IOWA STATE UNIVERSITY
Digital Repository

Retrospective Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

1969

Adaptations of the conjugate gradient method to optimal control problems with terminal state constraints

John Kendall Willoughby
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Engineering Commons

Recommended Citation
Willoughby, John Kendall, "Adaptations of the conjugate gradient method to optimal control problems with terminal state constraints " (1969). *Retrospective Theses and Dissertations*. 4166.
https://lib.dr.iastate.edu/rtd/4166

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

www.manaraa.com

70-13,646

WILLOUGHBY, John Kendall, 1943-
ADAPTATIONS OF THE CONJUGATE GRADIENT METHOD TO OPTIMAL CONTROL PROBLEMS WITH TERMINAL STATE CONSTRAINTS.

Iowa State University, Ph.D., 1969
Engineering, general

University Microfilms, Inc., Ann Arbor, Michigan

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

ADAPTATIONS OF THE CONJUGATE GRADIENT METHOD TO OPTIMAL CONTROL PROBLEMS WITH TERMINAL STATE CONSTRAINTS

by

John Kendall Willoughby

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of

The Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subjects: Aerospace Engineering
Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

Heads of Major Departments

Signature was redacted for privacy.

Dean of Graduate College

Iowa State University
Ames, Iowa

1969

TABLE OF CONTENTS

		Page
CHAPTER I.	INTRODUCTION AND HISTORICAL BACKGROUND	1
CHAPTER II.	THE CONJUGATE GRADIENT METHODS FOR UNCONSTRAINED MINIMIZATION PROBLEMS	10
	Application to Unconstrained Finite-Dimensional Problems	10
	Application to Unconstrained Optimal Control Problems	17
	Numerical Solutions of Unconstrained Optimal Control Problems	24
CHAPTER III.	SOLUTION OF OPTIMAL CONTROL PROBLEMS WITH TERMINAL STATE CONSTRAINTS USING THE CONJUGATE GRADIENT METHOD WITH PENALTY FUNCTIONS	41
	Characteristics of the Penalty Function Method	41
	Numerical Solutions Using Conjugate Gradient Methods With Penalty Functions	44
CHAPTER IV.	SOLUTION OF OPTIMAL CONTROL PROBLEMS WITH TERMINAL STATE CONSTRAINTS USING THE CONJUGATE GRADIENT METHOD WITH A PROJECTION TECHNIQUE	56
	Theoretical Basis of the Projection Method	56
	Application of the Projection Theory to the Conjugate Gradient Method	61
	Numerical Solutions Using the Conjugate Gradient-Projection Method	66
	Extension of the Method to Problems With Nonlinear Terminal Constraints	75
CHAPTER V.	A MODIFIED CONJUGATE GRADIENT METHOD FOR SOLVING CONSTRAINED MINIMIZATION PROBLEMS	79
	Development and Application of the Method in Finite-Dimensional Spaces	79
	Application of the Method to Constrained Optimal Control Problems	89

CHAPTER VI. COMPARATIVE DISCUSSION AND RECOMMENDATIONS FOR ADDITIONAL INVESTIGATION 100

LITERATURE CITED 104

ACKNOWLEDGMENTS 111

APPENDIX A. DERIVATION OF THE AUXILIARY EQUATIONS FOR THE PCG METHOD IN FUNCTION SPACE 112

APPENDIX B 118

CHAPTER I. INTRODUCTION AND HISTORICAL BACKGROUND

The development of the theory and the numerical methods of mathematical optimization continues to be of interest to a wide variety of scientific disciplines. The discovery of new areas of application and the need for solutions to more difficult problems have led to continuing efforts to develop more powerful theories and solution techniques. Of particular interest to the engineer is the branch of optimization usually referred to as optimal control. Although the field has received much specialized attention in recent years, optimal control cannot properly be disassociated from the non-control branches of optimization such as linear programming, nonlinear programming, and the calculus of variations. To the contrary, these non-control branches of optimization theory have contributed heavily to the development of iterative techniques for solving the control problem.

This dissertation treats a particular class of iteration techniques, the conjugate gradient methods. These techniques were originally developed for solving systems of linear algebraic equations, and have recently been extended and used to solve unconstrained optimal control problems. Several proposed modifications which attempt to make the conjugate gradient method applicable to problems with terminal state constraints are examined.

