimal control such as increased production, more efficient use of costly inputs, or tighter control of product quality. An optimal control system may be sought to cope with demanding and conflicting operational specifications. An example of this type of design problem is the requirement for an extremely rapid correction of system error while using a control element whose output cannot exceed a given maximum value. Another motivation for studying the optimal control problem is the great potential value of the optimal criterion in design methodology. Inherent in the formulation of the optimal feedback control problem is the question of control system synthesis. A complete solution of the optimal feedback control problem would immediately yield the proper controller structure.

A mathematical statement of the optimal control problem will now be given. The statement is of sufficient generality to cover all cases considered in this thesis. It is intended that the following discussion serve to make clear both the nature of the problem and the meaning of the solution. Detailed and precise conditions on the problem will not be stated here (see [1], for example).

The physical processes of interest are those which can be suitably modelled by a system of first-order ordinary differential equations

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

where $x = (x_1,\ldots,x_n)^T$ is the state vector ([2], Chpt. 1), $u = (u_1,\ldots,u_r)^T$ is the control vector, and $\dot{x}$ denotes dx/dt. No explicit account will be taken of stochastic disturbances of the process, that is, the model (1.1) is assumed to be deterministic. Solutions of Eq. (1.1) are referred to as trajectories.

The control signal $u(t)$ is assumed to act during an interval of interest called the control period defined as $t_f - t_o$ where t_o is the initial time and t_f is the terminal time. At the terminal time t_f, it is required that the state-time (x,t) belong to a certain set S, called the target set or terminal manifold, defined by

$$\Psi(x(t_f),t_f) = o \tag{1.2}$$

where $\Psi = (\Psi_1,\ldots,\Psi_m)^T$. The terminal manifold S is of dimension $(n+1)-m$ in the state-time product space. For example, if $\Psi = x(t_f)$, then S is the origin of state space which is of dimension one in the state-time space. The actual value of t_f is specified by Eq. (1.2) either explicitly (fixed-time problem) or implicitly (free-time problem). In the free-time case, t_f is determined as the first instant at which the optimal trajectory intersects S.

In general, the control function values $u(t)$ may be restricted to lie in a certain subset U of E^{r*} called the set of permissable control values, and the trajectories $x(t)$ may be required to remain within a certain subset X of E^n.

For a particular initial state x_o and initial time t_o, a control $u(t)$, $t_o \leq t \leq t_f$, is called allowable if it has values belonging to U and if it transfers the system from (x_o,t_o) to a point $(x(t_f),t_f)$ in S in such a way that the trajectory does not leave X. It should be noted that there may be no allowable controls. In general, the existence of such controls is extremely difficult to establish a-priori. In what follows, it will always be assumed that more than one allowable control exists.

The optimal control problem may now be stated as follows: for the initial state-time pair (x_o,t_o), find the allowable control u^* which minimizes the functional

$$J(u) = \emptyset(x(t_f),t_f) + \int_{t_o}^{t_f} F(x,u,t)dt \qquad (1.3)$$

where $\emptyset$ and F are scalar functions.

In order to begin a mathematical attack on the optimal control problem it is necessary to impose restrictions on the generality of the problem such as requiring the continuity or differentiability of some of the functions introduced above. Having done this, the optimal control

---

* E^r is r-dimensional Euclidean space.

problem can be reduced to a problem in the calculus of variations [1] or, from a slightly different point of view, the maximum principle of Pontryagin [3]. Necessary conditions and certain sufficiency conditions for an optimal control are given in the literature [1], [3], [4].

The nature and meaning of the solution of the problem posed above needs to be emphasized. To emphasize that the optimal control is a vector function of time based on the particular initial state-time pair (x_o,t_o), it should be written as

$$u = u^*(t;x_o,t_o) \tag{1.4}$$

The corresponding optimal trajectory is denoted

$$x = x^*(t;x_o,t_o) \tag{1.5}$$

With respect to the target set (1.2) and the performance functional (1.3), the control u^* is optimal for all states lying on the optimal trajectory, that is, for all pairs $(x(t),t)$ where $x(t) = x^*(t;x_o,t_o)$, $t_o \leq t \leq t_f$ ("principle of optimality" [5]). A control in this form is variously called a control program or open-loop control. There is no feedback to the controller of the actual system progress; the control input signal to the process is programmed.

1-2 Optimal Feedback Control.

In most control systems, satisfactory operation

can only be obtained by determining the controlled inputs to the system on the basis of measured process variables, that is, by using feedback control. Open-loop control is feasible only if the disturbances acting upon the process are very small. By disturbances are meant inputs to the system which are either not deterministic or not feasible to treat as deterministic. Thus, in certain cases, a varying set-point may be considered as a disturbance.

The most general form for the optimal feedback control is

$$u = u^*(x,t) \tag{1.6}$$

a function of the present state $x(t)$ and possibly also of time t. If the optimal feedback control law is employed, then irregardless of the state of the system which results from any disturbance, the control signal acting upon the process will still be optimal. Of course, "optimal" here means optimal for the system mathematical model where no account was taken of this particular disturbing signal.

In general, the optimal feedback control law will depend upon all of the state variables. Thus, the realizability of the optimal control law is contingent upon the "accessibility" of all the state variables. A state variable is said to be accessible if it is possible to determine its value at a given instant from measurements of process variables at that instant. In general, little can be said about the optimal control law when inaccessible

state variables exist. For one important problem class, however, it is known ([6], Chpt. 6) that the optimal system is achieved by substituting for the actual state in the control law, the best estimate of the state (in the Kalman sense[7]). Thus, the optimal control system consists of two separate functions; estimation and control. The design problem is decoupled into two subproblems: determining the optimal control law $u^*(x,t)$ and determining the best state estimator. Although this procedure has been proven optimal for a special problem class only, it is an intuitively reasonable one to adopt in any case. In this thesis, the estimation problem is not considered further.

An alternative procedure is to synthesize a suboptimal control law based on accessible state variables only. This alternative is discussed further in Section 4-2.

1-3 Scope of the Thesis.

In part I of the thesis, the problem of synthesizing a nearly-optimal feedback controller is investigated. Chapter 2 contains the development of a synthesis technique based on the solution of the optimal programming problem. This method processes the numerically computed trajectory data in a simple and efficient manner to obtain a suboptimal control. Examples are given to illustrate the use of the method together with the difficulties involved in its employment. Chapter 3 deals with controllers having a piecewise polynomial structure. Piecewise polynomial functions are

introduced in order to overcome the difficulties inherent in high-order polynomial approximation. In Chapter 4, the approach to suboptimal control law synthesis taken in this thesis is evaluated and compared with alternative procedures.

In applying the synthesis procedure of Part I, the two most difficult tasks facing the designer are the specification of a suitable controller structure (discussed in Chapter 3) and the computation of the optimal control programs. Since, in general, many optimal programming problems must be solved to provide the data required in the synthesis, it is essential that the numerical algorithms employed for solving the open-loop problem be extremely efficient with respect to computer solution time. The solution of optimal programming problems is the subject of Part II of the thesis. The division of the thesis into two parts was made because the material in Part II, while intimately connected with the procedure of Part I, also has applications in other areas.

In Chapter 5, a new technique is presented which is an extension and modification of the function space Newton-Raphson method [8]. The new algorithm has a relatively large region of convergence, a rapid rate of convergence in a neighborhood of the desired extremal, and numerical stability properties which are a considerable improvement over those of the original method. An entire

class of algorithms, herein called second variation methods, is derived in Chapter 6. Included in this group of algorithms are some of the currently popular numerical optimization techniques. Certain of these had previously been called "the second variation method" although the precise meaning of this term was formerly unclear. The method of Chapter 5 also belongs to this class of algorithms. The recognition of a class of second variation methods unifies several seemingly diverse techniques for solving the optimal programming problem as well as providing the means for developing new techniques.

2. CONTROL LAW APPROXIMATION

2-1 Control Synthesis Based on Optimal Trajectories.

Bellman's method of dynamic programming [5] is the only general procedure for computing the optimal feedback control directly. The result of performing the dynamic programming calculation would be a multidimensional numerical map giving the required value of optimal control at all points of a discrete grid in some relevant region of state-time space. As a numerical technique, discrete dynamic programming has some severe limitations when applied to the optimal control of continuous dynamic systems. Quite apart from the errors introduced by truncation and quantization, the main factor limiting its applicability is the size of the computer memory required and the rate of increase of the required storage with system dimension ([6], pp. 21-23).

Considerably more success has been experienced in solving the optimal programming problem. From a computational point of view, a solution of the optimal programming problem requires the solution of a two-point boundary value problem, which is generally nonlinear. Many techniques now exist for solving this difficult problem and the work in Chapters 5 and 6 of this thesis constitutes a further contribution towards achieving a rapid and efficient means of solution. Although more development and improvement of these numerical techniques remains to be done, it is felt

that they are now sufficiently well developed as to make it highly desirable that the control law synthesis procedure be based on optimal open-loop controls and trajectories rather than on dynamic programming calculations.

The optimal programming solution (Eqs. (1.4) and (1.5)) may be thought of as a parametric representation of the optimal feedback control law (1.6). What is required, then, is an algorithm to convert the parametric form to the closed-loop or present-state form. Such a procedure must necessarily be approximate in all but the simplest cases.

2-2 Reformulation for Time-Invariant Feedback.

Conceptual simplifications result if the explicit time dependence of the feedback control law is removed. For this purpose, the control problem dealt with in Part I has the following special form:

Special problem formulation

$$\text{Dynamics} \qquad \dot{x} = f(x,u) \quad x(t_o) = x_o \qquad (2.1)$$

$$\text{Terminal conditions} \qquad \Psi(x(t_f)) = 0 \qquad (2.2)$$

$$\text{Performance functional} \qquad J(u) = \int_{t_o}^{t_f} F(x,u)\,dt \qquad (2.3)$$

The terminal time t_f is determined implicitly by (2.2) and the bounds on u, if any, are considered to be independent of time. It is assumed that the state is constrained only by (2.1)

and (2.2). For this special problem, the initial time t_o has no explicit influence on the optimal control which can thus only depend on the current state, that is, it may be written as $u(x)$.

The more general formulation in Section 1-1 can be reduced to the stationary form above. Equation (1.3) is equivalent to

$$J(u) = \int_{t_o}^{t_f} (F(x,u,t) + \phi_x(x,t)f(x,u,t) + \phi_t(x,t))dt \tag{2.4}$$

The time dependence may be formally replaced by introducing the extra state variable x_{n+1} where

$$\dot{x}_{n+1} = 1 \qquad x_{n+1}(t_o) = t_o \tag{2.5}$$

$$x_{n+1}(t_f) = 1 \tag{2.6}$$

Then $x_{n+1}(t)$ is substituted for t in (1.1) and (2.4) and $x_{n+1}(t_f)$ for t_f in (1.2) which results in equations of the same form as (2.1), (2.2), and (2.3).

2-3 Development of the Approximation Scheme [9].

The optimal control law $u(x)$ is to be approximated over a specified region B of state space by a function $v(x;c)$, where c is an N-vector of adjustable parameters. To

avoid unnecessary complications in notation, the control vector u will be considered as a scalar. The form of the suboptimal control $v(x;c)$ must be prespecified. The parameters c are to be determined so that, in some sense, v is the best control law amongst all other controllers having the same structure. Controllers which are "optimal" subject to a prespecified input-output relation have been called "specific optimal controllers" [10].

A great deal of latitude exists in the choice of criteria for determining the adjustable parameters of the suboptimal controller. For example, the parameters could be chosen to minimize the performance value $J(v;x_o,c)$ obtained for some nominal initial condition, or to minimize some functional of the deviation between the optimal trajectory and the quasi-optimal one. Alternatively, a functional of the deviation between the optimal open-loop control and the control signal

$$v_c(t) = v(x_v(t;x_o);c) \tag{2.7}$$

could be minimized.

The criterion advocated is closely related to this last suggestion. To begin with, the quasi-optimal controller is restricted to have the form

$$v(x;c) = \sum_{j=1}^{N} c_j Z_j(x) \tag{2.8}$$

where the functions $Z_j(x)$ are termed basis functions. Let

$u(t;x_{ok})$ and $x(t;x_{ok})$, $0 \le t \le t_{fk}$, be respectively the optimal control and optimal trajectory from the initial point x_{ok}. The subscript k distinguishes different initial states; t_{fk} is the terminal time determined by (2.2). The parameter vector c is chosen to minimize

$$E(c) = \sum_{k=1}^{M} \int_{0}^{t_{fk}} (u(t;x_{ok}) - v(x(t;x_{ok});c))^2 dt \tag{2.9}$$

Note that in (2.9), the suboptimal control $v(x;c)$ is evaluated not along the trajectory which results from using control v as in (2.7) but along the optimal trajectory (produced by using u).

The M optimal trajectories $\{x(t;x_{ok})\}$ are referred to as development trajectories. More will be said about the location of the initial states x_{ok} in Section 2-3.2 but, in some sense, the development trajectories "cover" the region of operation B. Thus, the minimization of E(c) in (2.9) results in an approximation $v(x;c)$ to $u(x)$ which gives a least sum of mean-square deviations over several system trajectories.

If (2.8) is substituted into (2.9), the resulting integral can be written as

$$E(c) = c^T A c - 2b^T c + r \tag{2.10}$$

where

$$A_{ij} = \sum_{k=1}^{M} \int_{0}^{t_{fk}} Z_i(x(t;x_{ok}))Z_j(x(t;x_{ok}))dt \tag{2.11}$$

$$b_i = \sum_{k=1}^{M} \int_0^{t_{fk}} Z_i(x(t;x_{ok}))u(t;x_{ok})dt \qquad (2.12)$$

$$r = \sum_{k=1}^{M} \int_0^{t_{fk}} u^2(t;x_{ok})dt \qquad (2.13)$$

$$(i,j = 1,\ldots,N)$$

If A is positive definite, a unique minimizing c exists and is given by the solution of

$$Ac = b \qquad (2.14)$$

The minimum value of E is

$$E_{min} = r - b^T A^{-1} b \qquad (2.15)$$

Conditions on the basis functions which ensure that A is positive definite are given in Section 2-3.1.

A very important feature of the proposed method is the ease of computation. In general, the other possible criteria previously referred to lead to nonlinear minimization problems. Subsequent discussion wil indicate the practical necessity of having to experiment with different choices of basis functions. The attendant complexity and tedium of nonlinear minimization methods would thus greatly inhibit the overall synthesis procedure.

2-3.1 Conditions for a Positive Definite A.

Consider the integral

$$I(c) = \sum_{k=1}^{M} \int_{0}^{t_{fk}} v^2(x(t;x_{ok});c)dt \geq 0 \qquad (2.16)$$

If (2.8) is substituted into (2.16), it is easily shown that

$$I(c) = c^T A c \qquad (2.17)$$

Thus, from (2.16) and (2.17), A is positive definite provided the basis functions Z_j are linearly independent along at least one of the development trajectories. It is most unlikely that this condition would not be met in practice.

2-3.2 The Choice of Development Trajectories.

Little can be said about the "best" choice of development trajectories. In essence, a function defined over a multidimensional region B is being approximated by another function which is to be "close" along specific curves in B. This approach is analogous to a common technique used in approximating functions of a single variable over an interval of the real line where the approximation is carried out for a finite point subset of this interval. Even in this much simpler problem, the determination of an optimum number and spacing of points represents an exceedingly difficult computational task. In both cases, however, the hope is that in approximating on a subset of the desired region, the resulting approximation is adequate over the entire

region or interval.

In many problems, there exist preferred regions for disturbed states, that is, regions of probable system initial conditions. Sample or typical initial conditions within these preferred regions are a natural choice for development trajectory initial conditions. Otherwise, in the absence of any a-priori information, the most obvious choice is a geometrically uniform distribution of initial conditions around the boundary of the approximation region.

2-3.3 Computational Considerations.

Numerical computation of the best c is simple and convenient. As each new development trajectory is computed or is retrieved from storage, the basis functions are evaluated, multiplied and integrated according to Eqs. (2.11) to (2.13). The results are then added on to the previously accumulated coefficient matrix and the next development trajectory is generated. When all M development trajectories have been processed in this manner, the linear set of equations (2.14) is solved. The ratio E_{min}/r serves as a figure of merit in comparing the suitability of one set of basis functions with another.

It is remarked that the choice of integrating the squared difference in (2.9) rather than summing has the effect of making efficient use of trajectory data. The optimal trajectory is the solution of a differential equation and hence, is available in a numerical form which is

completely compatible with the quadrature operations in (2.11) - (2.13).

2-4 Example 1.

To illustrate the application of the synthesis technique, an example control problem for a nonlinear system with two state variables is presented. This example will also demonstrate how suitable basis functions may be chosen and how the number and distribution of development trajectories influences the resulting approximation.

It is desired to find a suboptimal feedback control which suitably approximates the optimal control for the problem of minimizing

$$J(u) = \tfrac{1}{2}\int_0^{t_f} (x_1^2 + x_2^2 + u^2)dt \tag{2.18}$$

where

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= (1-x_1^2)x_2 - x_1 + u\end{aligned} \tag{2.19}$$

and where t_f is determined as the first instant at which

$$\Omega(x(t_f)) = \tfrac{1}{2}(x_1^2(t_f) + x_2^2(t_f) - (0.1)^2) \tag{2.20}$$

is zero. The expected region of operation B in state space was taken to be the square $x_1 \epsilon (-2,2), x_2 \epsilon (-2,2)$.

Applying the minimum principle [4], the optimal programmed control from an initial state x_o is determined by

$$\frac{\partial H}{\partial u} = u + p_2 = 0 \tag{2.21}$$

where $H = \frac{1}{2}(x_1^2 + x_2^2 + u^2) + p_1 x_2 + p_2((1-x_1^2)x_2 - x_1 + u)$ (2.22)

and $$\dot{p}_1 = -H_{x_1} = -x_1 + (1+2x_1x_2)p_2 \tag{2.23}$$

$$\dot{p}_2 = -H_{x_2} = -x_2 - p_1 + (1-x_1^2)p_2$$

The terminal condition on the adjoint vector p is given by

$$p(t_f) = \mu \Omega_x^T = \mu x(t_f) \tag{2.24}$$

where μ is an undetermined scalar parameter. A further necessary condition for this free terminal-time problem is given by

$$H = 0 \tag{2.25}$$

along an optimal trajectory. From Eqs. (2.22), (2.24), (2.25), it can be deduced that μ is the positive root of the following quadratic equation:

$$-\tfrac{1}{2} x_2^2 \mu^2 + x_2^2(1 - x_1^2)\mu + \tfrac{1}{2}(x_1^2 + x_2^2) = 0 \tag{2.26}$$

where (x_1, x_2) is any optimal trajectory terminal point.

Optimal trajectories may be generated by choosing a point $x(t_f)$ on $\Omega = 0$, determining μ from (2.26), $p(t_f)$ from (2.24) and integrating the canonical system (state and adjoint equations) (2.19) and (2.23) in reverse time until the trajectory leaves B. Although one has no idea in

selecting a terminal point where the trajectory will end up, it was possible in this problem to generate a sufficient number of trajectories in this manner so that all parts of the operating region are covered. From the many trajectories generated, ten were selected as candidate development trajectories (numbered 1-10 in Fig. 2-1) and seven trajectories were chosen as test trajectories (lettered A-G in Fig. 2-1). These test trajectories, whose performance values are listed in Table 2.1 ,

Table 2.1 Optimal Performance Values for Test Trajectories.

	Test Trajectory. (see Fig. 2-1)						
	A	B	C	D	E	F	G
Optimal Perf J(u)	7.885	7.161	10.171	8.601	9.224	9.761	9.113

will be used to compare the performance of the optimal and suboptimal systems. Note that from the symmetry of Eqs. (2.18)-(2.20), $u(-x) = -u(x)$ so that only half of the approximating region B need be considered.

From the many trajectories that were generated together with crude interpolations, the general form of the optimal feedback surface u(x) was plotted (Fig. 2-2). While it may be possible to obtain a satisfactory approximation to u(x) by a curve-fitting procedure using Fig. 2-2 ([11], pp. 94-117), the knowledge of u(x) provided by Fig. 2-2 is here used only to suggest a suitable form for the suboptimal control. If a coordinate system (z,y) is introduced by a

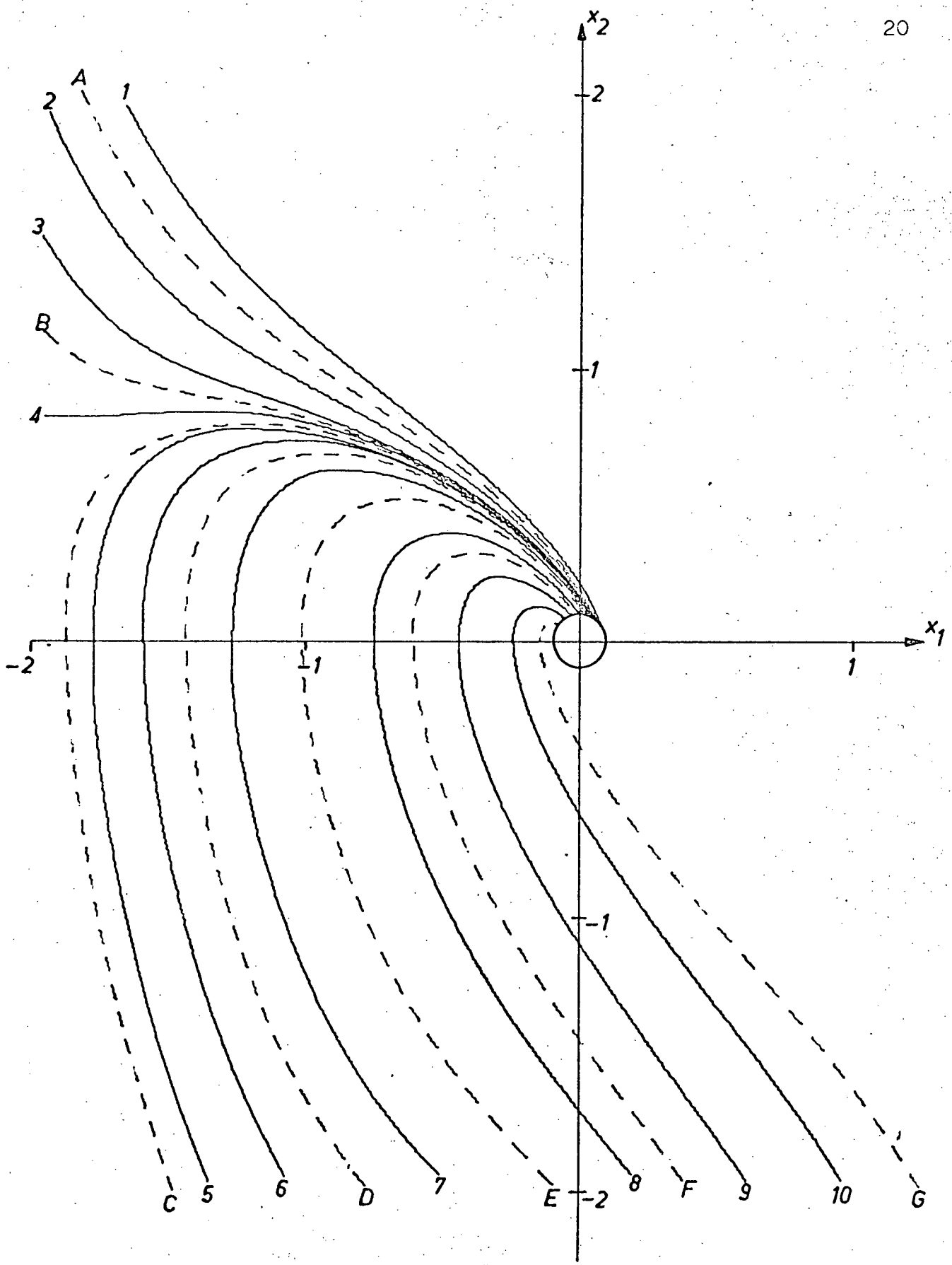

Fig. 2-1 Optimal Development (solid) and Test (broken) Trajectories.

rotation of the (u,x_1) system through an angle θ, a reasonable approximation of the curves in Fig. 2-2 is given by

$$z = -ax_2y^2 - bx_2 \tag{2.27}$$

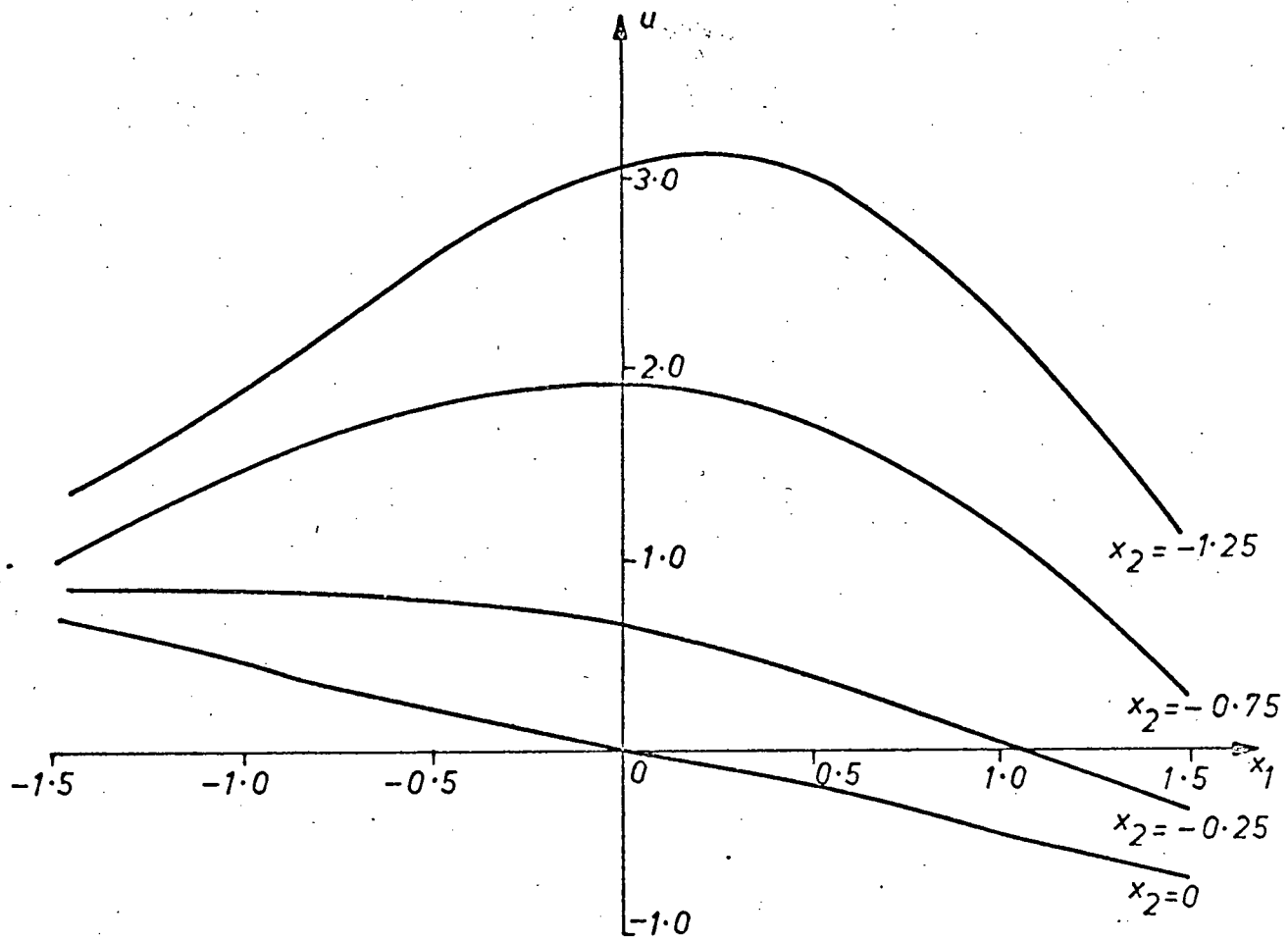

Fig. 2-2 Optimal Feedback Surface.

where

$$z = u \cos \theta + x_1 \sin \theta \tag{2.28}$$
$$y = x_1 \cos \theta - u \sin \theta$$

Substituting (2.28) into (2.27), solving for u, and then

expanding in a Taylor series neglecting terms of order five or more, the following controller structure is obtained:

$$v(x;c) = c_1x_1 + c_2x_2 + c_3x_1^2x_2 + c_4x_1x_2^2 + c_5x_2^3 \qquad (2.29)$$

The $c_i, i = 1,\ldots,5$ are functions of the parameters a,b,θ which will remain unspecified.

Employing one development trajectory only, the five controller parameters in (2.29) were evaluated according to Eqs. (2.11)-(2.14) with M = 1. Having obtained a set of parameter values, the suboptimal control (2.29) was evaluated by integrating the system (2.19) under suboptimal control from each of the seven test initial conditions. This procedure was repeated for four different development trajectories and the results are summarized in Table 2.2. The worst

Table 2.2 Controller Parameters and Worst Case Performance, Single Development Trajectory.

Trial	Development Trajectory (Fig. 2-1)	Controller Parameters c_1 c_2	c_3 c_4	c_5	Performance Deterioration Worst Case Test Traj.	% Increase
1	3	-0.4914 -2.4047	-0.8879 -4.2055	-3.3008	C	329
2	5	-0.4124 -2.6728	0.6698 0.1054	0.1641	A	5.3
3	8	-0.4137 -2.6171	0.8749 0.5096	0.1204	C	45
4	10	-0.4121 -2.7212	1.2872 0.8470	0.2151	C,D	∞

case performance increase listed in this table is obtained by evaluating

$$\max_k \frac{J_k(v) - J_k(u)}{J_k(u)}$$

where $J_k(v)$ is the suboptimal performance value for the k^{th} test trajectory and $J_k(u)$ is the optimal performance value((see Table 2.1)). The optimal and suboptimal trajectories for the seven test points are shown in Figs. 2-3 and 2-4 for trials 1 and 2. In trial 4 (development trajectory 10), the suboptimal system did not reach the terminal manifold from two test states. That the best results were obtained using the "middle trajectory" (#5) is not altogether unreasonable from an intuitive point of view.

The results where several development trajectories were employed are displayed in Table 2.3. In these eight trials, the suboptimal control was a general polynomial of the fourth-order but with the condition $u(-x) = -u(x)$ enforced:

$$v(x;c) = c_1x_1 + c_2x_2 + c_3x_1^3 + c_4x_1^2x_2 + c_5x_1x_2^2 + c_6x_2^3 \tag{2.30}$$

Note that (2.30) has one more term than (2.29). The optimal and suboptimal test trajectories for trials 1 and 8 are displayed in Figs. 2-5 and 2-6.

Several important observations can be made from the data in Table 2.3. First, it is noticed that the

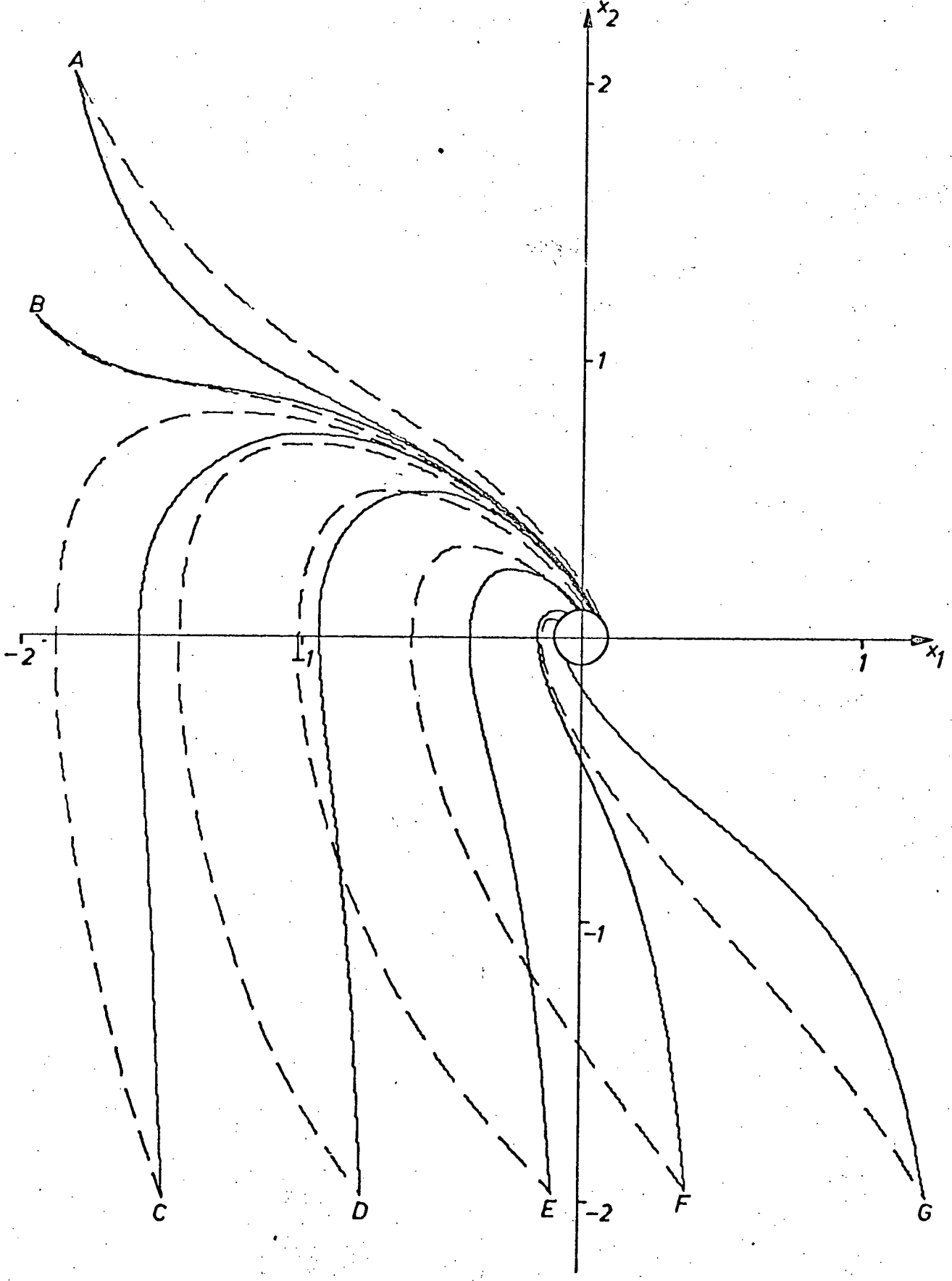

Fig. 2-3 Optimal (broken) and Suboptimal (solid) Test Trajectories for Trial 1, Single Development Trajectory.

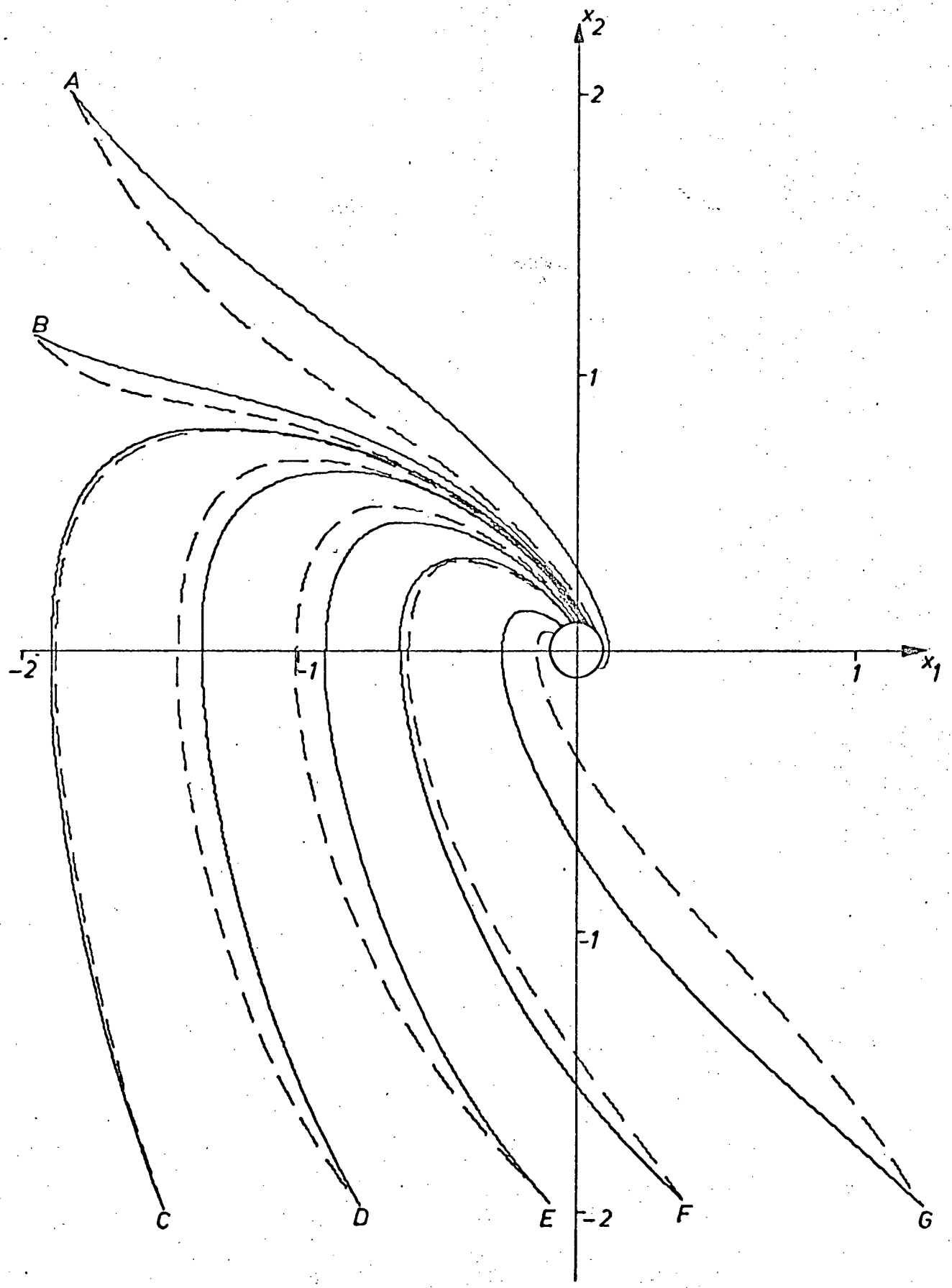

Fig. 2-4 Optimal (broken) and Suboptimal (solid) Test Trajectories for Trial 2, Single Development Trajectory.

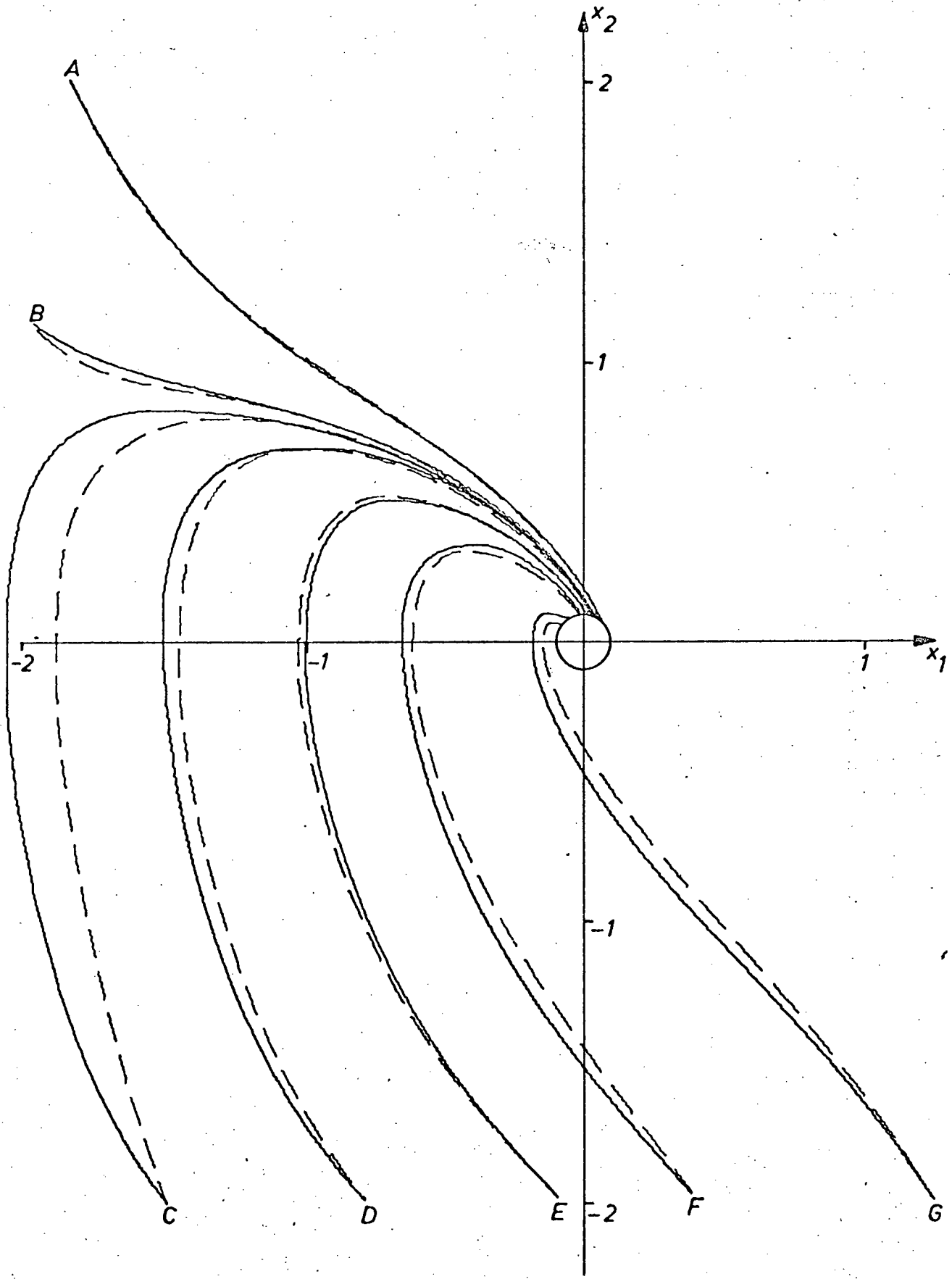

Fig. 2-5 Optimal (broken) and Suboptimal (solid) Test Trajectories for Trial 1, Multiple Development Trajectories.