The optimal control problem can be stated imprecisely as the problem of selecting from a specified set of functions that control function which minimizes a given functional and which satisfies specified differential and algebraic constraints involving the problem or state

variables. A more rigorous definition of the optimal control problem will be given in Chapter II. The necessary conditions for optimality of a control function have been derived by many authors including Berkovitz (7) whose work adapts the classical calculus of variations of Bliss (9) to the control problem. Similar necessary conditions have been derived from a geometric viewpoint by Pontryagin et al. (62). The latter work has resulted in the celebrated maximum principle of optimal control.[a] Although these necessary conditions rarely lead to an analytical determination of the optimal control, they form the theoretical foundation upon which the numerical solution techniques are built.

The terms direct and indirect are often used to classify the many numerical techniques that have been proposed. Indirect methods are those that attempt to produce the optimal control by satisfying the necessary conditions for optimality obtained from the calculus of variations or from Pontryagin's maximum principle. In general, the application of these necessary conditions leads to a nonlinear two-point boundary value problem. As a result, most indirect methods are characterized by an iterative modification of either the boundary conditions or the differential equations.

In contrast, direct methods are those that select successive trial control functions based on information obtained from the value of the functional and perhaps its variations for previous control choices.

---

[a]Pontryagin's Principle is also referred to as the minimum principle by many authors.

These methods usually require the choice of an initial control function which is used to determine a direction of search in the space of allowable controls. The control change is the product of the direction of search and a scalar parameter called the stepsize. From the new control, a new direction of search is determined, and the process is repeated. The various direct methods differ principally in the means used to determine the successive directions of search and the magnitude of the control correction taken in those directions.

A method of numerical optimization that is not easily classified as direct or indirect has been derived by Bellman (5,6). The method, known as dynamic programming, views the optimal control problem as a multi-stage decision process. By using the principle of optimality, dynamic programming reduces the problem to a sequence of single-stage decision processes or single variable minimizations. The method is highly compatible with repetitive digital computer techniques. An additional advantage is that constraints simplify rather than complicate, the solution process. Unfortunately the systematic simplicity of the method is often outweighed by its enormous storage requirements. Many optimal control problems, when cast in a form suitable for dynamic programming, require so much computer storage that solution by that method becomes infeasible.

The conjugate gradient methods are direct solution methods. Since this dissertation deals with the development and modification of the conjugate gradient logic, it is instructive to examine the evolution of related direct methods. The development of the conjugate gradient method as a tool for solving optimal control problems is currently paralleling

that of older direct methods.

Steepest descent is perhaps the oldest direct method of minimizing an objective function of several variables. According to Curry (14), an account of the method was given by Cauchy in 1847 and by Hadamard in 1907 who named it the "method of gradients". The technique is based on the simple principle of choosing a trial solution that lies along the direction of maximum decrease of the objective function from the previous trial. It is intuitively clear that if very small steps are taken, each being in the direction of steepest descent from the previous point, the rate of decrease of the objective function approaches a maximum. Arrow and Solow (2) take this approach to the steepest descent method by considering the limiting condition of infinitesimal stepsizes, or equivalently, of a continual and instantaneous readjustment of the direction of search. However, if the principle of steepest descent is to be used as a method of minimizing a function, very small stepsizes are impractical and inefficient. Curry and others suggested that from each point in the search, the negative gradient direction be followed to the one-dimensional minimum of the objecting function. Such a procedure is often called optimum steepest descent and will be referred to by that name in what follows. This method requires a mechanism for locating the minimum along each direction of search. However with that procedure implemented, optimum steepest descent becomes a useful and reasonably powerful computational method (1,53). It should be noted at this point however, that if finite steps are taken, the negative gradient directions may not be the best directions of search that can be chosen since they depend only upon the local nature of the objective function and not

upon its nature at previous search points.

The use of steepest descent for solving optimal control problems requires an extension of the method to function space. This extension was done by Bryson et al. (12,13) and by Kelley (41,42,44). In addition, these authors and others have derived methods of making steepest descent an effective tool for solving control problems involving terminal state constraints, state-space constraints, and control variable constraints. Developers of the method have used both a penalty function approach and Rosen's gradient projection method (64,65) in applying steepest descent to constrained problems. Because of the refinements made to the original unconstrained versions of steepest descent, the method is now applicable to a very broad class of control problems and often can be used to solve those problems which cannot be solved by other methods or to which other methods do not apply. As a result, steepest descent is a popular technique among practicing engineers.