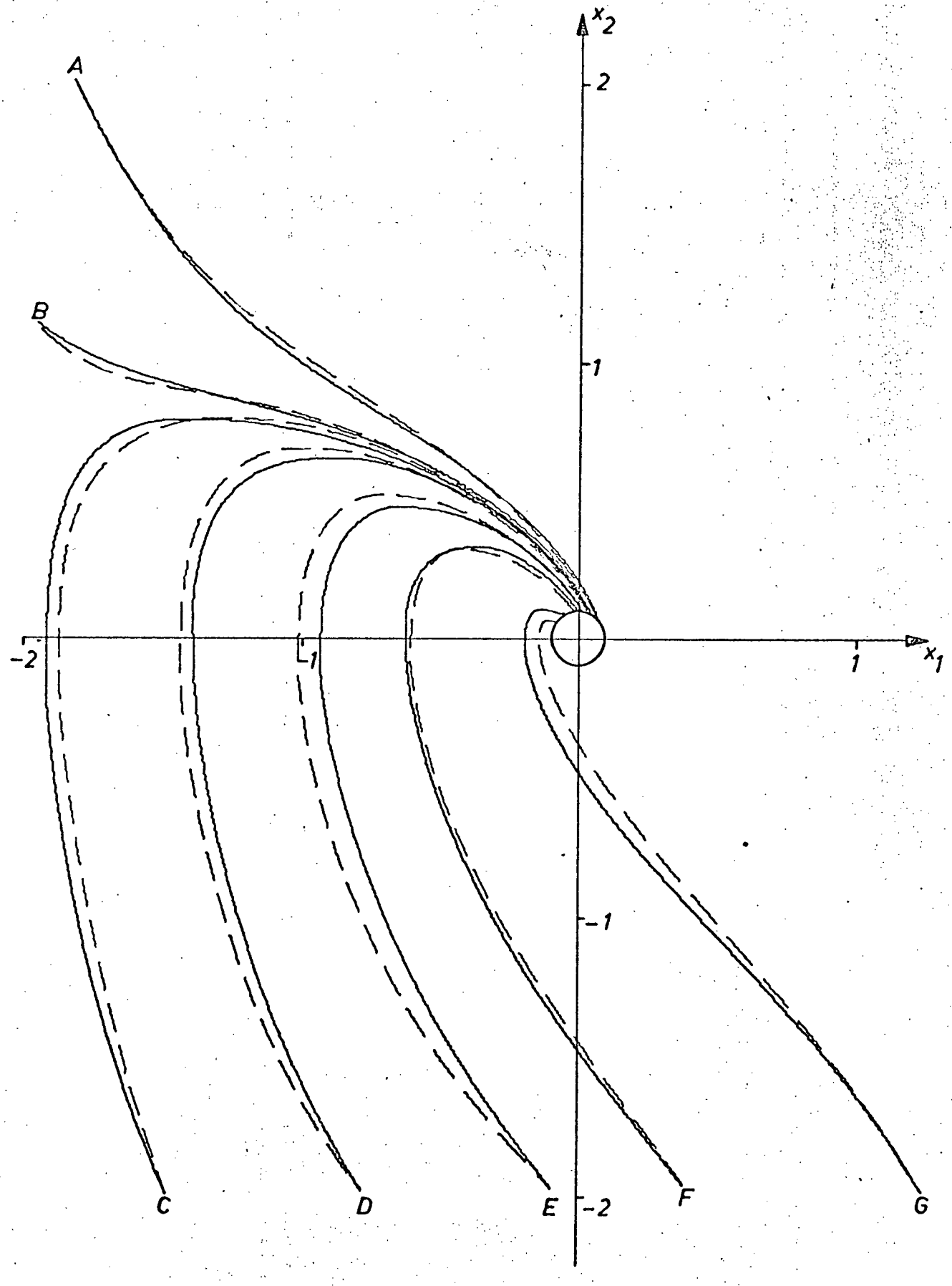

Fig. 2-6 Optimal (broken) and Suboptimal (solid) Test Trajectories for Trial 8, Multiple Development Trajectories.

Table 2.3 Controller Parameters and Worst Case Performance, Multiple Development Trajectories.

Trial	Development Trajectory (Fig. 2-1)	Controller Parameters			Performance Deterioration Worst Case	
		c_1 c_2	c_3 c_4	c_5 c_6	Test Traj.	% Increase
1	1,7	-0.4560 -2.6418	0.0101 0.7931	0.4349 0.1601	C	21
2	5,10	-0.5214 -2.6757	0.0233 0.6951	0.2440 0.0382	D	3.0
3	8,10	-0.4394 -2.6906	0.0267 1.0084	0.5896 0.1439	C	355
4	1,5,9	-0.4181 -2.6105	-0.0090 0.6809	0.2148 0.0359	D	2.5
5	3,7,10	-0.5232 -2.6828	0.0344 0.8600	0.5135 0.1318	C	39
6	1,3,5,7,9	-0.3118 -2.4926	-0.0644 0.6287	0.2872 0.0834	C	1.5
7	2,4,6,8,10	-0.4121 -2.5705	-0.0133 0.7005	0.3200 0.0779	C	3.3
8	1,2,3,4,5, 6,7,8,9,10	-0.3577 -2.5294	-0.0437 0.6578	0.3021 0.0805	C	2

parameter c_3 is comparatively small in all trials which confirms the suitability of the form (2.29) previously used. It further appears, from this data, that one can feel more confident of obtaining a "globally" valid approximation when more development trajectories are employed and that geometrically uniform distributions of trajectories produce better results than nonuniform distributions (compare trials 2 and 3). Comparing the controller parameters between trials, the ranges of values taken by the

parameters in the more successful trials (trials 2,4,6,7,8) indicates a relatively low sensitivity of performance to the suboptimal controller's parameters. Note that on the less successful trials (trials 1,3,5), the values for parameters c_4, c_5, c_6 are definitely outside the range of values taken by these parameters on the successful trials.

2-5 Example 2.

The second example belongs to the following problem class:

$$\dot{x} = Fx + Gu + f(x) + g(x)u \tag{2.31}$$

$$J(u) = \tfrac{1}{2} \int_{t}^{t+t_f} (x^T Q x + u^T R u)\,dt \tag{2.32}$$

$$\Omega(x(t+t_f)) = \tfrac{1}{2}(x^T K x - \varepsilon^2)\Big|_{t+t_f} = 0 \tag{2.33}$$

In the dynamical equations (2.31), F is a constant nxn matrix, G is a constant nxl matrix (assuming r = 1), f(x) and g(x) are continuous n-vector functions of state

satisfying

$$f(0) = g(0) = 0 \tag{2.34}$$

The matrix Q in the performance functional (2.32) is positive semidefinite and since it is assumed that dim(u) = 1, R may be taken as unity without loss of generality. Equation (2.33) defines the terminal time (stopping function) and K is the positive definite solution of the steady-state matrix Ricatti equation:

$$KGR^{-1}G^{T}K - KF - F^{T}K - Q = 0 \tag{2.35}$$

Since K is positive definite, the stopping function (2.33) represents a hyperellispsoid about the origin.

The qualitative purpose of the control system is to return the system to the origin (or the neighborhood inside the ellipsoid) following a disturbance in such a way that those state variable weighted in the performance integrand do not make large excursions and with limited use of control energy. The nonlinear functions f and g in the dynamics could represent higher-order terms of significance in the expansion of the original system equations. If f and g were both identically zero, the optimal feedback control would be given by

$$u_{R}(x) = -G^{T}Kx \tag{2.36}$$

that is, a linear combination of the state variables [34].

The control $u_R(x)$ will be referred to as the Ricatti control. Thus, at least in a neighborhood of the origin where f and g are small, the structure of u is known. It should be noted that the linear structure (2.36) applies only if K satisfies (2.35). If f and g are smooth functions of x, it is reasonable to expect that the linear function (2.36) will change in a smooth manner as x gets further from the origin. Hence, a polynomial structure for v(x;c) is likely to be a good approximation.

The specific example treated in this section is defined by Eqs. (2.31) - (2.33) and

$$F = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad f(x) = \begin{bmatrix} 0 \\ 0 \\ -x_1^3 \end{bmatrix}, \quad g(x) = 0,$$

$$Q = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad \begin{matrix} R = 1 \\ \varepsilon = 0.3 \end{matrix} \tag{2.37}$$

From Eq. (2.35), the steady-state Ricatti matrix is

$$K = \begin{bmatrix} 7 & 5 & 1 \\ 5 & 10 & 3 \\ 1 & 3 & 2 \end{bmatrix} \tag{2.38}$$

and the Ricatti control is

$$u_R(x) = (-1,-3,-2)x \tag{2.39}$$

An approximation region was chosen arbitrarily as the region B interior to the planes

$$\text{B: Interior of } \begin{cases} x_3 = \pm 2 \\ x_1 + x_2 = \pm 2 \\ x_1 - x_2 = \pm 2 \end{cases} \tag{2.40}$$

Based on the reasoning in the preceding paragraph, a fourth-order polynomial was chosen for $v(x;c)$. Since the symmetry condition $u(-x) = -u(x)$, applicable in the previous example, applies here as well, the suboptimal control may be written

$$v(x;c) = \sum_{i=1}^{3} x_i (c_i + x_i \sum_{j=1}^{3} c_{3i+j} x_j) + c_{13} x_1 x_2 x_3 \tag{2.41}$$

which is of the form (2.8) where

$$Z_m(x) = \begin{cases} x_m, & m = 1,2,3 \\ x_j^2 x_k, & m = 3j + k,\ j,k = 1,2,3 \\ x_1 x_2 x_3, & m = 13 \end{cases} \tag{2.42}$$

Initially, 7 development trajectory initial conditions (Group 1 in Table 2.4) were selected around the

Table 2.4 Development Trajectories Initial Conditions.

	Group 1 (7 Trajectories)							Group 2 (13 Trajectories) All of Group 1 plus:					
x_1	2	0	-2	0	2	0	0	1	1	-1	-1	1	1
x_2	0	2	0	-2	0	2	0	1	-1	1	-1	1	-1
x_3	2	2	2	2	0	0	2	2	2	2	2	0	0

boundary of B. The controller parameters computed by the procedure of this chapter are listed as Parameter Set 1 in Table 2.5. Then another trial was made with 13 develop-

Table 2.5 Controller Parameters, All Third-Order Terms.

	c_1 c_2 c_3	c_4 c_5 c_6	c_7 c_8 c_9	c_{10} c_{11} c_{12}	c_{13}
Parameter Set 1	-0.9427	0.1562	-0.7328	-0.0384	-0.3208
	-2.9854	-0.6032	-0.2583	-0.0476	
$E_{min}/r = 3.9\ 10^{-5}$	-2.0039	-0.1392	-0.1876	-0.0101	
Parameter Set 2	-0.9236	0.1498	-0.7560	-0.0320	-0.3637
	-2.9645	-0.6412	-0.2615	-0.0556	
$E_{min}/r = 7.3\ 10^{-5}$	-2.0013	-0.1454	-0.1956	-0.0047	

ment trajectories (Group 2 in Table 2.4) and the corresponding controller parameters are given as Parameter Set 2 in Table 2.5. The development trajectories and all other optimal trajectories used in this example were computed by employing the extended Newton-Raphson method of Chapter 5.

From the data in Table 2.5, it can be seen that the controller parameters did not change significantly between the two trials. In effect, this means that approximating to the Group 1 trajectories has produced a hypersurface which is not significantly altered by asking it to be "close" to other trajectories in the range covered by the first

group. Although the ratio E_{min}/r is listed for each trial in Table 2.5, it is not significant in comparing the two cases since different development trajectories have been used. It is interesting to observe the closeness of the data (c_1, c_2, c_3) to the coefficients in the Ricatti control, Eq. (2.39).

If it is desired to reduce the number of basis functions but still retain the polynomial structure of Eq. (2.41), a simple procedure exists for trying subsets of the original set of basis functions. Eliminating a particular basis function Z_m from the suboptimal control merely involves deleting the m^{th} row and m^{th} column from the augmented matrix $[A:b]$ of Eqs. (2.11) - (2.12). Thus, any number of basis functions can be eliminated in this way and the resulting reduced system of linear equations is then solved for the reduced parameter set. A comparison can be made between various subsets on the basis of the ratio E_{min}/r.

For example, suppose it is desired to have only 5 third-order terms in Eq. (2.41) instead of the full 10. There are 252 possible 5-element subsets of $\{Z_i, i = 4,\ldots,13\}$. For each subset, the corresponding 8 x 8 linear subsystem of the augmented matrix for the Group 2 trajectories was solved and E_{min}/r evaluated. The smallest figure of merit resulted when basis functions 6,9,10,11, and 12 were omitted. The controller parameters and E_{min}/r are given in Table 2.6. This procedure was repeated for 7 third-order terms (120 possible subsets) and the parameter values for the best subset are also listed in Table 2.6. It is interesting to observe from

Table 2.6 Controller Parameters, Constraint on Number of Third-Order Terms.

	c_1 c_2 c_3	c_4 c_5 c_6	c_7 c_8 c_9	c_{10} c_{11} c_{12}	c_{13}
Best with 5	-1.3142	0.2793	-0.5800	0	-0.2895
3rd-order terms	-3.3035	-0.5430	-0.1460	0	
$E_{min}/r = 1.96 \times 10^{-3}$	-2.3249	0	0	0	
Best with 7	-0.9576	0.1526	-0.7530	0	-0.2735
3rd-order terms	-3.0143	-0.6144	-0.2602	0	
$E_{min}/r = 2.90 \times 10^{-4}$	-2.0047	-0.1201	-0.1688	0	

Table 2.5 and Table 2.6 that with the single exception of c_9 in the first case, the parameters omitted were the smallest. Although a great number of possible subsets have to be tried, the procedure is not time consuming. The computer (IBM 7044) time required for the 5-element subsets was 38.5 seconds and for the 7-element subsets, 24.5 seconds.

The suboptimal control with the controller parameters in Table 2.5 was tested over a range of initial conditions within the approximation region. These initial conditions were chosen as worst case tests in that they were maximally distant from development trajectory initial conditions. The increase in performance values over optimal

was negligible in every case. In order to observe deviations from optimal performance, it was necessary to use test initial conditions outside B. The values of terminal time and performance for some cases of interest are listed in Table 2.7. Terminal times for the suboptimal trajectories listed in Table 2.7 are accurate to within $\pm$ 0.025 (the integration step size). In addition to the suboptimal feedback controls specified by Eq. (2.41) and the parameter sets in Table 2.5, the results of using optimal control, the linear Ricatti control and a control based on an expansion of the Hamilton-Jacobi equation are shown in the Table. This latter control is of exactly the same form as (2.40) but with different parameter values (see Appendix A). The response $x_1(t)$ to the optimal control, nonlinear suboptimal controls and the Ricatti linear control are shown in Fig. 2-7 for the initial condition (2,2,2).

Table 2.7 Evaluation of Various Suboptimal Feedback Controls.

Int. Condition			Performance J Terminal time t_f				
x_1	x_2	x_3	Optimal	Parameter Set 1	Parameter Set 2	Merriam's Method (AppA)	Linear Ricatti
2	-2	2	13.1 2.83	13.1 2.95	13.1 2.80	13.1 2.79	13.7 3.50
2	1.5	1.5	144.4 5.12	144.9 5.10	144.9 5.10	158.1 5.31	160.4 6.50
2	2	0	175.9 5.15	176.9 5.15	176.8 5.15	191.7 5.34	199.5 6.70
2	2	1	196.7 5.20	197.9 5.20	197.8 5.20	216.1 5.40	230.1 6.85
2	2	2	221.6 5.25	223.0 5.25	222.9 5.25	244.6 5.46	269.2 7.00
2	2	3	250.6 5.31	252.4 5.30	252.3 5.30	277.3 5.55	318.6 7.15
3	3	3	1003.2 6.31	1036.6 6.15	1027.3 6.10	1146.0 6.33	(Unstable)

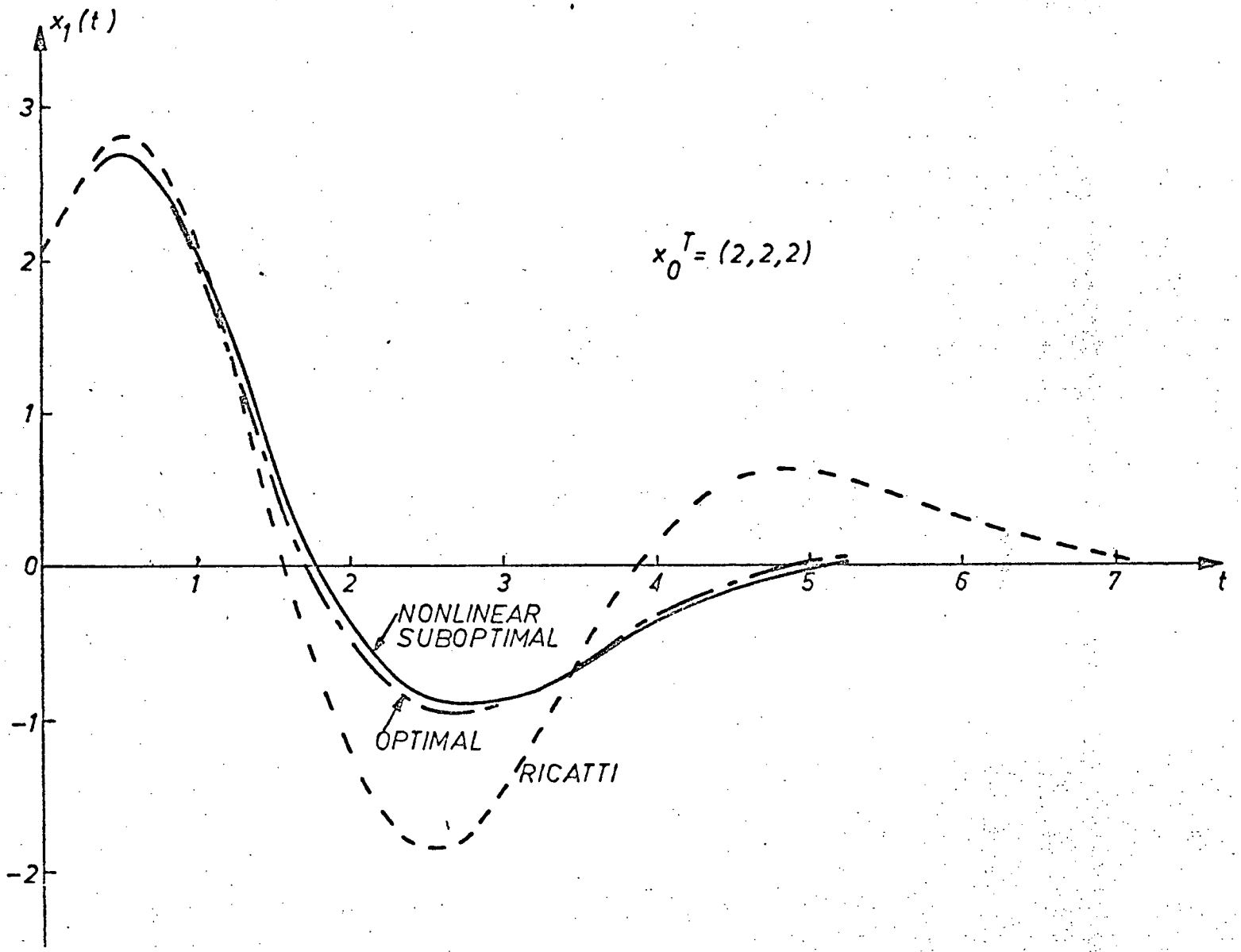

Fig. 2-7 $x_1(t)$ for the Optimal, Nonlinear Suboptimal (Eq. (2.40)) and Linear Ricatti Controls.

3. PIECEWISE POLYNOMIAL APPROXIMATION

3-1 Polynomial Basis Functions.

Successful application of the synthesis technique given in Chapter 2 depends upon the designer's ability to specify a suitable controller structure (suitable basis functions if $v(x;c)$ is to be linear in c). There are two basic approaches to this question. The first involves making a detailed study of a particular problem or problem class in an attempt to obtain clues about the structure of $u(x)$. While this study could be completely empirical, it should capitalize on any existing theoretical knowledge of the solution and be aided by whatever analysis is feasible. Examples of this approach are contained in Sections 2-4 and 2-5. In the second approach, the basis functions employed are of general utility, that is, they are ones that will adequately serve for a large class of problems. The most common example of such a general utility set of basis functions is the polynomials.

If high-order polynomials are required to obtain an adequate approximation, several difficulties will be encountered. In computing c from Eq. (2.14), there is reason to believe that the matrix A will become ill-conditioned as the order of the polynomial increases. The existence of this phenomenon is supported by the author's computational experience and is well-known in single-variable least-squares

approximation theory ([12], Section 17.5). High-order polynomial approximations also exhibit numerical instability ([13], p. 296). This property is closely associated with an even more serious condition which may arise, namely unwanted fluctuations of $v(x;c)$ in regions between development trajectories.

3-2 Piecewise Polynomial Functions.

The above objections to high-order polynomial approximation may be largely overcome through the use of another class of general utility basis functions, the piecewise polynomial functions. A function v of this class is a polynomial on each of several disjoint regions B_m, that is, if there are N_r such regions and $\bigcup_{m=1}^{N_r} B_m = B$, then

$$v(x;c) = \sum_{m=1}^{N_r} P_m(x;c^m)K_m(x) \tag{3.1}$$

on B, where $P_m(x;c^m)$ is a polynomial in x,

$$c = \begin{bmatrix} c^1 \\ \cdot \\ \cdot \\ \cdot \\ c^{N_r} \end{bmatrix} \tag{3.2}$$

and

$$K_m(x) = \begin{cases} 1, & x \epsilon B_m \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

The vector c^m has N_m components where

$$\sum_{m=1}^{N_r} N_m = N \qquad (3.4)$$

It should be observed that if the regions B_m are pre-specified, (3.1) is a linear approximating function of the form (2.8). The optimal choice of subregions will not be considered in this thesis.

One of the most powerful arguments in favor of polynomial approximation is provided by the Weierstrass Approximation Theorem ([14], Sect. 6.6) which states that any real continuous function can be approximated arbitrarily closely on any closed and bounded set by polynomials of sufficiently high degree. For a given approximation accuracy, it is evident that a polynomial of lower degree will suffice if the approximation region is reduced in size. Thus, by using low-order polynomials in each of several subregions, the accuracy of high-order polynomial approximation is maintained while the undesirable properties attributable to high order are eliminated.

Interpolation and approximation by spline functions of a single variable has received considerable attention ([15], for example). Spline functions are a subclass of piecewise polynomial functions. For scalar x, v(x;c) in Eq. (3.1) is a spline function of degree k if it is equal to a polynomial of k^{th} degree on each subinterval B_i of the interval B and the parameters c are such that on B, v is continuous and has continuous derivatives up to order k-1.

Further discussions on the advantages of spline approximation may be found in [16] p. 17, and [17].

3-3 Discontinuous Suboptimal Control Law.

The polynomials $P_m(x;c^m)$ in Eq. (3.1) have the form

$$P_m(x;c^m) = \sum_{r=1}^{N_m} c_r^m \, Y_r^m(x) \tag{3.5}$$

where $Y_r^m(x)$ is a product of powers of the state variables. From Eqs. (2.8), (3.1), and (3.5), the basis functions $Z_i(x)$ are given by

$$Z_i(x) = Y_r^m(x) K_m(x) \tag{3.6}$$

for $r = 1,\ldots,N_m, \quad m = 1,\ldots,N_r$

where $i = i(r,m) = N_1 + N_2 + \ldots + N_{m-1} + r$ (3.7)

With no restrictions on the parameters c, the optimal c can be obtained directly from Eq. (2.14). If $i = i(1,m_1)$ and $j = j(r,m_2)$, then from Eqs. (2.11) and (3.6),

$$A_{ij} = \sum_{k=1}^{M} \int_0^{t_{fk}} (Y_1^{m_1} Y_r^{m_2} K_{m_1} K_{m_2}) dt \tag{3.8}$$

where the argument of each factor in the integrand is $x(t;x_{ok})$.

But along any trajectory,

$$K_{m_1}(t)K_{m_2}(t) = \begin{cases} K_{m_1}(t), & m_1 = m_2 \\ 0, & m_1 \neq m_2 \end{cases} \tag{3.9}$$

Hence, $A_{ij} = 0$ if the indices i and j correspond to different subregions of B, thus permitting the following decomposition of the linear system (2.14):

$$\begin{bmatrix} A^1 & 0 & 0 & & \\ 0 & A^2 & 0 & & \\ 0 & 0 & \cdot & & \\ & & & \cdot & \\ & & & & \begin{matrix} 0 \\ N_r \end{matrix} \\ & & & \begin{matrix} 0 & A^{N_r} \end{matrix} \end{bmatrix} \begin{bmatrix} c^1 \\ c^2 \\ \cdot \\ \cdot \\ \cdot \\ c^{N_r} \end{bmatrix} = \begin{bmatrix} b^1 \\ b^2 \\ \cdot \\ \cdot \\ \cdot \\ b^{N_r} \end{bmatrix} \tag{3.10}$$

The vector of parameters c^m corresponding to subregion B_m is the solution of

$$A^m c^m = b^m \tag{3.11}$$

It is probable that discontinuous piecewise polynomial basis functions might only be used in an initial investigation into the nature of u(x) and that a more efficient controller structure would be chosen on the basis of this investigation.

3-4 Grid-Dependent Parameters. [18]

If B is partitioned into a rectangular grid, basis

functions may be formed from piecewise polynomial functions of a single variable with parameters dependent upon the grid coordinates. To illustrate, the case of two state variables, denoted x and y, is discussed.

The rectangular grid is shown in Fig. 3-1. Along each grid segment parallel to one of the axes, say the x axis, the suboptimal control is taken to be a low-order polynomial in x (for concreteness, assume second order).

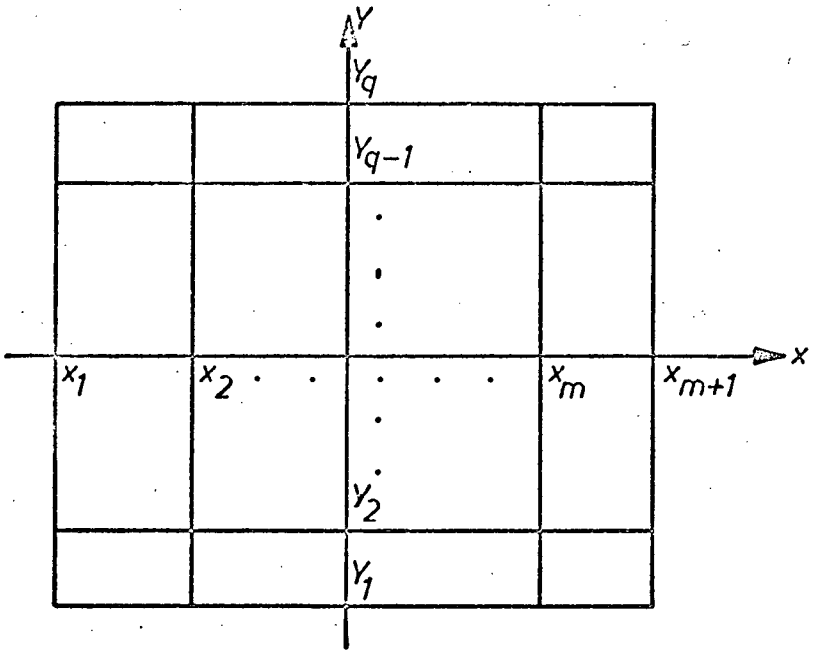

Fig. 3-1 Approximation Grid

Thus along the grid line $y = y_k$, $v(x,y;c)$ is given by

$$g_k(x;c) = \sum_{i=1}^{m} (\alpha_{ki}x^2 + \beta_{ki}x + \gamma_{ki})K_i(x) \quad (3.12)$$

where $K_i(x) = \begin{cases} 1, x_i \leq x \leq x_{i+1} \\ 0, \text{ otherwise} \end{cases}$

and c is the vector of parameters α_{ik}, β_{ik}, γ_{ik}, i=1,...,m,

$k = 1,\ldots,q$. The suboptimal control v between constant -y grid lines is obtained by a linear interpolation process in the y-direction. For example, if Lagrangian interpolation over the full range of y is employed, then

$$v(x,y;c) = \sum_{k=1}^{q} L_k(y) g_k(x;c) \tag{3.13}$$

where

$$L_k(y) = \frac{p(y)}{(y-y_k)p'(y_k)}$$

$$p(y) = \prod_{i=1}^{q} (y-y_i)$$

$$p'(y) = \frac{dp}{dy}$$

Conditions of continuity and/or smoothness can be imposed on (3.13), thus reducing the number of free parameters but without affecting the linearity of v with respect to these parameters. To demonstrate, if we require the functions $g_k(x;c)$ to be spline functions of degree 2 (see Section 3-2), then for each $k(k = 1,\ldots,q)$,

$$\alpha_{ki}x_i^2 + \beta_{ki}x_i + \gamma_{ki} = \alpha_{k,i-1}x_i^2 + \beta_{k,i-1}x_i + \gamma_{k,i-1}$$

$$2\alpha_{ki}x_i + \beta_{ki} = 2\alpha_{k,i-1}x_i + \beta_{k,i-1} \tag{3.14}$$

for $i = 2,\ldots,m$. The minimum of the quadratic function E(c)

given by Eq. (2.10) subject to the linear constraints (3.14) can be computed directly by introducing Lagrange multipliers. However, because of the simple recursive nature of Eqs. (3.14), 2q(m-1) parameters may be explicitly eliminated from (3.13). If the parameters β_{ik}, γ_{ik}, i=2,...,m, k=1,...,q are eliminated, Eq. (3.12) becomes

$$g_k(x;c) = \gamma_{k1} + \beta_{k1}x + \alpha_{k1}(x^2 - (x-x_2)^2U_2(x)) + \sum_{i=2}^{m} \alpha_{ki}((x-x_i)^2U_i(x) - (x-x_{i+1})^2U_{i+1}(x)) \tag{3.15}$$

where $U_i(x) = \begin{cases} 1, & x > x_i \\ 0, & x \leq x_i \end{cases}$

and c denotes the reduced parameter vector. If Eq. (3.15) is substituted into (3.13), the resulting control law is linear in the components of c. The technique of the previous chapter can thus be applied to the determination of c.

3-5 Example.

The technique of the previous section is here applied to the control problem of Section 2-4. For purposes of this example, the two state variables x_1 and x_2 are renamed x and y respectively.

The rectangular grid chosen for the approximation

procedure is shown in Fig. 3-2. Along each of the six grid segments parallel to the x axis, v(x,y;c) was taken to be a polynomial in x of order 3:

$$g_k(x;c) = \sum_{i=1}^{2} (\eta_{ki}x^3 + \alpha_{ki}x^2 + \beta_{ki}x + \gamma_{ki})K_i(x) \qquad (3.16)$$

$$k = 1,2,3$$

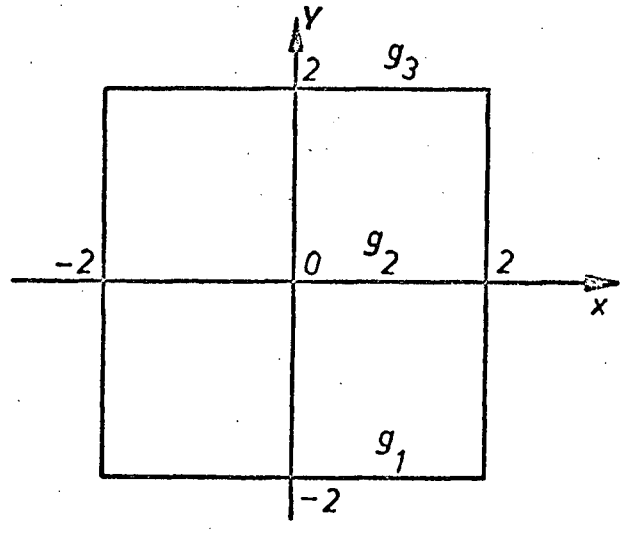

Fig. 3-2 Approximation Grid for Example Problem.

where $K_1(x) = \begin{cases} 1, & -2 \le x \le 0 \\ 0, & x > 0 \end{cases}$ $\quad K_2(x) = \begin{cases} 1, & 0 < x \le 2 \\ 0, & x \le 0 \end{cases}$

At each of the joints (0,2), (0,0), and (0,-2), continuity of the functions $g_k(x;c)$ and their first derivatives is imposed. The symmetry condition

$$v(-x,-y;c) = -v(x,y;c) \qquad (3.17)$$

is also enforced. In terms of the reduced parameter vector, Eqs. (3.16) are then given by

$$g_3(x;c) = (\eta_{31}x^3 + \alpha_{31}x^2)K_1 + \beta_{31}x + \gamma_{31} + (\eta_{32}x^3 + \alpha_{32}x^2)K_2 \qquad (3.18)$$

$$g_2(x;c) = \eta_{21}x^3 + \alpha_{21}x^2\,\mathrm{sgn}(x) + \beta_{21}x \qquad (3.19)$$

and $g_1(x;c)$ follows from (3.18) and the symmetry condition (3.17). Using Lagrangian interpolation, the suboptimal control approximating form is given by

$$v(x,y;c) = \sum_{k=1}^{3} L_k(y)g_k(x;c) \qquad (3.20)$$

where $L_1 = y(y-2)/8 \quad L_2 = (4-y^2)/4 \quad L_3 = y(y+2)/8$

The nine free parameters in (3.20) were evaluated (see Table 3.1) by the procedure of Chapter 2 using the ten development trajectories numbered 1-10 in Fig. 2-1.

Table 3.1 Controller Parameters.

β_{21}	-0.1872	γ_{31}	-4.9151	η_{31}	0.6531
α_{21}	-0.4272	β_{31}	2.3054	α_{32}	1.2016
η_{21}	-.0432	α_{31}	3.5534	η_{32}	-0.7532

Testing the resulting control law for the seven test trajectories (A-G) displayed in Fig. 2-1 revealed that the worst case performance increase was only ½% (attained for test trajectory E). In Fig. 3-3, the quasi-optimal feedback control v is plotted. This figure should be compared with the optimal surface in Fig. 2-2.

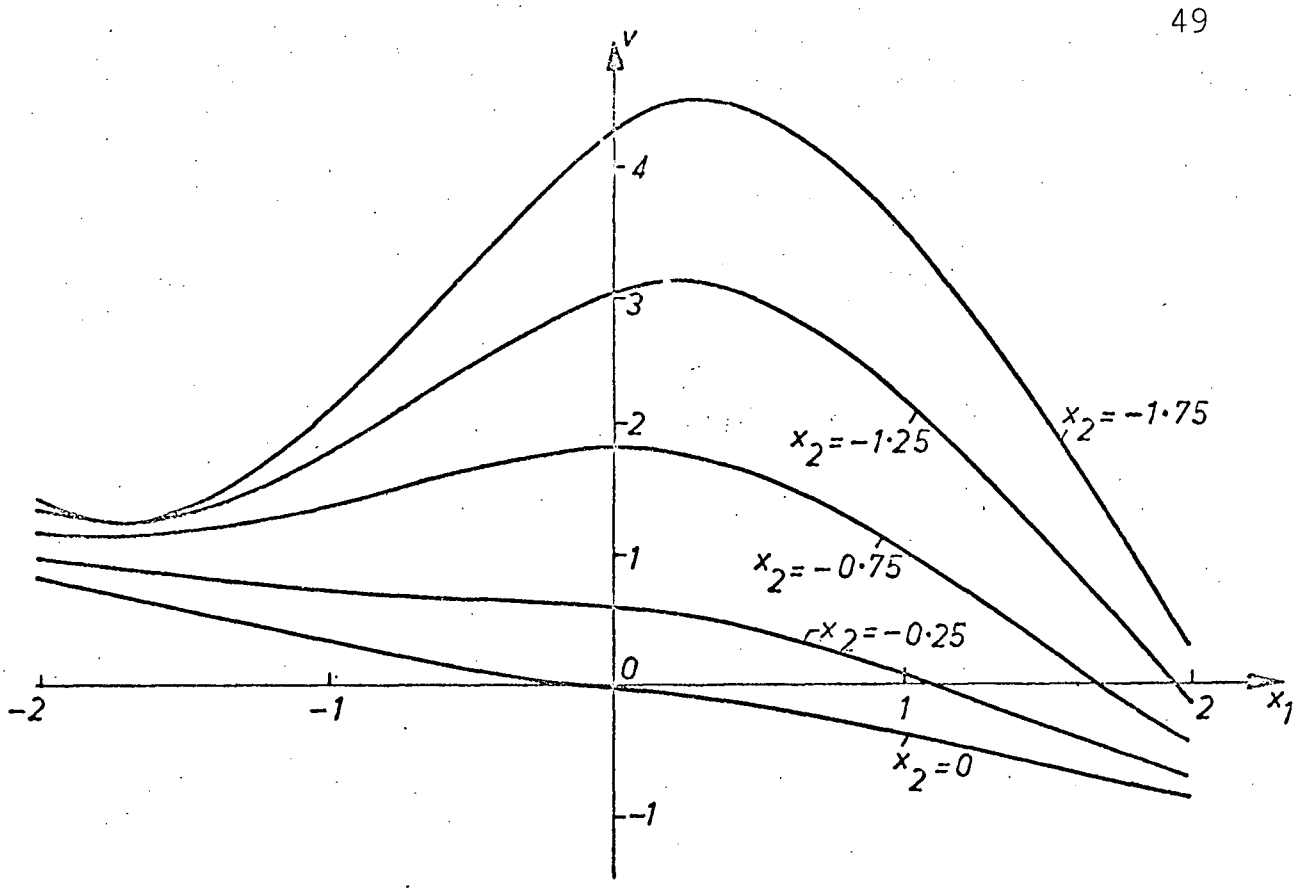

Fig. 3-3 Quasi-Optimal Feedback Surface.

3-6 Kolmogorov's Representation Theorem.

A theorem recently developed by Kolmogorov and others ([19], Chpt. 11) is of considerable potential importance, not only for the determination of suitable sub-optimal feedback control laws but also for the approximation of general multivariable functions. This theorem, which states in essence, that multivariable functions can be represented by functions of a single variable, is presented in this section as a stimulus to further research into control law representations.

The following statement of Kolmogorov's theorem is taken from [19] with slight modifications in symbolism. There

exist n constants $0 < \lambda_i \leq 1$, i=1,...,n and 2n+1 functions $\phi_q(x)$,q=0,1,...,2n defined on $[0,1]$ and with values in $[0,1]$, which have the following properties: the ϕ_q are strictly increasing and belong to a class Lip α*. For each continuous function f defined on the unit cube in E^n, one can find a continuous function g(u), $0 \leq u \leq n$ such that

$$f(x_1,\ldots,x_n) = \sum_{q=0}^{2n} g(\lambda_1\phi_q(x_1) + \ldots + \lambda_n\phi_q(x_n)) \quad (3.21)$$

The representation of multivariable functions given by Eq. (3.21) has several appealing features. First, a definite form with some structural information is provided as a starting point for approximation. Secondly, if analog instrumentation of the suboptimal feedback control is contemplated, the advantages of the representation (3.21) should be evident. Each of the single variable functions ϕ_q can be easily formed with a function generator as can the single-variable function g(u). The remaining linear algebraic operations are also conveniently performed with analog equipment. It is remarked that the class Lip α includes the class of continuous piecewise polynomial

---

* Functions f in this class satisfy a Lipshitz condition of order α, that is, if f(x) is defined on an interval I, there exist two positive constants M and α such that

$$|f(x_1) - f(x_2)| \leq M\,|x_1 - x_2|^{\alpha} \quad \text{for all } x_1, x_2 \epsilon I$$

functions, of which the continuous piecewise linear functions appear particularly attractive for use with function generators. Perhaps the most important benefit which might accrue from exploiting Kolmogorov's representation is the conceptual insight possible with single variable functions.

These advantages will not be easily gained, however. Simultaneous approximation of the functions ϕ_q and the function g in the form (3.21) will require that a nonlinear approximation problem be solved. Whether the ϕ_q can be approximated separately from g is a topic for future research.

4. EVALUATION AND CONCLUSIONS: PART I

4-1 Control Sensitivity.

Successful application of the proposed synthesis technique will depend to a large extent on the sensitivity of end constraints and performance functional to control perturbations. The greater the insensitivity to control errors that exists in a problem, the greater will be the tolerable approximation error and hence, a greater likelihood of obtaining an acceptable suboptimal control.

Bélanger [20] has employed a first-order analysis to determine the effect of control errors on terminal accuracy. He has shown that only in the case where the dimension of the terminal manifold is one less than the dimension of the state-time product space (that is, $m = 1$ in Eq. (1.2)) is it possible to specify a tolerance on the optimal control such that all controls within this tolerance will transfer the system to the target set. Moreover, to first order, such controls cause no deviation from optimal performance since each perturbed trajectory reaches the target set and the variation of J is zero, to first-order, for an optimal trajectory. For example, if the terminal manifold is geometrically equivalent to a closed hypersurface surrounding the origin ($m = 1$ case), any control law will be allowable in the sense of meeting end conditions if the closed-loop

system is asymptotically stable.

It is often feasible to convert the end condition requirements for a problem to a single equation. For example, if the original target set is a point, this can be approximated by a suitably small sphere or ellipsoid about this point, as in the examples of Chapter 2. Physical considerations of accuracy and engineering considerations of economy usually dictate that mathematically "tight" terminal specifications be relaxed somewhat for the control system design.

4.2 Instrumentation: Incomplete State Feedback.

Optimal control laws may require unjustifiably sophisticated instrumentation to implement. One of the desirable features of specific optimal controllers (see Section 2-3) is that instrumentation constraints can be incorporated into the suboptimal control law Eq. (2.8). An important example of such a constraint is the inaccessibility of certain state variables (discussed briefly in Section 1-2). A direct, though not necessarily satisfactory means of dealing with this problem is simply to omit the inaccessible state variables from the suboptimal feedback law. This approach is discussed in [10] and [21] for a linear system and a quadratic performance functional.