Second-order direct methods of solving optimal control problems have been developed by Breakwell et al. (11), Kelley et al. (44), McReynolds (54), Mayne (51), Jacobson (38) and others. These techniques are extensions of Newton's method for minimizing a function of several variables. It is easily shown that if finite steps are taken, a quadratic function of a finite number of variables can be minimized in one step if the direction of search is taken to be the negative gradient direction premultiplied by the inverse Hessian matrix (the matrix of second partial derivatives of the objective function). Since any function of class C^3 can be expanded in a Taylor's series about its minimum, a quadratic approximation is

valid in some neighborhood of the minimum. If the objective function is globally convex, the inverse Hessian matrices evaluated at the search points can be used to calculate directions of search which lead to faster convergence rates than those obtained from gradient information alone. The improved convergence rates result from the second-order terms that are retained in Newton's method but are disregarded in the steepest descent method. As expected, the superior performance is not achieved without cost. Newton's method requires the evaluation of the Hessian matrix, a task that for complicated functions of several variables is very time consuming. In terms of function evaluations, Newton's method of minimizing a function of n variables requires $\frac{n}{2}(n+1)$ evaluations for the Hessian matrix plus n evaluations of the gradient components at each step. In constrast, steepest descent requires only n gradient component evaluations. In addition, if the inverse Hessian matrix is not positive definite everywhere in the search space, Newton's method may not converge at all. In spite of these difficulties however, Newton's method produces convergence rates that are attractive enough to have led to its extension to function space and thus to its application to optimal control problems.

McGill and Kenneth (52) have developed an indirect second-order technique called quasilinearization. This method solves the two-point boundary value problem obtained from the necessary conditions for optimality by choosing iterates that satisfy the boundary conditions exactly and that approach satisfaction of the differential equations as the iteration proceeds. Another second-order indirect method is called the neighboring extremal method (53). It differs from quasilinearization

in that the differential equations are satisfied exactly at each step, and the boundary conditions are satisfied iteratively.

Although all second-order methods demonstrate rapid convergence near the minimum, they require greater computational effort than do the first-order techniques, and in addition, they may not converge at all from starting iterates that are "far" from the minimum (53). Computational techniques that possess the efficiency of first-order methods but exhibit convergence properties approaching those of the second-order methods are currently of great interest. Several methods are under development or refinement for use on optimal control problems. These methods, like the first and second-order techniques, have their origins in analogous methods for minimizing unconstrained functions of several variables in a finite-dimensional vector space. A class of numerical techniques called conjugate direction methods combines the computational simplicity of the gradient techniques with the rapid convergence properties typical of second-order techniques. These methods do not require the computation of second-order partial derivatives in determining the directions of search. Basically, the improved directions of search are a result of the assumption that the objective function can be approximated by a quadratic function in the neighborhood of the current search point. The properties of the quadratic function are used implicitly in the derivation of the methods to produce directions of search that are superior to the negative gradient directions.

In 1952, Hestenes and Stiefel (32) published the conjugate gradient method as a means of solving a system of linear algebraic equations. The technique was used by Fletcher and Reeves (26) in 1964 to minimize a

function of several variables, or equivalently, to solve a set of non-linear equations.

In 1959, Davidon (17) published another conjugate direction method that he called the variable metric method but which is often referred to by his name. Davidon's method, when applied to a quadratic function, sequentially constructs a matrix which converges to the inverse Hessian matrix. The directions of search chosen are the negative gradient directions premultiplied by the Davidon weighting matrix. The parallel between this and Newton's method is obvious. With the exception of the first, each direction of search is a particular linear combination of the current gradient and the previous direction of search. Thus past gradient information is accumulated as the search proceeds.

In 1963, Fletcher and Powell (25) improved the original formulation of Davidon's method and published computational results. Both the conjugate gradient (CG) technique and Davidon's method have been the subjects of many recent articles. Beckman (4) presented an explanation of the CG method that is based on generalized orthogonalization of successive gradient vectors. A descriptive discussion of the theoretical basis of the CG method is given by Antosiewicz and Rheinboldt (1). Important relationships between the CG method and Davidon's method have been derived by Myers (57) who shows that the same directions of search are generated for quadratic functions if Davidon's method is started using the identity matrix. Mehra (55) gives a method of estimating the inverse Hessian matrix from the sequence of gradients and directions of search obtained using the CG method.