In certain cases, simply ignoring inaccessible state variables may yield a satisfactory suboptimal control system and, in these cases, the synthesis procedure of this

thesis may be used to determine it. However, considerable improvement at little extra cost may result from simple estimation schemes. This is illustrated by the improved system behaviour which often results from the use of compensating networks in classical linear control systems.

4-3 Switching (Bang-Bang) Control Systems.

Certain optimal control systems require that the control signal switch from one value to another discontinuously. The optimal feedback control is then determined by the algebraic sign of switching functions. In such cases, it is obviously more convenient to approximate the switching surfaces directly rather than attempting an approximation in the form of Eq. (2.8) for the control function. An approach based on a mean-square fit to points on the switching surface has been taken by Smith [22]. A learning-algorithm approach was taken by Mendel and Zapalac [23] who describe their synthesis technique as off-line training of a realizable controller. It is interesting that the method proposed in this thesis may also be looked upon as one of off-line training in which the development trajectories are regarded as constituting the "training set", the prespecified controller function (Eq. (2.8)) as the "trainable controller" and the computational procedure described in Section 2-3.3 as the "training algorithm".

4-4 Comparison With Alternative Procedures.

There are essentially three different quantities which might be approximated in attempting to synthesize a nearly optimal feedback control. In this thesis, a direct approximation of the optimal feedback control $u(x)$ is attempted. Another possibility is $p(x)$, the optimal adjoint variable as a function of state. Finally, an approximation of the minimum value $V(x)$ of the performance index as a function of state may be attempted. If either $p(x)$ or $V(x)$ is approximated, the suboptimal control is obtained through the necessary conditions for an optimum. For the control problem of Section 2-2, the Hamiltonian is

$$H(x,p,u) = F(x,u) + p^T f(x,u) \tag{4.1}$$

If $p(x)$ is approximated by $p(x;c)$, the suboptimal control $v(x;c)$ is obtained by minimizing (4.1) with $p = p(x;c)$. It can be shown (see for example [24], pp. 14-17) that

$$p(x) = V_x^T(x) \tag{4.2}$$

Thus, if $V(x)$ is approximated by $V(x;c)$, the suboptimal control is obtained by minimizing (4.1) with the gradient of $V(x;c)$ substituted for p.

Assume that

$$u = g(x,p) \tag{4.3}$$

minimizes (4.1). Only those components of p which appear

explicitly in the relation (4.3) need be approximated. Even so, several components of p may have to be approximated in order to obtain the suboptimal control. This extra work could only be justified in the case where a switching control is expected since p will be continuous and, presumably, easier to approximate than u. Kipiniak ([11], pp. 94-117) presents a scheme for approximating p(x) which involves elaborate plotting and cross plotting and eventual curve fitting of the graphical data.

If either p(x) or V(x) is approximated, the designer cannot exercise direct control on the instrumentation required to implement the control since further operations must be performed on the approximation to obtain the suboptimal control.

The chief advantage in approximating V(x) is that V is always a continuous, non-negative scalar function, independent of the dimension of u. This advantage is more than offset, however, by the fact that partial derivatives of V must be taken to obtain a suboptimal control. A close approximation of V does not necessarily imply a close approximation of its derivatives. Durbeck [25] proposed a method where V is approximated for an infinite interval ($t_f = \infty$) process. The parameters in the approximation must be determined by a cumbersome descent minimization of a non-linear function. This technique is limited to a relatively few number of parameters because of the required use of non-linear programming methods. Gragg [24] approximates V by a

least-squares procedure. The chief disadvantage of Gragg's procedure is the previously mentioned one of having to take partial derivatives of the approximation.

4-5 Summary and Conclusions.

A technique for synthesizing nearly-optimal feedback control functions has been presented. To begin the synthesis procedure, a controller input-output relation dependent upon a set of adjustable parameters must be prescribed. The "distance" between the suboptimal controller and the optimal feedback controller is measured by a sum of integral square deviations between the optimal control and the suboptimal control along several system trajectories. Choosing the controller parameters to minimize this distance results in an overall computational algorithm which is simple enough to make experimentation with different controller structures completely feasible.

If little is known about the algebraic form of the optimal feedback control function, piecewise polynomial basis functions are advocated. Low-order piecewise polynomial basis functions will provide an accurate approximation of the optimal feedback surface but without having the undesirable numerical properties of high-order polynomial basis functions. In addition, piecewise polynomial functions possess a flexibility which could never be equalled by analytic functions (polynomials, for example). Even if it were numerically possible to compute a high-order polynomial

approximation, the flexibility of piecewise polynomial functions will allow more complicated optimal feedback functions to be approximated with a fewer number of parameters.

It was shown in Section (4.1) how the allowable approximation error is related to the terminal specifications. Full advantage must be taken of terminal constraint tolerances to reduce the sensitivity to control errors.

In principle, the synthesis technique explored in this thesis is very general. In applying it to practical problems, two major hurdles must be overcome. First, a suitable controller structure must be found. This is the major topic of Part I. Secondly, many optimal trajectories must be computed. This job is, by far, the most (computer) time consuming numerical task faced by the user of this technique and constitutes the subject of the next two chapters.

5. THE EXTENDED NEWTON-RAPHSON METHOD WITH THE GENERALIZED RICATTI TRANSFORMATION

5-1 Applications of Optimal Control Programs.

Applications of optimal control programs can be grouped into those requiring on-line solution of the optimization problem and those requiring off-line computation only. On-line computation would be called for in a closed-loop application using optimal open-loop control with periodic updating based on the latest sampled state. To the author's knowledge, this is a speculated application only and has never been actually implemented. Off-line solutions of the optimization problem may be utilized in the design stage. Many optimal trajectories are required for the synthesis procedure advocated in this thesis. Moreover, any preliminary study of an optimal or suboptimal control system design will require the computation of a few optimal trajectories. In guidance applications, an optimal trajectory is often used as a reference trajectory for the guidance law ("open-loop steering") and feedback control is based on deviations from this reference trajectory [26].

Many approaches to numerically solving the optimal control problem have been taken, all of which lead to iterative procedures. The properties of an iterative algorithm considered most desirable in applications are, first of all, a wide region of convergence and second, a fast speed of convergence. It is difficult to compare regions of convergence between various techniques because

often the regions belong to different spaces. Qualitatively, however, a comparison can be based on the ease with which starting elements can be specified so that the iterations will converge from these elements without intervention. The desirability of having a wide region of fast convergence is not largely influenced by whether the computation is to be performed on-line or off-line. For on-line applications, it is probably desirable that the algorithm have a rather small memory requirement. On the other hand, in off-line applications where a block of fast memory is allotted and costs do not depend on what fraction of that block is actually used, the memory demanded by the algorithm is of no consequence, provided, of course, the demand does not exceed the memory allottment.

5-2 The Newton-Raphson Method.

The method to be presented in this chapter is an extension and modification of the function space Newton-Raphson method [8] (also called quasilinearization [27]). Experience gained in several numerical studies ([28], for example) has shown that for the Newton-Raphson method, it is normally rather easy to choose starting elements from which the process will converge. Moreover, the rate of convergence is quadratic in the vicinity of the iteration fixed point.

In [8] and subsequent papers of these authors [29], only the "point-type" terminal condition is treated, that is,

a particular state variable has either a specified terminal value or is free. By relabelling the state variables if necessary, it may be assumed that the first m are specified. Thus for this case, Ψ in Eq. (1.2) has the special form

$$\Psi(x(t_f),t_f) = I_{nm}^T x(t_f) - \hat{x}_f \tag{5.1}$$

where $\hat{x}_f$ is an m-vector of constants, I_{nm} is a nxm matrix defined by

$$I_{nm} = \begin{bmatrix} I_m \\ \hline 0 \end{bmatrix} \tag{5.2}$$

I_m is the mxm unit matrix and 0 in (5.2) is the (n-m)xm zero matrix. In addition, the original Newton-Raphson method assumes that t_f in (5.1) is fixed. Free terminal time problems are handled by solving a sequence of fixed time problems, a device which will be explained in more detail in Section 5-5.

The primary difficulty encountered in employing the Newton-Raphson method lies in the instability of the differential equations which must be integrated at each iteration. In Section 5-4, a transformation is introduced which largely overcomes this difficulty.

5-3 The Extended Newton-Raphson Method.

For the free terminal time control problem described by Eqs. (1.1), (1.2) and (1.3), the conditions

which must be satisfied by the extremal are [26]:

$$\dot{x} = f(x,u,t) \qquad x(t_o) = x_o \tag{5.3}$$

$$\dot{p} = - H_x^T(x,p,u,t) \tag{5.4}$$

$$0 = H_u^T(x,p,u,t) \tag{5.5}$$

$$p(t_f) = \Phi_x^T(x(t_f),\nu,t_f) \tag{5.6}$$

$$\Psi(x(t_f),t_f) = 0 \tag{5.7}$$

$$\Omega(x(t_f),p(t_f),u(t_f),\nu,t_f) = \Phi_t(x(t_f),\nu,t_f) + H(x(t_f), p(t_f),u(t_f),t_f) = 0 \tag{5.8}$$

where $\Phi(x,\nu,t) = \phi(x,t) + \nu^T\Psi(x,t)$

$H(x,p,u,t) = F(x,u,t) + p^Tf(x,u,t)$

p is an n-vector of time-varying multipliers

ν is an m-vector of constant multipliers

It is assumed that any state or control inequality constraints have been approximated by including penalty terms in the performance functional. Equations (5.3) - (5.8) represent a nonlinear two-point boundary-value problem (TPBVP).

The Newton-Raphson method for solving optimization problems is a function space generalization of the familiar zero-finding technique of the same name. Suppose it is desired to find a real number x such that the scalar function

f vanishes at x. If an estimate x^{i-1} is available, the next estimate x^i is obtained by assuming $f(x^i) = 0$ and then expanding about x^{i-1} up to linear terms only; that is

$$0 \simeq f(x^{i-1}) + f'(x^{i-1})(x^i - x^{i-1}) \qquad (5.9)$$

Thus, the solution of the nonlinear equation is replaced by the solution of a sequence of linear equations.

Proceeding analogously, suppose that estimates $\bar{x}$, $\bar{p}$, $\bar{u}$, $\bar{\nu}$, $\bar{t}_f$ of the solution of Eqs. (5.3) - (5.8) are available. The overbar is used instead of a superscript i-1 for notational convenience. In addition, arguments of functions are not written explicitly where no confusion should arise. Thus for example, $\bar{x}$ stands for a vector time function defined on $[t_o, t_f]$. It is assumed that Eq. (5.5) is satisfied by $\bar{x}$, $\bar{p}$, $\bar{u}$, that is, u is eliminated implicitly or explicitly by (5.5). Otherwise, the $(i-1)^{st}$ iterate need not satisfy any of the other necessary conditions. If the next iterate satisfied Eqs. (5.3) - (5.8), then x^i, for example, would be the solution of

$$\dot{x}^i = f(x^i, u^i, t) \qquad (5.10)$$

The Newton-Raphson linearization of (5.10) is

$$\dot{x}^i = f(\bar{x}, \bar{u}, t) + \bar{f}_x(x^i - \bar{x}) + \bar{f}_u(u^i - \bar{u}) \qquad (5.11)$$

Dropping the superscript i (for ease of notation) and the overbar from partial derivatives where it is to be understood that all partial derivatives are evaluated for

previous iterate quantities, the desired linearized state equation is

$$\dot{x} = f_x x + f_u u + (\bar{f} - f_x \bar{x} - f_u \bar{u}) \qquad (5.12)$$

Equation (5.4) is treated in exactly the same manner. The control can be eliminated since for each iterate,

$$H_u^T(x^i, p^i, u^i, t) = 0 \qquad (5.13)$$

The first-order difference $u - \bar{u}$ is

$$u - \bar{u} = -H_{uu}^{-1}(H_{ux}(x - \bar{x}) + f_u^T(p - \bar{p})) \qquad (5.14)$$

where it is assumed that H_{uu} is nonsingular. The Newton-Raphson linearization of Eqs. (5.6) - (5.8) is carried out in Appendix B.

Using (5.14) to eliminate $(u - \bar{u})$, the complete set of linear equations analogous to Eq. (5.9) is given below and the symbol definitions are in Table 5.1.

$$\dot{x} = Ax + Bp + a \qquad (5.15)$$

$$\dot{p} = Cx - A^T p + b \qquad (5.16)$$

$$p(\bar{t}_f) = \Phi_{xx} x(\bar{t}_f) + \psi_x^T \nu + \alpha t_f + \lambda \qquad (5.17)$$

$$\psi = 0 \quad = \psi_x x(\bar{t}_f) + \beta t_f + \theta \qquad (5.18)$$

$$\Omega = 0 \quad = \bar{\alpha}^T x(\bar{t}_f) + \bar{\beta}^T \nu + \gamma t_f + \omega \qquad (5.19)$$

Equations (5.15) - (5.19) represent a linear

Table 5.1 Symbol Definitions for Eqs. (5.15)-(5.19).

Symbol	Use reference	Definition
A *	5.15	$f_x - f_u H_{uu}^{-1} H_{ux}$
B *	5.15	$-f_u H_{uu}^{-1} f_u^T$
C *	5.16	$-H_{xx} + H_{xu} H_{uu}^{-1} H_{ux}$
a *	5.15	$\bar{f} - A\bar{x} - B\bar{p}$
b *	5.16	$-H_x^T - C\bar{x} + A^T\bar{p}$
$\bar{\beta}$	5.19	$(\Psi_t + \Psi_x \bar{f})_{\bar{t}_f}$
β	5.18	$\bar{\beta} - (\Psi_x X)_{\bar{t}_f}$
X *	β	$\bar{f} - \dot{\bar{x}}$
$\bar{\alpha}$	5.19	$(\Phi_{xt} + \Phi_{xx}\bar{f} + H_x^T)_{\bar{t}_f}$
α	5.17	$\bar{\alpha} - (\Phi_{xx} X + P)_{\bar{t}_f}$
P *	α	$H_x^T + \dot{\bar{p}}$
γ	5.19	$\left[(\Phi_{tx} + \bar{\alpha}^T - X^T \Phi_{xx})(\bar{f} - X) + \Omega_t\right]_{\bar{t}_f}$
λ	5.17	$(\Phi_x^T - \Phi_{xx}\bar{x} - \Psi_x \bar{\nu})_{\bar{t}_f} - \alpha \bar{t}_f$
Θ	5.18	$(\bar{\Psi} - \Psi_x \bar{x})_{\bar{t}_f} - \beta \bar{t}_f$
ω	5.19	$(\bar{\Omega} + (\Phi_x - \bar{p}^T)\bar{f} - \bar{\alpha}^T\bar{x} - \bar{\beta}^T\bar{\nu})_{\bar{t}_f} - \gamma \bar{t}_f$

* Functions of time

TPBVP, the solution of which is the next (i^{th}) iterate. Thus, the original nonlinear TPBVP has been replaced by a sequence of linear TPBVP's. Useful conditions for the convergence of this iterative process have not yet been established. Some convergence theorems are given in [29] for the related Newton-Raphson method [8] but these sufficient conditions for convergence are generally very difficult or impossible to check and are extremely restrictive in the sense that they will not be satisfied for most problems of interest. In practice, it is found that convergence is generally obtained if the starting functions are "close" enough to the converged solution.

The Newton-Raphson method of [8] was also extended to cover the general terminal condition (Eq. (5.7)) by Lewallen [30]. Before discussing the solution of the linear TPBVP (5.15) - (5.19), it will be made clear how the present method differs from that in [30]. Lewallen's procedure requires that the terminal conditions (5.6) - (5.8) be expressed in the form

$$h(x(t_f),p(t_f),t_f) = 0 \qquad (5.20)$$

where h is an (n+1)-vector function. In essence, this requires that ν be explicitly eliminated by solving m of the n equations (5.6) for ν and substituting in the remaining (n-m) equations of (5.6) and in (5.8). Apart from the algebraic complexity which can arise, the resulting linearized boundary conditions (linearization of (5.20)) are such

that, in general, a Ricatti transformation cannot be utilized in solving the TPBVP. Thus, the present treatment differs first, in the form in which the boundary conditions are handled and second, in the method of solving the linear TPBVP.

5-4 Generalized Ricatti Transformation.

The standard method of solving the linear TPBVP associated with the Newton-Raphson method requires several integrations of the linearized canonical system, Eqs. (5.15) and (5.16). This gives rise to fundamental problems of numerical stability [31] since as a coupled system, the canonical differential equations have solutions containing both fast-growing (unbounded) and fast-decaying components.

An approach utilizing the Ricatti transformation in connection with the Newton-Raphson method has been taken [32] for the special problem where the terminal time is specified and the terminal states are free. In [33], the generalized Ricatti transformation was used in connection with a different computational technique. It is here applied to the extended Newton-Raphson technique of Section (5.3).

Consider the transformation

$$\begin{bmatrix} p(t) \\ \Psi \\ \Omega \end{bmatrix} = \begin{bmatrix} R(t) & L(t) & l(t) \\ L^T(t) & Q(t) & m(t) \\ \bar{l}^T(t) & \bar{m}^T(t) & n(t) \end{bmatrix} \begin{bmatrix} x(t) \\ \nu \\ t_f \end{bmatrix} + \begin{bmatrix} q(t) \\ r(t) \\ s(t) \end{bmatrix} \tag{5.21}$$

This transformation must be compatible with the linearized equations (5.15)-(5.19). Differentiating Eqs. (5.21) yields

$$\begin{bmatrix} \dot{p} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{R} & \dot{L} & \dot{l} \\ \dot{L}^T & \dot{Q} & \dot{m} \\ \dot{\bar{l}}^T & \dot{\bar{m}}^T & \dot{n} \end{bmatrix} \begin{bmatrix} x \\ \nu \\ t_f \end{bmatrix} + \begin{bmatrix} R & L & l \\ L^T & Q & m \\ \bar{l}^T & \bar{m}^T & n \end{bmatrix} \begin{bmatrix} \dot{x} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{q} \\ \dot{r} \\ \dot{s} \end{bmatrix} \tag{5.22}$$

where $$\dot{x} = (A + BR)x + B(L\nu + lt_f + q) + a \tag{5.23}$$

and $$\dot{p} = (C - A^T R)x - A^T(L\nu + lt_f + q) + b \tag{5.24}$$

If (5.22) is to be an identity in x, ν, and t_f, and the Ricatti transformation (5.21) is to be compatible with the linearized boundary conditions (5.17) - (5.19), the Ricatti coefficients must satisfy the differential equations and terminal conditions listed in Table 5.2.

Note that the differential equations satisfied by l and $\bar{l}$ and by m and $\bar{m}$ are, in each case, of exactly the same form; only the terminal conditions are different. However, examining Table 5.1, it can be seen that if the previous iterate were in fact the extremal, then $X \equiv P \equiv 0$ and hence, $\alpha = \bar{\alpha}$, $\beta = \bar{\beta}$. Thus, in the limit as the iteration converges, $l \to \bar{l}$, $m \to \bar{m}$ and hence the transformation matrix in (5.21) becomes symmetric since R and Q are symmetric.

Table 5.2 Differential Equations and Terminal Conditions for the Ricatti Coefficients.

Coefficient	Differential Equation	Terminal Condition
R	$-\dot{R} = RA + A^T R + RBR - C$	Φ_{xx}
L	$-\dot{L} = (A^T + RB)L$	Ψ_x^T
Q	$-\dot{Q} = L^T B L$	0
q	$-\dot{q} = (A^T + RB)q + Ra - b$	λ
r	$-\dot{r} = L^T(a + Bq)$	θ
l	$-\dot{l} = (A^T + RB)l$	α
$\bar{l}$	$-\dot{\bar{l}} = (A^T + RB)\bar{l}$	$\bar{\alpha}$
m	$-\dot{m} = L^T B l$	β
$\bar{m}$	$-\dot{\bar{m}} = L^T B \bar{l}$	$\bar{\beta}$
n	$-\dot{n} = \bar{l}^T B l$	γ
s	$-\dot{s} = \bar{l}^T(a + Bq)$	ω

5-4.1 Computational Procedure.

R,L,l,$\bar{l}$ and q are integrated in reverse time from the terminal conditions specified in Table 5.2 to the initial time. The remaining coefficients are required only at $t = t_o$ and can be obtained by quadrature. The new values of ν and t_f are found by solving

$$\begin{bmatrix} Q(t_o) & m(t_o) \\ \bar{m}^T(t_o) & n(t_o) \end{bmatrix} \begin{bmatrix} \nu \\ t_f \end{bmatrix} = \begin{bmatrix} \Psi - L^T(t_o)x_o - r(t_o) \\ \Omega - \bar{l}^T(t_o)x_o - s(t_o) \end{bmatrix} \qquad (5.25)$$

which are the last two equation sets from (5.21) evaluated at $t = t_o$. According to the actual Newton-Raphson linearization of the necessary conditions, Ψ and Ω in (5.25) should be zero. Step-size control may be exercised, however, by requiring that only a fraction of the remaining necessary condition errors be corrected at any one step. Thus, at the i^{th} iteration take

$$\begin{bmatrix} \Psi \\ \Omega \end{bmatrix} = \varepsilon^i \begin{bmatrix} \bar{\Psi} \\ \bar{\Omega} \end{bmatrix} \qquad 0 \leq \varepsilon^i < 1 \qquad (5.26)$$

The other necessary condition errors, namely X, P and $(\Phi_x^T - \bar{p})_{\bar{t}_f}$, may be limited in a similar way.