As in the development of steepest descent and Newton's method, the CG method has been generalized to apply to functionals on a suitable function space. Pierson (59) has solved optimal control problems by applying the finite-dimensional CG method to discrete approximations to the continuous control problem. The first extension of the method to a Hilbert Space was presented by Hayes (30) in 1954. Other treatments of the extension have been given by Daniel (15,16), Varaiya (74), Mehra and Bryson (56), Lasdon et al. (50), Sinnott and Luenberger (69), and Pagurek and Woodside (58). The contributions of many of these authors will be discussed in greater detail in later chapters. Tripathis and Narendra (72) and Horwitz and Sarachik (35) have treated Davidon's method in function space.

The generalization of the CG method to most optimal control problems requires a means of handling constraint relations which involve the state variables at the terminal time. Constraints of this type are referred to here as terminal state constraints. Lasdon et al. (50) and Mehra and Bryson (56) have suggested the use of penalty functions for treating terminal state constraints with the CG method. Sinnott and Luenberger (69) have used a projection method on problems with linear terminal state constraints. However, these techniques have met with only partial success or have applied to a limited class of problems. This dissertation attempts to expand the knowledge concerning the applicability of the CG method to control problems with terminal state constraints and broadens the class of problems that have been solved by the method.

CHAPTER II. THE CONJUGATE GRADIENT METHODS FOR UNCONSTRAINED MINIMIZATION PROBLEMS

Application to Unconstrained Finite-Dimensional Problems

The theoretical basis and the computational efficiency of the conjugate gradient method are most apparent from an examination of the finite-dimensional version from which the function space extensions have been derived. The approach taken here is to present first the algorithm itself so that the sequence of calculations is clear from the onset and then to move to a discussion of the formulae involved. The method is discussed in the context of minimizing a function f of n real variables which are elements of a real Euclidean vector space E_n. In this and other chapters, the superscript $*$ is used to indicate the value of the variable at the minimum, i.e.

$$f(\underline{x}^*) \leq f(\underline{x}) \qquad \forall \ \underline{x} \in E_n \tag{II-1}$$

It is assumed for simplicity that only one minimum of f exists over E_n, since most numerical algorithms can at best reach a relative minimum.

The solution procedure involves choosing a new trial vector $\underline{x}_{i+1}$ using the relation

$$\underline{x}_{i+1} = \underline{x}_i + \alpha_i \underline{s}_i \tag{II-2}$$

where the subscript i represents the iteration number, α_i, is a scalar called the stepsize, and $\underline{s}_i$ is an n-vector called the direction of search. Specifically, the CG procedure is as follows:

1. For $i=0$, guess an initial state vector $\underline{x}_0$
2. Calculate the gradient vector $\underline{g}_i$ at $\underline{x}_i$

$$\underline{g}_i = \underline{g}(\underline{x}_i) = \nabla f(\underline{x}_i) \tag{II-3}$$

3. Calculate the CG parameter β_i

$$\beta_i = \frac{\langle \underline{g}_i, N \underline{s}_{i-1} \rangle}{\langle \underline{s}_{i-1}, N \underline{s}_{i-1} \rangle} \tag{II-4}$$

$\langle \underline{y}, \underline{z} \rangle$ denotes the Euclidean inner product defined to be

$$\langle \underline{y}, \underline{z} \rangle = \sum_{j=1}^{n} y_j z_j = \underline{y}^T \underline{z}, \tag{II-5}$$

and N is the Hessian matrix defined by

$$N = \left[\frac{\partial^2 f}{\partial \underline{x}^2} \right]_{\underline{x} = \underline{x}_i} \tag{II-6}$$

If $i=0$, $\beta_0 = 0$.

4. Calculate the direction of search $\underline{s}_i$

$$\underline{s}_i = -\underline{g}_i + \beta_i \underline{s}_{i-1} \tag{II-7}$$

5. Perform a one-dimensional minimization to determine $\underline{x}_{i+1}$ i.e.

$$\underline{x}_{i+1} = \underline{x}_i + \alpha_i \underline{s}_i \tag{II-8}$$

where α_i is such that

$$f(\underline{x}_i + \alpha_i \underline{s}_i) \leqslant f(\underline{x}_i + \gamma \underline{s}_i) \quad \forall \ \gamma > 0 \tag{II-9}$$

6. Increase i and repeat from step 2 until the minimum is reached.

The above procedure is quadratically convergent meaning that it will find the minimum of any quadratic function in a finite number of steps. In particular, the CG method will minimize a quadratic function of n variables in at most n steps (32).