Having obtained the new estimates of ν and t_f,

the new state trajectory estimate is obtained by integrating (5.23) forward from t_o to t_f. The new adjoint vector estimate is computed from the first equation set in (5.21).

Convergence is checked by evaluating some distance function of the present and previous iterates. For example, let

$$\hat{t}_f = \min(t_f^i,\ t_f^{i-1}) \tag{5.27}$$

and let

$$z = \begin{bmatrix} x \\ p \end{bmatrix} \tag{5.28}$$

Then, the distance between the 2n-vector functions of time $z^i(t)$ and $z^{i-1}(t)$ defined on the intervals $\left[t_o, t_f^i\right]$ and $\left[t_o, t_f^{i-1}\right]$ respectively is given by

$$\left\| z^i - z^{i-1} \right\| = \max_{t \epsilon \left[t_o, \hat{t}_f\right]} \left(\max_k \left| z_k^i(t) - z_k^{i-1}(t) \right| \right) + w \left| t_f^i - t_f^{i-1} \right| \tag{5.29}$$

where w is a positive constant weighting the terminal time difference. The iterations are stopped when $\left\| z^i - z^{i-1} \right\|$ is less than some convergence factor.

5-4.2 Stability of the Differential Equations.

From Eq. (5.23) and Table 5.2, it can be seen that

there are essentially only two types of differential equations that need be solved in this iterative scheme. The first is the equation for R (the matrix Ricatti differential equation) which must be solved in reverse time and the other is the linear system typified by the homogeneous part of Eq. (5.23). The equations for L,l and $\bar{l}$ are adjoint to this system as is the equation for q but with a driving function. All the others can be solved by quadrature.

Sufficient conditions for the asymptotic stability of the Ricatti equation in reverse time have been given by Kalman [34] and computational experience is reported by Merriam [35] . From the work in this latter reference and the experience of others including the author's, it may be said that, in practice, for problems having a nonsingular H_{uu} (as assumed here), the matrix Ricatti equation has a bounded solution in reverse time provided the Ricatti matrix has a positive semi-definite value at $\bar{t}_f$. This statement applies in a neighborhood of the sought for extremal. Thus, although adjustments may have to be made to the starting trajectories and initial value of ν such that the above conditions apply, this had not yet been necessary in any computational work attempted.

The behaviour of Eq. (5.23) can best be judged by recognizing the homogeneous part as having the properties of the closed-loop linearized system for the problem with no terminal conditions (see Section 5-5.1). Again possibly just in a neighborhood of the extremal, it is safe to assume that

the solution is decaying. The equations for L,q,l and $\bar{l}$ have qualitatively similar properties in reverse time since they are adjoint to (5.23).

In summary, then, the differential equations which must be solved at each iteration have stability properties which are much more desirable for numerical computation than those possessed by the coupled linearized canonical system.

5-5 Algorithm for Fixed Terminal Time Problems.

The linearization of the necessary conditions is considerably less complex when t_f is given explicitly since variations in t_f need not be considered and Eq. (5.8) is no longer necessary.

For this case, the appropriate Ricatti transformation is

$$\begin{bmatrix} p \\ \psi \end{bmatrix} = \begin{bmatrix} R & L \\ L^T & Q \end{bmatrix} \begin{bmatrix} x \\ \nu \end{bmatrix} + \begin{bmatrix} q \\ r \end{bmatrix} \tag{5.30}$$

where R,L,Q,q and r are defined by the first five entries in Table 5.2. If α and β are taken to be zero, the definitions of λ and θ given in Table 5.1 are correct for the fixed time case as well. The decoupled state differential equation is given by

$$\dot{x} = (A + BR)x + BL\nu + Bq + a \tag{5.31}$$

In solving the free terminal time problem, it may be

desirable to solve a sequence of fixed-time problems rather than to employ the algorithm in Section 5-4. Referring to Eq. (5.23), it can be seen that if the newly determined t_f is larger than the old value, extrapolation of the time-dependent coefficients in (5.23) will be required. Although inconvenient, this difficulty is common to all optimization techniques which determine a new estimate of the terminal time at each iteration. The fixed-time approach consists in guessing a value for t_f, solving the fixed-time problem and then repeating this for another value of t_f. From then on, the sequence of terminal times $\left\{t_f^k\right\}$ is determined by

$$t_f^{k+1} = t_f^k + \left(\frac{t_f^k - t_f^{k-1}}{\Omega^k - \Omega^{k-1}}\right)(-\Omega^k) \tag{5.32}$$

where Ω is defined by Eq. (5.8) and $\Omega^k = \Omega(t_f^k)$. Equation (5.32) is a discrete approximation of the ordinary Newton method for finding the zero of $\Omega(t_f)$.

5-5.1 Free Terminal State. ($m = 0$)

Since m=0, L,Q and r vanish and (5.30) simplifies further to

$$p = Rx + q \tag{5.33}$$

The differential equations for R and q are unchanged but the terminal conditions become

$$R(t_f) = \phi_{xx} \tag{5.34}$$

$$q(t_f) = \phi_x^T - \phi_{xx}\bar{x} \tag{5.35}$$

and the state equation is (5.31) with $L \equiv 0$. For this special case, the algorithm coincides with that in [32].

5-6 Numerical Integration Method.

The choice of numerical integration procedure for the extended Newton-Raphson method (ENRM) requires some consideration. At each iteration, there are two periods of integration required. The first is the reverse-time integration of the Ricatti coefficients R,L and q whose equations involve the stored time functions $\bar{x},\bar{p}$. After obtaining the other coefficients by quadrature and solving for ν, the decoupled state system (5.31) which contains the stored time functions just generated as well as $\bar{x}$ and $\bar{p}$ is integrated forward. If it is desired that elaborate interpolation of stored data not be required, two conclusions should be evident: throughout any integration the same step size should be used and the same step size should be used in integrating the Ricatti system as in integrating the state equations. Thus, numerical integration methods which attempt to "optimize" the step size at each step or which exercise error control by interval halving during an integration are not desirable for the purposes of this algorithm. Furthermore, the popular single-step methods such as the Runge-Kutta method are not a good choice because they require evaluation of the derivative functions at fractions of intervals which again would require interpolation of time functions.

These considerations dictate that a multistep predictor-corrector integration formula be used. Although somewhat more difficult to use than Runge-Kutta methods, predictor-corrector integration is about twice as fast for a given accuracy (truncation error) and a fixed step size. A technique found most satisfactory for the ENRM is a fifth-order method due to Hamming [36] in which the corrector is not iterated and only two evaluations of the derivative functions are required at each step (see Appendix C).

5-7 Neighborhood Optimal Controller.

The iterative scheme which yields the optimal trajectory also determines the time-varying gains for optimal linear feedback control about the optimal trajectory. Consider the fixed terminal time problem and the algorithm of Section 5-5. (The following discussion applies with only slight modifications to the free time case as well). If the previous iterate was, in fact, the optimal trajectory, the linearized control correction required for deviations from optimal is given by Eq. (5.14):

$$\delta u = -H_{uu}^{-1}(H_{ux}\,\delta x + f_u^T\,\delta p) \tag{5.36}$$

From the first set of Eqs. (5.30), the perturbation δp is given by

$$\delta p = R\,\delta x + L\,\delta \nu \tag{5.37}$$

and the second set specifies $\delta\nu$:

$$\delta\nu = - Q^{-1}L^T \delta x \tag{5.38}$$

Substituting (5.37) and (5.38) into (5.36) yields the optimal linear neighborhood control law:

$$\delta u = -K(t)\,\delta x \tag{5.39}$$

where
$$K(t) = H_{uu}^{-1}(H_{ux} + f_u^T(R - LQ^{-1}L^T)) \tag{5.40}$$

The neighborhood controller (5.39) has been derived several times before from different approaches (see [26] and [33], for example). Because of the terminal condition on Q, the gain matrix K is infinite at the terminal time unless the terminal states are free (m=0).

5-8 "Point-Type" Terminal Condition.

In this section, the fixed terminal time algorithm of Section 5-5 is specialized to the "point-type" terminal condition Eq. (5.1), in order to compare the present method with that advocated in [8]. It is further assumed that $\phi=0$ although this is not restrictive because $\dot{\phi}$ may be included in the performance integral. The terminal condition on p thus becomes, from Eqs. (5.1) and (5.6),

$$p(t_f) = I_{nm}\nu \tag{5.41}$$

The Newton-Raphson method (NRM) of [8] solves the linear TPBVP specified by Eqs. (5.15), (5.16), (5.1) and (5.41)

by writing the solution to the linearized canonical system as

$$\begin{bmatrix} x(t) \\ p(t) \end{bmatrix} = Y(t)c_p + y(t) \tag{5.42}$$

where $Y(t)$ is a 2nxn matrix of solutions to the homogeneous part of Eqs. (5.15) - (5.16) with initial conditions specified by

$$Y(0) = \begin{bmatrix} 0 \\ \hline I_n \end{bmatrix} \tag{5.43}$$

I_n in Eq. (5.43) is the nxn unit matrix, 0 is the nxn zero matrix, c_p in (5.42) is an n-vector of undetermined parameters, and $y(t)$ is a 2n-vector particular solution of (5.15)-(5.16) with initial conditions

$$y(0) = \begin{bmatrix} x_0 \\ \hline \bar{p}(0) \end{bmatrix} \tag{5.44}$$

where $\bar{p}(0)$ is the latest estimate of the adjoint initial condition. Equation (5.1) specifies the first m values of $x(t_f)$ and Eq. (5.41) specifies the last (n-m) values of $p(t_f)$ to be zero. Thus, n components of the combined (x,p) vector have known terminal values. From Eq. (5.42) at $t=t_f$, the corresponding n equations are extracted and solved for c_p. The new estimate of $p(0)$ is obtained from (5.42) at $t=0$, that is

$$p(0) = c_p + \bar{p}(0) \tag{5.45}$$

and the new trajectory is generated by integrating (5.15)-(5.16) with the new initial condition or can be computed directly from(5.42) if the solutions $Y(t)$ and $y(t)$ are stored. It can be seen that the constant Lagrange multiplier ν is unnecessary and that the specified terminal values are met by every iterate.

For the extended Newton-Raphson with Ricatti transformation approach (ENRM), the terminal conditions of the Ricatti coefficients are, in this special case,

$$R(t_f) = 0, \quad L(t_f) = I_{nm}, \quad Q(t_f) = 0, \quad q(t_f) = 0,$$

$$r(t_f) = -\hat{x}_f \tag{5.46}$$

From Eqs. (5.30),

$$\begin{aligned} p(t_f) &= L(t_f)\nu \\ \psi = 0 &= L^T(t_f)x(t_f) + r(t_f) \end{aligned} \tag{5.47}$$

which are identical with the required conditions (5.1) and (5.41). Hence, the terminal conditions are satisfied by every iterate in the ENRM as well.

To illustrate the application of the two methods and to compare their suitability for numerical computation, a simple closed-form example is presented in the next section.

5-9 Simple Example.

The dynanics are first order

$$\dot{x} = u \qquad x(0) = x_o \tag{5.48}$$

the final value of x is specified

$$\psi(x(t_f)) = x(t_f) - \hat{x}_f \tag{5.49}$$

and the performance functional is quadratic

$$J(u) = \tfrac{1}{2} \int_0^{t_f} (x^2 + u^2)dt \tag{5.50}$$

The canonical system and terminal conditions are

$$\dot{x} = -p \tag{5.51}$$

$$\dot{p} = -x \tag{5.52}$$

$$x(t_f) = \hat{x}_f \tag{5.53}$$

$$p(t_f) = \nu \tag{5.54}$$

Standard approach (NRM).

Following the procedure described in Section 5-7, the solution is written as Eq. (5.42) where

$$\dot{Y} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} Y \qquad Y(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.55}$$

Solving (5.55) yields

$$Y(t) = \begin{bmatrix} -\sinh(t) \\ \cosh(t) \end{bmatrix} \tag{5.56}$$

The particular solution of Eqs. (5.51) and (5.52) with initial condition (5.44) is

$$y(t) = \begin{bmatrix} \cosh(t)x_o - \sinh(t)\bar{p}_o \\ -\sinh(t)x_o + \cosh(t)\bar{p}_o \end{bmatrix} \tag{5.57}$$

Equation (5.42) becomes, for this example,

$$\begin{bmatrix} x(t) \\ p(t) \end{bmatrix} = \begin{bmatrix} -\sinh(t) \\ \cosh(t) \end{bmatrix} c_p + \begin{bmatrix} \cosh(t)x_o - \sinh(t)\bar{p}_o \\ -\sinh(t)x_o + \cosh(t)\bar{p}_o \end{bmatrix} \tag{5.58}$$

Solving the first of Eqs. (5.58) at $t = t_f$ for c_p and with Eq. (5.53) satisfied yields

$$c_p = -\bar{p}_o + \coth(t_f)x_o - \operatorname{csch}(t_f)\hat{x}_f \tag{5.59}$$

and the new estimate for p_o is, from (5.45),

$$p_o = \coth(t_f)x_o - \operatorname{csch}(t_f)\hat{x}_f \triangleq p_o^* \tag{5.60}$$

The value p_o^* given by (5.60) is the optimal initial condition for p, that is, the method has converged in one iteration since the original TPBVP was linear.

Now suppose that on the previous iterate, the estimate of p_o was

$$\bar{p}_o = p_o^* + \varepsilon \tag{5.61}$$

Then, as is easily demonstrated, the terminal values for the particular solution (5.57) would be

$$\begin{bmatrix} x(t_f) \\ \\ p(t_f) \end{bmatrix} = \begin{bmatrix} x_f \\ \\ p_f^* \end{bmatrix} + \begin{bmatrix} -\sinh(t_f) \\ \\ \cosh(t_f) \end{bmatrix} \varepsilon \qquad (5.62)$$

where p_f^* is the optimal terminal value for p. For example, if $t_f = 10$ and $\hat{x}_f = 0$, then $p_f^* \simeq 0$ and

$$\begin{bmatrix} x(t_f) \\ \\ p(t_f) \end{bmatrix} = \begin{bmatrix} -11{,}000 \\ \\ 11{,}000 \end{bmatrix} \varepsilon \qquad (5.63)$$

New Approach (ENRM).

From Table (5.2) and Eq. (5.46), the Ricatti coefficients satisfy the following equations and terminal conditions

$$\begin{aligned} -\dot{R} &= R^2 + 1 & R(t_f) &= 0 \\ -\dot{L} &= -R(t)L & L(t_f) &= 1 \\ -\dot{q} &= -R(t)q & q(t_f) &= 0 \\ -\dot{r} &= LBq & r(t_f) &= -\hat{x}_f \\ -\dot{Q} &= -L^2 & Q(t_f) &= 0 \end{aligned} \qquad (5.64)$$

whose solutions are given by

$$\begin{aligned} R(t) &= \tanh(t_f - t) \\ L(t) &= \operatorname{sech}(t_f - t) \\ q(t) &= 0 \\ Q(t) &= -\tanh(t_f - t) \\ r(t) &= -\hat{x}_f \end{aligned} \qquad (5.65)$$

Note that all solutions (5.65) are bounded as $(t_f - t) \to \infty$. The new value of ν is given by

$$\begin{aligned} \nu &= -Q^{-1}(0)(L(0)x_o + r(0)) \\ &= \frac{\operatorname{sech}(t_f)x_o - \hat{x}_f}{\tanh(t_f)} \triangleq \nu^* \end{aligned} \tag{5.66}$$

which is also optimal because of the previously mentioned linearity. The decoupled state system satisfies

$$\dot{x} = -R(t)x - L(t)\nu \tag{5.67}$$

and at $t = t_f$,

$$x(t_f) = \operatorname{sech}(t_f)x_o - \tanh(t_f)\nu \tag{5.68}$$

Again, suppose that the previous estimate of ν was

$$\bar{\nu} = \nu^* + \varepsilon \tag{5.69}$$

Then, from (5.68),

$$\left| x(t_f) - \hat{x}_f \right| \leq \varepsilon \tag{5.70}$$

and from (5.30) with $t_f \simeq 10$ as before,

$$p(t_f) \simeq x(t_f) \tag{5.71}$$

Thus, the sensitivity of the end values to changes in the boundary value parameter is reduced several orders of magnitude by utilizing the approach of this chapter.

5-10 Numerical Example.

The many optimal trajectories and control programs required for the feedback control synthesis in Example 2, Section 2-5, were computed by the method of this chapter. In particular, the method outlined in Section 5-5 for solving free terminal time problems as a sequence of fixed time problems was employed. Typically, six to eight shifts of the terminal time according to Eq. (5.32) were required to reduce Ω to zero. In this section, the numerical solution of a problem very similar to the one in Section 2-5 is discussed In order to compare the ENRM and the NRM, the terminal conditions are of the point type, namely,

$$\Psi(x) = I_{23}x(t_f) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t_f) = 0 \tag{5.72}$$

and the terminal time is considered fixed at $t_f = 5$. Otherwise, the dynamics, Eqs. (2.31) and (2.37) and the performance functional, Eqs. (2.32) and (2.37) are the same as in Section 2-5. The initial condition is taken to be $x^T = (2,0,3)$.

Linearization of the canonical system Eqs. (5.3) - (5.5) and eliminating control results in

$$\left[\begin{array}{c} \dot{x} \\ \hline \dot{p} \end{array}\right] = \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -(1+3\bar{x}_1^2) & -1 & -1 & 0 & 0 & -1 \\ \hline -3+6\bar{x}_1\bar{p}_3 & 0 & 0 & 0 & 0 & (1+3\bar{x}_1^2) \\ 0 & -5 & 0 & -1 & 0 & 1 \\ 0 & 0 & -2 & 0 & -1 & 1 \end{array}\right] \left[\begin{array}{c} x \\ \hline p \end{array}\right] + \left[\begin{array}{c} 0 \\ 0 \\ 2\bar{x}_1^3 \\ \hline -6\bar{x}_1^2\bar{p}_3 \\ 0 \\ 0 \end{array}\right] \tag{5.73}$$

which corresponds to Eqs. (5.15) and (5.16). From Eq. (5.41), the terminal adjoint condition is

$$p(t_f) = I_{32}\nu \tag{5.74}$$

The appropriate Ricatti transformation is given by Eq. (5.30) where the differential equations for the Ricatti coefficients are listed in Table 5.2 with terminal conditions (5.46) ($\hat{x}_f = 0$).

To initiate the ENRM, "starting elements" must be supplied: initial state and adjoint trajectories $\bar{x},\bar{p}$ and an initial $\bar{\nu}$. Quite arbitrarily, $\bar{\nu}_1$ and $\bar{\nu}_2$ were each taken to be unity. Initial trajectories were generated by choosing a reasonable feedback control (programmed control would do equally as well) and integrating the state equations (5.3) forward, determining the terminal condition on $p(t_f)$ from Eq. (5.6) using $\bar{x}(t_f)$ and $\bar{\nu}$, and integrating the adjoint

system (5.4) in reverse time back to the initial time with $x = \bar{x}(t)$. Two initial controls were utilized: the linear Ricatti control

$$u_R = -x_1 - 3x_2 - 2x_3 \tag{5.75}$$

and a control which was deliberately chosen to be poor,

$$u_p = x_1^3 + \frac{20}{27} x_1 \tag{5.76}$$

All numerical integration in these trials employed the procedure described in Appendix C and quadrature was by Simpson's Rule. The integration step size is determined by the requirement for an acceptably small error in the solution, error arising mainly from the per step truncation error of the integration formula. For both the ENRM and the NRM, a convenient measure of integration accuracy exists, namely, the precision with which the linearized terminal conditions are met at each iteration. The linearized terminal conditions would only be satisfied exactly if the numerical integration and intermediate data processing could be performed without error. Thus, the extent to which the linearized terminal conditions are not met is a convenient indication of the truncation error accumulated and propagated each iteration.

Both the ENRM and the NRM with 50 step integration converged to the extremal when the Ricatti control (5.75) was used to start the process. For the particular initial state (2,0,3), the Ricatti control produces a trajectory not too unlike the extremal. In Table 5.3 is listed data

Table 5.3 Progress of the Iterates: Eq. (5.75) as Initial Condition

Iter. No.	Extended Newton-Raphson			Newton-Raphson		
	Trajectory Norm	$x_1(5)$	$x_2(5)$	Trajectory Norm	$x_1(5)$	$x_2(5)$
1	$4.3\ 10^{1}$	$1.1\ 10^{-4}$	$0.3\ 10^{-4}$	$3.8\ 10^{1}$	$2500\ 10^{-4}$	$3500\ 10^{-4}$
2	$8.3\ 10^{-1}$	-.9	1.0	$2.6\ 10^{0}$	-15	-18
3	$1.1\ 10^{-2}$	-1.0	1.2	$1.1\ 10^{-1}$	-2.0	2.4
4	$4.0\ 10^{-5}$	-1.0	1.2	$2.2\ 10^{-3}$	-1.3	-1.6
5	$7.6\ 10^{-6}$	-1.0	1.2	$2.0\ 10^{-4}$	-1.6	-1.8
6	$9.5\ 10^{-6}$	-1.0	1.2	$1.6\ 10^{-3}$	0.8	0.9

which describes the progress of the iterations for both methods. The trajectory norm is given by Eq. (5.29) with w=0. Note that although the state terminal conditions are ultimately met with about equal precision by either method, a comparison of the precision away from the extremal (initially) indicates that the accumulated error due to numerical integration is considerably less in the ENRM than in the NRM.