The derivation of the method requires the notion of conjugacy between

vectors. Two vectors $\underline{v}$ and $\underline{w}$ are said to be conjugate, N-conjugate, or N-orthogonal with respect to the matrix N if

$$\langle \underline{v}, N\underline{w} \rangle = 0 . \tag{II-10}$$

If the objective function is quadratic so that

$$f(\underline{x}) = f(\underline{x}^*) + \frac{1}{2} \langle (\underline{x}-\underline{x}^*), N(\underline{x}-\underline{x}^*) \rangle \tag{II-11}$$

and N is a positive definite matrix with constant elements corresponding to the second partial derivatives of f, then the directions of search given by II-7 form a mutually conjugate set with respect to N, i.e.

$$\langle \underline{s}_i, N\underline{s}_j \rangle = 0 , \qquad i \neq j \tag{II-12}$$

It follows that the $\underline{s}_i$ are linearly independent vectors which span E_n. Therefore

$$\underline{x}^* = \sum_{k=0}^{n-1} c_k \underline{s}_k . \tag{II-13}$$

The objective is to determine the coefficients c_k in II-13. Forming the inner product $\langle N\underline{x}^*, \underline{s}_k \rangle$, we see that

$$\langle N\underline{x}^*, \underline{s}_k \rangle = c_k \langle \underline{s}_k, N\underline{s}_k \rangle \qquad k = 0, 1, 2, \cdots n-1 \tag{II-14}$$

since II-12 eliminates all the terms with mixed subscripts. Thus

$$c_k = \frac{\langle N\underline{x}^*, \underline{s}_k \rangle}{\langle \underline{s}_k, N\underline{s}_k \rangle} \tag{II-15}$$

But

$$\nabla f(\underline{x}_i) = N(\underline{x}_i - \underline{x}^*) = \underline{g}_i \tag{II-16}$$

so that

$$c_k = \frac{\langle N \underline{x}_i, \underline{s}_k \rangle - \langle \underline{g}_i, \underline{s}_k \rangle}{\langle \underline{s}_k, N \underline{s}_k \rangle} \tag{II-17}$$

Using II-8 repetitively results in

$$\underline{x}_i = \underline{x}_j + \sum_{k=j}^{i-1} \alpha_k \underline{s}_k , \qquad 0 \leq j \leq i \tag{II-18}$$

From II-16 and II-18, we have that

$$\underline{g}_i = N \underline{x}_j - \sum_{k=j}^{i-1} N \alpha_k \underline{s}_k - N \underline{x}^* \tag{II-19}$$

$$= \underline{g}_j - \sum_{k=j}^{i-1} N \alpha_k \underline{s}_k , \qquad 0 \leq j \leq i \tag{II-20}$$

Also

$$\langle \underline{g}_j , \underline{s}_{j-1} \rangle = 0 \quad , \quad i \leq j \tag{II-21}$$

as a result of the one-dimensional minimization in II-8. Therefore, if the inner product $\langle \underline{g}_i , \underline{s}_{j-1} \rangle$ is calculated from II-20, we have

$$\langle \underline{g}_i , \underline{s}_{j-1} \rangle = \langle \underline{g}_j , \underline{s}_{j-1} \rangle - \sum_{k=j}^{i-1} \alpha_k \langle N \underline{s}_k , \underline{s}_{j-1} \rangle \qquad 1 \leq j \leq i . \tag{II-22}$$

The first term is zero from II-21 and the last term is zero from II-12. Therefore

$$\langle \underline{g}_i , \underline{s}_{j-1} \rangle = 0 \qquad 1 \leq j \leq i \tag{II-23}$$

i.e. the gradient at the ith iteration of the search is orthogonal to all the previous directions of search. Returning to II-17, if $k = i-1$ then

$$c_i = \frac{\langle N x_{i+1}, s_i \rangle}{\langle s_i, N s_i \rangle} = \frac{\langle N(x_i + \alpha_i s_i), s_i \rangle}{\langle s_i, N s_i \rangle} \quad i = 1, 2, \cdots n-1 \quad \text{(II-24)}$$

Thus if s_i is chosen N-conjugate to all previous directions of search, and the stepsize α_i is found using a one-dimensional minimization, the value of c_i is determined from II-24. It is clear that after at most n steps, all of the coefficients in II-13 are determined and the minimum is located. Convergence can be completed in less than n steps if the ith iterate x_i minimizes f over a q-dimensional subspace where $q > i$ and $i < n$. For non-quadratic functions, the rate of convergence of the method depends upon the nature of f, and the location of x_0.