The computer (IBM 7044) time per iteration is slightly less for the ENRM than for the NRM: 1.4 seconds as opposed to 1.7 seconds. This shorter interval is a result of the fewer number of differential equations to be integrated at each iteration (18 integrations plus 5 quadratures compared to 30 integrations for the NRM). In fact, if a single quadrature is counted as being equivalent to half an integration, the number of integrations per iteration is always less for the ENRM and the difference grows more significant as the

number of terminal conditions (m) decreases.

When control (5.76) was used to obtain a starting trajectory, the ENRM converged again using 50 step integration but the NRM diverged. Because the same TPBVP is being solved at each iteration in both methods, divergence of the NRM could only be caused by excessive integration error. Successive doubling of the number of steps finally resulted in convergence with 400 step integration. The initial and final iterates are displayed in Figs. 5-1 and 5-2 and the progress of the iterations is shown in Table 5.4. Again it can be seen that even with the much smaller step size, the integration error is much larger in the NRM as indicated by the terminal values of the early iterates. The computer time per iterate was 13.2 seconds for the NRM and 1.4 seconds for the ENRM.

This example gives further substantiation of the claim that the differential equations pertinent to the ENRM can be numerically integrated with significantly less accumulation of error than can the linearized canonical system, for the same integration step size.

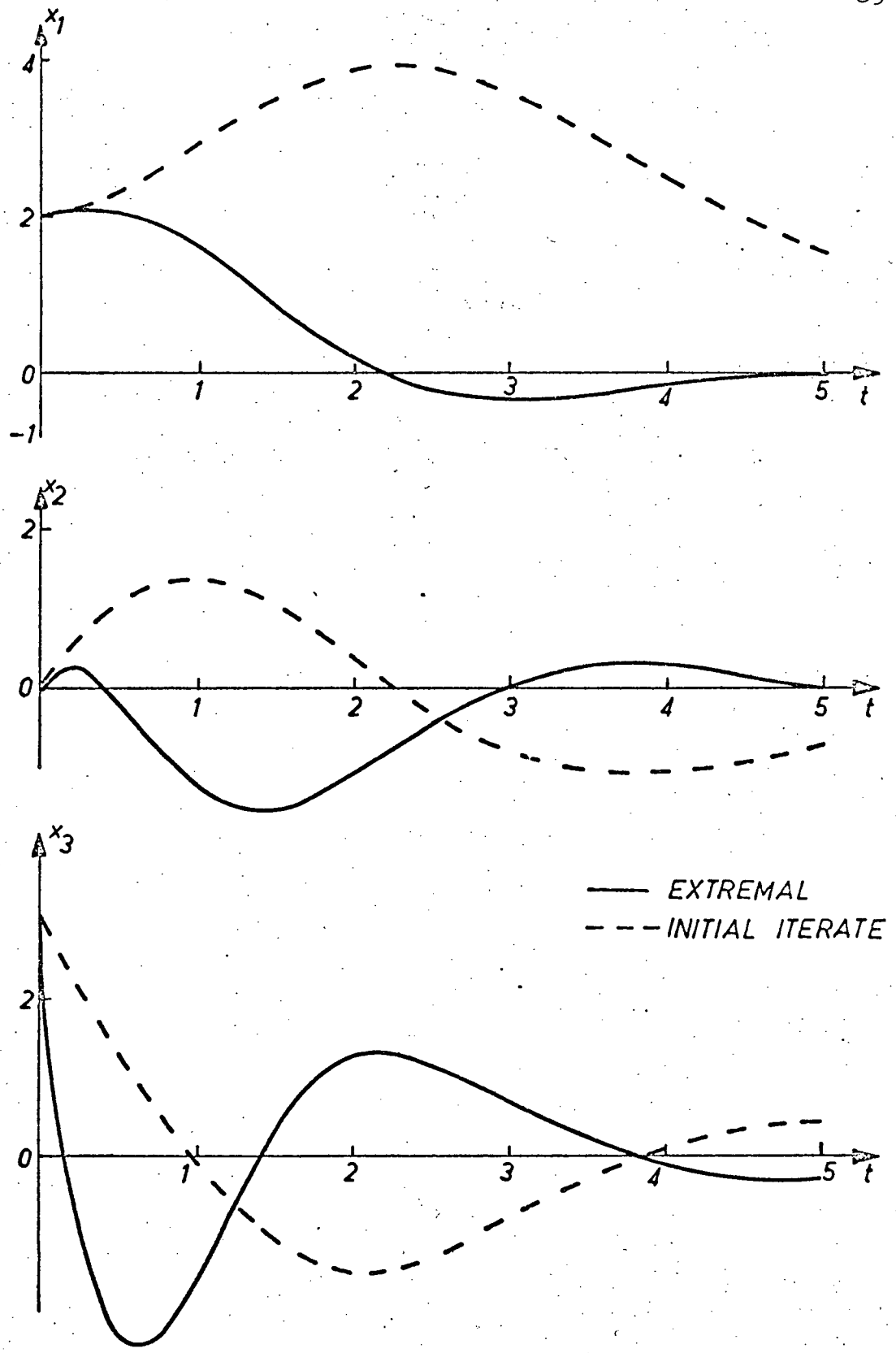

Fig. 5-1 Initial (broken) and Final (solid) State Iterates.

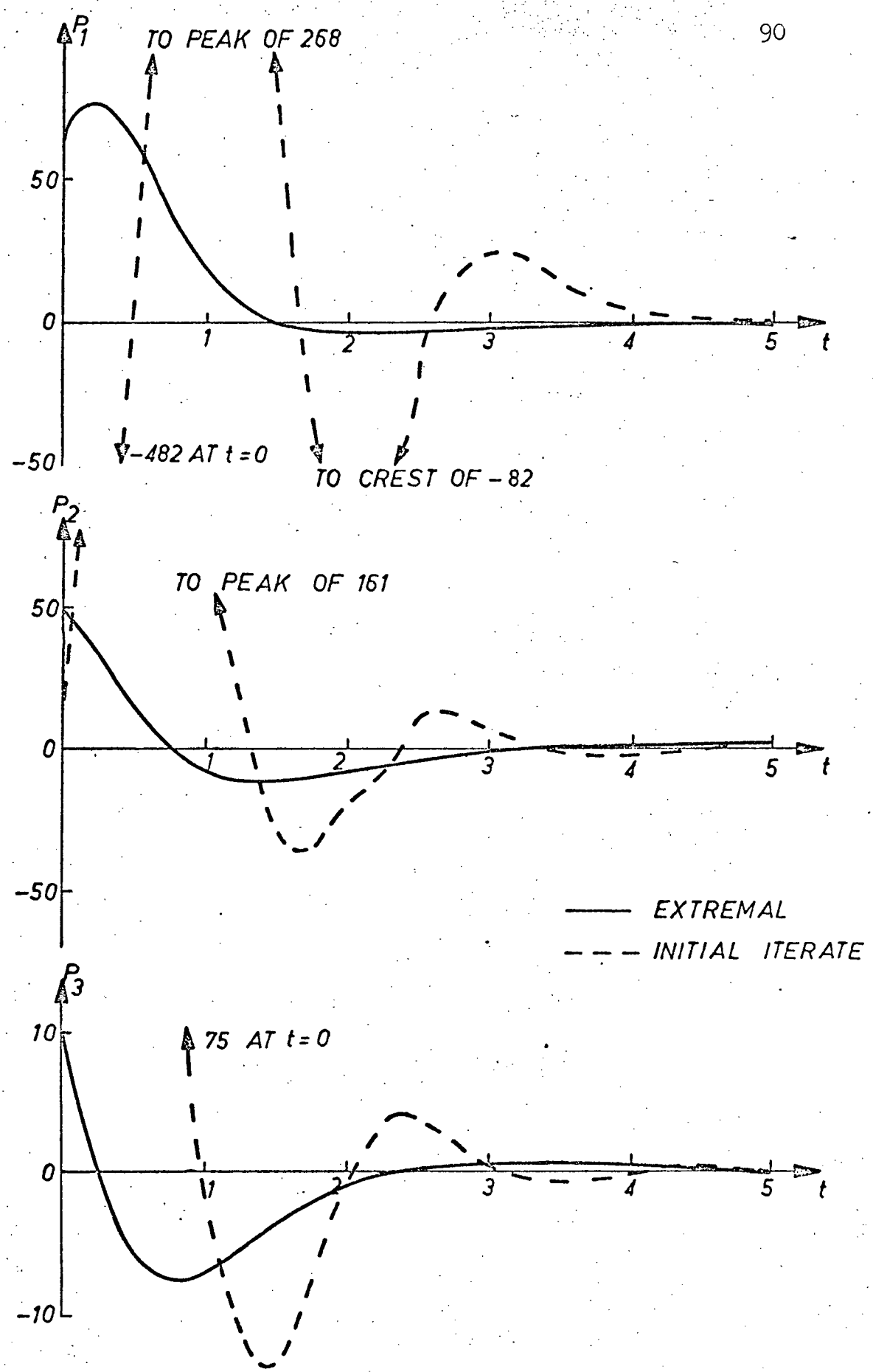

Fig. 5-2 Initial (broken) and Final (solid) Adjoint Iterates.

Table 5.4 Progress of the Iterates: Eq. (5.76) as Initial Control

Iter. No.	Extended Newton-Raphson			Newton-Raphson		
	Trajectory Norm	$x_1(5)$	$x_2(5)$	Trajectory Norm	$x_1(5)$	$x_2(5)$
1	$3.2\ 10^{2}$	$1.3\ 10^{-4}$	$3.5\ 10^{-4}$	$1.8\ 10^{3}$	$-53{,}000\ 10^{-4}$	$-240{,}000\ 10^{-4}$
2	$6.4\ 10^{1}$	-0.9	-1.9	$7.6\ 10^{3}$	1000	4300
3	$5.5\ 10^{1}$	-27	-3.3	$5.5\ 10^{3}$	-1600	-7800
4	$2.5\ 10^{1}$	-0.5	-0.3	$8.9\ 10^{1}$	-400	1700
5	$1.4\ 10^{1}$	-1.1	0.3	$1.1\ 10^{2}$	30	150
6	$5.6\ 10^{0}$	-0.9	0.8	$4.6\ 10^{1}$	-3.9	-13
7	$2.3\ 10^{-1}$	-1.0	1.2	$1.0\ 10^{1}$	19	48
8	$2.4\ 10^{-3}$	-1.0	1.2	$1.1\ 10^{1}$	6.5	9.5
9	$9.5\ 10^{-6}$	-1.0	1.2	$1.6\ 10^{0}$	-6.5	-7.7
10	$5.7\ 10^{-6}$	-1.0	1.2	$7.8\ 10^{-2}$	6.3	7.6
11				$9.2\ 10^{-3}$	-7.3	-8.8
12				$7.8\ 10^{-3}$	4.4	5.3

6. GENERAL THEORY OF SECOND VARIATION METHODS

6-1 Introduction.

The extended Newton-Raphson method presented in Chapter 5 has local convergence properties characteristic of a group of optimization techniques, namely, rapid ("quadratic") convergence in a certain neighborhood of the extremal. In this chapter, an entire class of rapid-convergence numerical optimization algorithms is derived. Algorithms belonging to this class are herein given the name second variation methods.

Many, but not all of the second variation method have already appeared in the literature. The approach taken in this chapter develops these existing techniques from a single, unified point of view allowing the relationships between them to be seen clearly. It is interesting that certain of the established methods have been called "the second variation method", most commonly [37], [38] and occasionally [26]. The term "second variation method" is made precise in this chapter and in so doing, other important methods in addition to the three just cited are shown to belong to the same class of algorithms. These include the "successive sweep method" [33], the function-space Newton-Raphson method [8], and the extended Newton-Raphson method of Chapter 5.

The recognition of a class of second variation methods

is valuable for other reasons as well. A basis is provided for developing new and useful optimization techniques from this class. In addition, computational devices utilized in one of these methods can easily be exploited in the others when their underlying similarities are exposed. For example, the use of the generalized Ricatti transformation in the previous chapter was suggested by an almost identical transformation in McReynolds and Bryson [33] who propose the successive sweep method. Although the advantages have not yet been investigated, the Ricatti transformation can be applied in conjunction with any second variation method.

6-2 First and Second Variations.

The control problem dealt with is again that specified by Eqs. (1.1), (1.2) and (1.3). In order to simplify the ensuing derivations, the terminal functions ϕ and Ψ thus depend only on the terminal value of the state. Again the assumption is made that any inequality constraints on the control or state variables are accounted for by including them as penalty function terms in the performance functional integrand.

Following the variational calculus method of Lagrange multipliers, a time-varying n-vector p and a constant m-vector ν are introduced to form the augmented functional J_a:

$$J_a(u,x,p,\nu,x(t_f)) = \Phi(x(t_f),\nu) + \int_{t_o}^{t_f} (H(x,p,u,t)-p^T\dot{x})dt \tag{6.1}$$

where $\Phi(x(t_f),\nu) = \phi(x(t_f)) + \nu^T\psi(x(t_f))$ (6.2)

$H(x,p,u,t) = F(x,u,t) + p^T f(x,u,t)$ (6.3)

The arguments of J_a are now considered to be independent and subject to no constraints. J_a has a stationary point with respect to its arguments at the same function u for which J(u) is a minimum subject to the dynamical constraints and the terminal constraints.

Let $\bar{u},\bar{x},\bar{p},\bar{\nu},\bar{x}(t_f)$ be nominal values of the arguments of J_a and $\delta u = u - \bar{u}$, $\delta x = x - \bar{x}$, etc. be perturbations from these arguments. Assuming that J_a is twice differentiable [39] and that $\delta x(t_o) = 0$, the change in J_a to second order, derived in Appendix D, is given by

$$\Delta_2 J_a(\delta u, \delta x, \delta p, \delta\nu, \delta x(t_f)) = \delta J_a + \delta^2 J_a \tag{6.4}$$

where

$$\delta J_a = \int_{t_o}^{t_f} (U^T\delta u + P^T\delta x + X^T\delta p)dt + \xi^T\delta\nu + \pi^T\delta x(t_f) \tag{6.5}$$

is the first variation and

$$\delta^2 J_a = \tfrac{1}{2}\delta x(t_f)^T\Phi_{xx}\delta x(t_f) + \delta\nu^T\psi_x\delta x(t_f) + \int_{t_o}^{t_f} (\delta p^T(f_x\delta x + f_u\delta u - \delta\dot{x}) + \tfrac{1}{2}\delta x^T H_{xx}\delta x + \delta u^T H_{ux}\delta x + \tfrac{1}{2}\delta u^T H_{uu}\delta u)dt \tag{6.6}$$

is the second variation. The coefficients appearing in the first variation are defined as follows:

$$U = H_u^T(\bar{x},\bar{p},\bar{u},t) \tag{6.7}$$

$$P = H_x^T(\bar{x},\bar{p},\bar{u},t) + \dot{\bar{p}} \tag{6.8}$$

$$X = f(\bar{x},\bar{u},t) - \dot{\bar{x}} \tag{6.9}$$

$$\xi = \Psi(\bar{x}(t_f)) \tag{6.10}$$

$$\pi = \Phi_x^T(\bar{x}(t_f),\bar{\nu}) - \bar{p}(t_f) \tag{6.11}$$

where the vector functions U,P and X are defined on $[t_o,t_f]$ and ξ and π are constant vectors.

6-3 Second Variation Methods. [40]

It is convenient for the subsequent discussion to define the set

$$\Lambda = \{U,P,X,\xi,\pi\} \tag{6.12}$$

The five vectors which compose the set Λ will be referrred to as the elements of Λ. A necessary condition that a nominal path be optimal is that $\delta J_a = 0$ which requires that the elements of Λ be zero over their domain of definition. The resulting equations are the standard necessary conditions for an optimum. It is desired to construct an iterative process such that the "fixed point" of this process is the set of arguments of J_a which satisfy these necessary conditions.

Regarding the nominal values as values obtained on the $(i-1)^{st}$ iteration, the iterative procedure is established by

adhering to the following steps:

A. Some elements of the set Λ are constrained to be zero over their domain of definition, that is, certain of the necessary conditions are satisfied at each iteration. Let δJ_R denote the remainder of the first variation after imposing the selected constraints.

B. Determine the increments δu, δx, δp, $\delta \nu$, $\delta x(t_f)$ so that $\Delta_2 J_a = \delta J_R + \delta^2 J_a$ is stationary.

C. To satisfy the necessary conditions selected in step A, certain missing parameters and/or functions must be furnished. The new (i^{th}) iterate is obtained by satisfying the constraints imposed in step A, using the increments computed in step B to form the new estimate of any missing parameters of functions.

To clarify step C, consider the case where the imposed constraints are $X = 0$, $P = 0$, $\pi = 0$, that is, each iterate must satisfy

$$\dot{x} = f(x,u,t) \tag{6.13}$$

$$\dot{p} = -H_x^T(x,p,u,t) \tag{6.14}$$

$$p(t_f) = \Phi_x^T(x(t_f),\nu) \tag{6.15}$$

Step C would then involve integrating (6.13) forward with $u^i = u^{i-1} + \delta u$ and (6.14) backward with terminal condition (6.15) where $\nu^i = \nu^{i-1} + \delta \nu$. The increments δu and $\delta \nu$ are determined in step B.

As another example, assume that the constraints $U = 0$, $P = 0$ and $X = 0$ are imposed, that is, each iterate must

satisfy Eqs. (6.13), (6.14) and

$$H_u^T(x,p,u,t) = 0 \tag{6.16}$$

If (6.16) is used to determine u^1, then all that is required for the integration of (6.13) and (6.14) is a new estimate of the missing initial condition $p^i(t_o)$. This is found in step B by evaluating $\delta p(t_o)$.

Iterative computational algorithms which can be derived by following steps A-C above will be defined as <u>second variation methods</u>.

6-3.1 <u>The Auxiliary Minimization Problem</u>.

Step B of the iterative procedure will now be performed for the general case where all the elements of the set Λ are unconstrained. Stated in other terms, all of the standard necessary conditions for a minimum are relaxed. This analysis therefore includes all possible selections of relaxed necessary conditions as special cases. Of course, this most general case is itself a second variation method.

The second-order functional $\Delta_2 J_a$ is rewritten here in a slightly different form:

$$\begin{aligned}\Delta_2 J_a = {} & \pi^T \delta x(t_f) + \tfrac{1}{2}\delta x^T(t_f)\,\Phi_{xx}\,\delta x(t_f) + \delta\nu^T(\xi+\psi_x\,\delta x(t_f)) \\ & + \int_{t_o}^{t_f} \big(\delta p^T(\dot{X} + f_x\delta x + f_u\delta u - \delta\dot{x}) + p^T\delta x + U^T\delta u \\ & \qquad + \tfrac{1}{2}\delta x^T H_{xx}\delta x + \delta u^T H_{ux}\delta x + \tfrac{1}{2}\delta u^T H_{uu}\delta u\big)dt\end{aligned} \tag{6.17}$$

To determine $\delta u, \delta x, \delta p, \delta \nu, \delta x(t_f)$ such that $\Delta_2 J_a$ is stationary, the first variation of $\Delta_2 J_a$ may be taken and set to zero. While this is straightforward, the labor involved may be circumvented by recognizing $\Delta_2 J_a$ as the augmented functional for an auxiliary minimization problem. This axuxiliary problem has a linear differential constraint

$$\delta \dot{x} = f_x \delta x + f_u \delta u + X \tag{6.18}$$

and δp is the Lagrange multiplier associated with this constraint. The functional to be minimized is

$$\Delta_2 J = \pi^T \delta x(t_f) + \tfrac{1}{2} \delta x^T(t_f)\ \Phi_{xx}\, \delta x(t_f) + \int_{t_o}^{t_f} (P^T \delta x + U^T \delta u + \tfrac{1}{2} \delta x^T H_{xx} \delta x + \delta u^T H_{ux} \delta x + \tfrac{1}{2} \delta u^T H_{uu} \delta u) dt \tag{6.19}$$

and the linear terminal constraint is

$$\Psi_x \delta x(t_f) + \xi = 0 \tag{6.20}$$

where $\delta \nu$ is the Lagrange multiplier associated with this constraint.

The necessary conditions for a solution to the auxiliary minimization problem are obtained from Eqs. (6.7) - (6.11) by setting the elements of Λ to zero and accounting for the change in symbolism. This yields Eqs. (6.18), (6.20), and

$$-\delta \dot{p} = H_{xx} \delta x + f_x^T \delta p + H_{xu} \delta u + P \tag{6.21}$$

$$0 = H_{ux}\delta x + f_u^T \delta p + H_{uu}\delta u + U \tag{6.22}$$

$$\delta p(t_f) = \Phi_{xx}\delta x(t_f) + \psi_x^T \delta\nu + \pi \tag{6.23}$$

Equations (6.18), (6.20) - (6.23) represent a linear two-point boundary-value problem (TPBVP). The equations associated with any other second variation method can be obtained directly from Eqs. (6.18), (6.20) - (6.23) by equating the appropriate elements of the set Λ to zero.

6-3.2 Step Size Control.

From the equations for the linear TPBVP, it can be seen that the "size" (in terms of appropriate norms) of the elements of Λ determine how large the perturbations of the nominal values will be (step size). Since these relaxed conditions may be regarded as necessary condition errors, step-size control may be exercised by attempting to correct for only a fraction of the total error in any one step. This device was utilized in the technique of Chapter 5, as well as by others.

6-3.3 Solution of the Auxiliary Problem.

If Eq. (6.22) is solved for δu (assuming H_{uu} to be nonsingular) and substituted into Eqs. (6.18) and (6.21), the following linearized canonical system results:

$$\begin{bmatrix} \delta\dot{x} \\ \delta\dot{p} \end{bmatrix} = \begin{bmatrix} A & B \\ C & -A^T \end{bmatrix} \begin{bmatrix} \delta x \\ \delta p \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \tag{6.24}$$

where $A = f_x - f_u H_{uu}^{-1} H_{ux}$ (6.25)

$$B = -f_u H_{uu}^{-1} f_u^T \tag{6.26}$$

$$C = -H_{xx} + H_{xu} H_{uu}^{-1} H_{ux} \tag{6.27}$$

$$a = X - f_u H_{uu}^{-1} U \tag{6.28}$$

$$b = -P + H_{xu} H_{uu}^{-1} U \tag{6.29}$$

Two basic approaches to the solution of the linear TPBVP (6.20), (6.23), (6.24) can be distinguished. In the first approach, the linearized canonical system or its corresponding adjoint is integrated as a single set of coupled differential equations. This approach was outlined in Section 5-8 in connection with the Newton-Raphson method. The other technique utilizes the generalized Ricatti transformation, as in the extended Newton-Raphson method of Chapter 5. The details of applying these two approaches are slightly different for each second variation method.

6-4 Particular Methods.

The purpose of this section is to relate the definition of second variation methods given in Section 6-3 to some well-known optimization methods. It is claimed that these techniques are second variation methods according to this definition. Because each technique was originally developed separately and from different points of view, a brief indication will now be given of how the three steps in the

definition apply in each case.

6-4.1 Successive Sweep Method. [33]

The necessary conditions satisfied at each iterate are $X = 0$, $P = 0$ and $\pi = 0$. This corresponds to the first example illustrating step C in Section 6-3 and the comments there explain how the results of step B are utilized in satisfying the selected constraints. As has already been stated, a generalized Ricatti transformation was used in solving the auxiliary minimization problem.

The successive sweep method generalizes Merriam's procedure [37] which treated the fixed terminal time, free terminal state problem. Thus, only the condition $U = 0$ is not satisfied in Merriam's method.

6-4.2 Method of Breakwell, Speyer and Bryson. [26]

For this technique, the constraints imposed are $X = 0$, $P = 0$ and $U = 0$ which coincides with the second example in Section 6-3. In essence, all the function constraints are satisfied at each iterate and the adjoint initial condition is iteratively adjusted until the terminal conditions are met. The linearized canonical system is treated in its uncoupled form in solving the linear TPBVP which determines the increment δp_o. For details of this phase, the original reference should be consulted.

Another technique [38], though advocating an approach based on removing the terminal conditions by penalty functions

in the initial phase of the computation, is identical to [26] in the rapidly convergent terminal phase.

6-4.3 Newton-Raphson Methods.

In the Kenneth and McGill algorithm [8], the necessary conditions satisfied at each iterate are given by $U = 0$, $\xi = 0$ and $\pi = 0$ whereas in the extended Newton-Raphson method of Chapter 5 and also in Lewallen [30], only the condition $U = 0$ is enforced when the more general terminal condition is allowed. No further discussion of these methods is given here as they are discussed fully in Chapter 5.

7. CONCLUSIONS AND EXTENSIONS: PART II

7.1 Conclusions.

A technique for solving the nonlinear two-point boundary-value problem arising in dynamic optimization problems was presented in Chapter 5. Basically a function space Newton-Raphson scheme, the method is applied to an optimization problem with more general terminal conditions than the original method [8]. The method of handling terminal conditions is different from either [8] or another work [30] which also treats the general end conditions. A generalized Ricatti transformation is applied to decouple the linearized canonical system, a procedure which considerably enhances the numerical properties of the overall algorithm. This enhancement takes the form of a greatly reduced sensitivity to boundary value parameters and differential equations which can be integrated numerically with less error for a given step size. The new algorithm requires considerably more fast memory since the Ricatti coefficients must be stored. For on-line computation, this is unquestionably a disadvantage but for off-line applications, as in the synthesis technique of Part I for example, it should be unimportant unless the problem is very big or the available memory unusually small.

In Chapter 6, the extended Newton-Raphson method was shown to belong to a class of algorithms called second variation methods. The definition and development of this family of algorithms serves to unify several seemingly diverse optimization techniques and provides a firm basis for further in-

vestigation of computational methods belonging to this class.

7.2 Extensions.

It was assumed in developing the extended Newton-Raphson method that state and control inequality constraints, if any existed, were approximated by suitable penalty terms in the performance functional. The penalty function approach is often a convenient theoretical device but is not without its disadvantages when actual numerical results are desired. A more direct method of handling these constraints is desireable. The case where magnitude limits are imposed on control variables should be investigated first since it is the most common type of inequality constraint and probably the easiest to handle. An approach taken by Kenneth and McGill [29] treats the control bounds as additional algebraic constraints to be satisfied along trajectories and then applies the Newton-Raphson method to the nonlinear problem having algebraic as well as differential constraints. Future work should examine the possibility of applying this technique while continuing to utilize the generalized Ricatti transformation.

An investigation into the advantages and disadvantages of using the generalized Ricatti transformation in conjunction with second variation methods where it has not yet been applied might yield improved algorithms. In particular, the method of Breakwell, Speyer and Bryson outlined in Section 6-4.2 should be examined in this connection. The suitability

of unexploited second variation methods for numerical computation is also a subject for a future research.

APPENDIX A

Approximate Solution of the Hamilton-Jacobi Equation, Example 2.

Let $V(x)$ denote the minimum value of the functional (2.32) starting from the initial state x. A necessary condition for optimality is given by the Hamilton-Jacobi equation (valid for the problem class in Example 2):

$$H(x,V_x^T,u^*(x,V_x^T)) = \min_u H(x,V_x^T,u) = 0 \qquad (A.1)$$

where $$H(x,V_x^T,u) = \tfrac{1}{2}(x^TQx + u^TRu) + V_x f(x,u,t) \qquad (A.2)$$

Hence, for the problem in Section 2-5, the Hamilton-Jacobi equation is:

$$x^TQx + V_x(Fx + f(x)) + (Fx + f(x))^TV_x^T - V_xGG^TV_x^T = 0 \qquad (A.3)$$

and the optimal feedback control is given by

$$u^*(x,V_x^T(x)) = -G^TV_x^T(x) \qquad (A.4)$$

An approximate solution of (A.3) can be obtained by assuming a power series for $V(x)$ and then equating coefficients of like terms to zero. Details may be found in Merriam [35]. Since, for this problem, V must be an even function of x, a power series up to fourth-power terms may be written

$$V(x) \simeq \tfrac{1}{2}x^T\hat{K}x + \hat{V}(x) \qquad (A.5)$$

where $\hat{K}$ is an nxn symmetric matrix and

$$\hat{V}(x) = \sum_{i=0}^{4} \sum_{j=0}^{i} a_k x_1^{4-i} x_2^{(i-j)} x_3^j$$

$$k = \frac{i(i+1)}{2} + j + 1 \qquad (A.6)$$

After computing the partial derivatives, (A.5) is substituted into (A.3) and coefficients of like terms are collected and set to zero. This results in

$$\hat{K} = K \tag{A.7}$$

where K is defined in Eq. (2.38), and fifteen linear algebraic equations for the coefficients $a_k, k = 1,\ldots,15$. The solution of these equations is listed in Table A.1.

Table A.1

a_1	a_4	a_7	a_{10}	a_{13}
a_2	a_5	a_8	a_{11}	a_{14}
a_3	a_6	a_9	a_{12}	a_{15}
0.3228	1.2772	0.9088	0.0132	0.0960
0.5441	0.1456	0.5250	0.2311	0.0199
-0.5000	0.0110	0.1324	0.2272	0.0017

From Eqs. (A.4), (A.5), (A.7) and (2.36), the sub-optimal control is given by

$$u_A(x) = u_R(x) - \frac{\partial \hat{V}}{\partial x_3} \tag{A.8}$$

APPENDIX B

Newton-Raphson Linearization of Equations (5.6) - (5.8).

Treating Eq. (5.6) first, the first-order expansion is

$$\Delta p_f = \bar{\Phi}_x^T - \bar{p}(\bar{t}_f) + \Phi_{xx}\Delta x_f + \psi_x^T \delta\nu + \Phi_{xt}\,\delta t_f \qquad (B.1)$$

where, to first-order,

$$\Delta p_f = \delta p(\bar{t}_f) + \dot{\bar{p}}(\bar{t}_f)\,\delta t_f \qquad (B.2)$$

$$\Delta x_f = \delta x(\bar{t}_f) + \dot{\bar{x}}(\bar{t}_f)\,\delta t_f \qquad (B.3)$$

Substituting (B.2) and (B.3) into (B.1) and solving for $p(\bar{t}_f)$ results in Eq. (5.17).

Similarly for Eq. (5.7),

$$\Psi(x(t_f),t_f) = 0 = \bar{\Psi} + \psi_x\Delta x_f + \psi_t\,\delta t_f \qquad (B.4)$$

which is the same as (5.18) when (B.3) is substituted.

The Newton-Raphson linearization of (5.8) is given by

$$\Omega = 0 = \bar{\Omega} + \Omega_x\Delta x_f + H_p\Delta p_f + H_u\Delta u_f + \psi_t^T\delta\nu + \Omega_t\,\delta t_f \qquad (B.5)$$

The term in Δu_f drops out since $H_u = 0$ at each iteration. Substituting (B.1) into (B.5) yields

$$\Omega = 0 = \bar{\Omega} + H_p(\Phi_x^T - \bar{p}(\bar{t}_f)) + (\Omega_x + H_p\Phi_{xx})\Delta x_f + (\psi_t^T + H_p\psi_x^T)\delta\nu + (\Omega_t + H_p\Phi_{xt})\,\delta t_f \qquad (B.6)$$

If (B.3) is substituted into (B.6) and noting that $H_p = \bar{f}^T$, Eq. (5.19) is obtained.

APPENDIX C

Hamming [36] Numerical Integration Formula.

The differential equation to be integrated is assumed to be of the form

$$\dot{x} = f(x,t) \tag{C.1}$$

Let $t_n = t_o + nh$, $x_n = x(t_n)$, where n is the stage index and h is the step size and let $f_n = f(x_n,t_n)$. Then the formula is:

Predict: $$p_{n+1} = x_{n-3} + \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2}) \tag{C.2}$$

Modify: $$m_{n+1} = p_{n+1} - \frac{112}{121}(p_n - c_n) \tag{C.3}$$

Correct: $$c_{n+1} = \frac{1}{8}(9x_n - x_{n-2} + 3h(f(m_{n+1},t_{n+1}) + 2f_n - f_{n-1})) \tag{C.4}$$

Modify: $$x_{n+1} = c_{n+1} + \frac{9}{121}(p_{n+1} - c_{n+1}) \tag{C.5}$$

Note that only two evaluations of f are required at each stage. The truncation error term is proportional to h^6. To start the integration (first three steps), a fourth-order Runge-Kutta (Gill modified) routine was used. At the fourth stage, which is the first stage after the Runge-Kutta integration, values are not available for p_n and c_n and hence the predicted value cannot be modified.

APPENDIX D

Second-Order Expansion of the Augmented Functional.

The change in J_a, Eq. (6.1), is expanded as follows:

$$\begin{aligned}\Delta J_a = {} & \Phi_x \delta x(t_f) + \Phi_\nu \delta\nu + \tfrac{1}{2}\delta x(t_f)^T \Phi_{xx} \delta x(t_f) \\ & + \delta\nu^T \Phi_{\nu x} \delta x(t_f) + \tfrac{1}{2}\delta\nu^T \Phi_{\nu\nu} \delta\nu \\ & + \int_{t_o}^{t_f} (H_x \delta x + H_p \delta \dot{x} + H_u \delta u + \tfrac{1}{2}\delta x^T H_{xx} \delta x + \delta p^T H_{px} \delta x \\ & + \tfrac{1}{2}\delta p^T H_{pp} \delta p + \delta p^T H_{pu} \delta u + \delta u^T H_{ux} \delta x \\ & + \tfrac{1}{2}\delta u^T H_{uu} \delta u - \delta p^T \dot{\bar{x}} - \bar{p}^T \delta \dot{x} - \delta p^T \delta \dot{x}) dt \\ & + \text{Terms of order 3 or more} \end{aligned} \tag{D.1}$$

Simplifications can be made if the following relationships are recognized:

$$\Phi_{\nu\nu} = 0, \quad H_{pp} = 0, \quad H_p = \bar{f}^T, \quad H_{pu} = f_u$$

$$H_{px} = f_x, \quad \Phi_\nu = \bar{\psi}^T, \quad \Phi_{\nu x} = \psi_x$$

Also, an integration by parts is performed:

$$-\int_{t_o}^{t_f} \bar{p}^T \delta \dot{x} dt = -\bar{p}^T \delta x \Big|_{t_o}^{t_f} + \int_0^{t_f} \dot{\bar{p}}^T \delta x dt$$

Assuming $\delta x(t_o) = 0$, Eq. (D.1) becomes

$$\Delta J_a = (\Phi_x - \bar{p}(t_f)^T)\,\delta x(t_f) + \delta\nu^T\bar{\Psi}$$

$$+ \tfrac{1}{2}\delta x(t_f)^T \Phi_{xx}\,\delta x(t_f) + \delta\nu^T \Psi_x\,\delta x(t_f)$$

$$+ \int_0^{t_f} (H_u\,\delta u + (H_x + \dot{\bar{p}}^T)\,\delta x + \delta p^T(\bar{f} - \dot{\bar{x}}))dt$$

$$+ \int_0^{t_f} (\,\delta p^T(f_x\,\delta x + f_u\,\delta u - \delta\dot{x}) + \tfrac{1}{2}\,\delta x^T H_{xx}\,\delta x$$

$$+ \delta u^T H_{ux}\,\delta x + \tfrac{1}{2}\,\delta u^T H_{uu}\,\delta u)dt + \text{3rd order terms}$$

(D.2)

By definition [39], if J_a is twice differentiable, the first variation can be identified as the linear functional part of ΔJ_a (Eq. (6.5)) and the second variation as the quadratic functional part (Eq. (6.6)).

REFERENCES

1. Berkovitz, L.D., "Variational Methods in Problems of Control and Programming", J. of Math. Analysis and Appl., vol. 3, pp. 145-169, 1961.

2. Zadeh, L.A. and Desoer, C.A., Linear System Theory, McGraw-Hill, New York, 1963.

3. Pontryagin, L.S., Bol'tanskii, V.G., Gamkrelidze, R.S. and Mischenko, E.F., The Mathematical Theory of Optimal Processes, Pergamon, New York, 1964.

4. Athans, M. and Falb, P.L, Optimal Control, McGraw-Hill, New York, 1966.

5. Bellman, R.E., Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1957.

6. Noton, A.R.M., Introduction to Variational Methods in Control Engineering, Pergamon, New York, 1965.

7. Kalman, R.E., "A New Approach to Linear Filtering and Prediction Problems", Trans. ASME, Series D, vol. 82, pp. 35-45, March, 1960.

8. McGill, R. and Kenneth, P., "Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator", American Institute of Aeronautics and Astronautics J., vol. 2, pp. 1761-1766, October, 1964.

9. Longmuir, A.G. and Bohn, E.V., "The Synthesis of Sub-optimal Feedback Control Laws", IEEE Trans. Automatic Control, vol. AC-12, December, 1967.

10. Koivuniemi, A.J., "A Computational Technique for the Design of Specific Optimal Controllers", IEEE Trans. Automatic Control, vol. AC-12, pp. 180-183, April, 1967.

11. Kipiniak, W., Dynamic Optimization and Control, Wiley, New York, 1961.

12. Hamming, R.W., Numerical Methods for Scientists and Engineers, McGraw-Hill, New York, 1962.

13. Fraser, D.A.S., Statistics: An Introduction, Wiley, New York, 1958.

14. Davis, P.J., Interpolation and Approximation, Blaisdell, New York, 1963.

15. Ahlberg, J.H., Nilson, E.N. and Walsh, J.L., The Theory of Splines and Their Applications, Academic Press, New York, 1967.

16. Rice, J.R., The Approximation of Functions, Addison-Wesley, Reading, Mass., 1964.

17. Birkhoff, G. and de Boor, C.R., "Piecewise Polynomial Interpolation and Approximation", in H.L. Garabedian (Ed), Approximation of Functions, Elsevier, Amsterdam, 1965.

18. Longmuir, A.G. and Bohn, E.V., "The Synthesis of Sub-optimal Feedback Control Laws", Preprints of the Joint Automatic Control Conference, Philadelphia, Pa., pp. 492-498, June, 1967.

19. Lorentz, G.G., Approximation of Functions, Holt, Rinehart and Winston, New York, 1966.

20. Bélanger, P.R., "Some Aspects of Control Tolerances and First-Order Sensitivity in Optimal Control Systems", IEEE Trans. Automatic Control, vol. AC-11, pp. 77-83, January, 1966.

21. Rekasius, Z.V., "Optimal Linear Regulators with Incomplete State Feedback", IEEE Trans. Automatic Control, vol. AC-12, pp. 296-299, June, 1967.

22. Smith, F.W., "Design of Quasi-Optimal Minimum-Time Controllers", IEEE Trans. Automatic Control, vol. AC-11, pp. 71-77, January, 1966.

23. Mendel, J.M. and Zapalac, J.J., "Realization of a Suboptimal Controller by Off-Line Training Techniques", Preprints of the Joint Automatic Control Conference, Philadelphia, Pa., pp. 258-266, June, 1967.

24. Gragg, B.B., "Computation of Approximately Optimal Control", SUDAER No. 179, Dept. of Aeronautics and Astronautics, Stanford University, 1964.

25. Durbeck, R.C., "An Approximation Technique for Suboptimal Control", IEEE Trans. Automatic Control, vol. AC-10, pp. 144-149, April, 1965.

26. Breakwell, J.V., Speyer, J.L. and Bryson, Arthur E., "Optimization and Control of Nonlinear Systems Using the Second Variation", S.I.A.M., Journal on Control, vol. 1, pp. 193-223, 1963.

27. Bellman, R.E. and Kalaba, R.E., Quasilinearization and Nonlinear Boundary-Value Problems, Elsevier, Amsterdam, 1965.

28. Kopp, R.E. and Moyer, H.G., "Trajectory Optimization Techniques", in C.T. Leondes (Ed.), Advances in Control Systems, vol. 4, Academic Press, New York, 1966.

29. Kenneth, P. and McGill, R., "Two-Point Boundary-Value-Problem Techniques", in C.T. Leondes (Ed.), Advances in Control Systems, vol. 3, Academic Press, New York, 1966.

30. Lewallen, J.M., "A Modified Quasilinearization Method for Solving Trajectory Optimization Problems", American Institute of Aeronautics and Astronautics J., vol. 5, pp. 962-965, May, 1967.

31. Rothenberger, B.F. and Lapidus, L., "The Control of Nonlinear Systems. Part IV. Quasilinearization as a Numerical Method", American Institute of Chemical Engineers J., vol. 13, pp. 973-981, September, 1967.

32. Schley, C.H. Jr. and Lee, I., "Optimal Control Computation by the Newton-Raphson Method and the Ricatti Transformation", IEEE Trans. Automatic Control, vol. AC-12, pp. 139-144, April, 1967.

33. McReynolds, S.R. and Bryson, A.E., "A Successive Sweep Method for Solving Optimal Programming Problems", Preprints of the Joint Automatic Control Conference, Troy, N.Y., pp. 551-555, June, 1965.

34. Kalman, R.E., "Contributions to the Theory of Optimal Control", Bol. Soc. Mat. Mex., vol. 102, pp. 102-119, 1960.

35. Merriam, C.W., Optimization Theory and the Design of Feedback Control Systems, McGraw-Hill, New York, 1964.

36. Hamming, R.W., "Stable Predictor-Corrector Methods for Ordinary Differential Equations", J. Association for Computing Machinery, vol. 6, pp. 37-47, 1959.

37. Merriam, C.W., "An Algorithm for the Iterative Solution of a Class of Two Point Boundary Value Problems", S.I.A.M. Journal on Control, vol. 2, pp. 1-10, 1964.

38. Kelley, H.J., Kopp, R.E., and Moyer, H.G., "A Trajectory Optimization Technique Based Upon the Theory of the Second Variation", Progress in Astronautics and Aeronautics, vol. 14, chpt. 5, Academic Press, New York, 1964.

39. Gelfand, I.M., and Fomin, S.V., Calculus of Variations, chapter 5, Prentice-Hall, Englewood Cliffs, N.J., 1963.

40. Longmuir, A.G. and Bohn, E.V., "Second Variation Methods in Dynamic Optimization", submitted for publication to the Journal of Optimization Theory and Applications.