The previous results apply to all methods that generate mutually conjugate directions of search. Davidon's method (17) is another conjugate direction method that is quadratically convergent. The method of constructing the sequence of conjugate directions of search from values of the function and its derivatives at the search points distinguishes between different conjugate direction techniques.

Beckman (4) has shown that the CG method determines the directions of search by a process that is equivalent to a generalized Gram-Schmidt orthogonalization of successive gradients. Each new direction of search is determined once the gradient at the current search point is known. That gradient depends of course upon the previous direction of search. Thus as the iteration proceeds, information about the objective function at all previous search points is used to determine new directions of search. It is the this accumulation of information, i.e. the dependence of the successive directions of search, that accounts for convergence that is

superior to that obtained by steepest descent. The latter uses only current gradient information.

Many useful relationships exist between the gradients and the directions of search at various steps of the CG iteration. These are given in (32) along with their derivations. An important simplification of II-4 results from the inner product of the direction of search with the gradient. Using Equations II-20, II-12, and II-7, we have

$$\langle s_i , g_i \rangle = \langle s_i , g_{i-1} \rangle - \alpha_i \cancelto{0}{\langle s_i , N s_{i-1} \rangle}$$

$$= - \langle g_i , g_{i-1} \rangle + \beta_i \langle s_{i-1} , g_{i-1} \rangle . \qquad \text{(II-25)}$$

But from Equations II-23 and II-7

$$\langle g_i , s_{i-1} \rangle = 0 = - \langle g_i , g_{i-1} \rangle + \beta_i \cancelto{0}{\langle g_i , s_{i-2} \rangle} \qquad \text{(II-26)}$$

or

$$\langle g_i , g_{i-1} \rangle = 0 . \qquad \text{(II-27)}$$

Therefore from II-25

$$\beta_i = \frac{\langle s_i , g_i \rangle}{\langle s_{i-1} , g_{i-1} \rangle} \qquad \text{(II-28)}$$

Replacing s_i using II-7 results in

$$\beta_i = \frac{- \langle g_i , g_i \rangle + \beta_i \cancelto{0}{\langle s_{i-1} , g_i \rangle}}{- \langle g_{i-1} , g_{i-1} \rangle + \beta_{i-1} \cancelto{0}{\langle s_{i-2} , g_{i-1} \rangle}} \qquad \text{(II-29)}$$

$$= \frac{\langle g_i , g_i \rangle}{\langle g_{i-1} , g_{i-1} \rangle} . \qquad \text{(II-30)}$$

This result is valid only when the objective function is quadratic. The use of II-30 instead of II-4 makes it unnecessary to evaluate the Hessian matrix at each step, and thus the CG method requires only the same first-order information that steepest descent requires. Both methods of determining β_i are used in the literature. Daniel (16) and Kelley and Myers (45) present comparisons of the two methods on finite-dimensional problems. However, very little discussion of a comparative nature is reported for the function space extension of the CG method. Since the purpose of this thesis is an examination of the CG method as applied to continuous control problems, a more complete discussion of the alternate formulae for determining β_i is deferred until the continuous problem is considered.

The convergence of the CG method has been considered by several authors and therefore is not the subject of extensive treatment here. Antosiewicz and Rheinboldt (1) present a convergence proof based on the "method of expanding subspaces". They show that at the jth iteration $(j<n)$ the directions of search $\underline{s}_0, \underline{s}_1, \cdots \underline{s}_{j-1}$ generated by application of the algorithm to a quadratic objective function span a j-dimensional subspace over which the function has been minimized. If the minimum is an element of an n-dimensional space, convergence is theoretically completed in at most n steps. Daniel (15) has derived an error estimate that is superior to the best known estimate for steepest descent. The treatments given in both the cited references are sufficiently general to apply to Hilbert space extensions of the CG method. Before presenting this generalization however, it is necessary to give an explicit statement of the optimal control problem.

Application to Unconstrained Optimal Control Problems

A rigorous and formal definition of a general optimal control problem is given by Athans and Falb (3, pp. 191-194). To a large extent, their nomenclature and their definitions are adopted here. A problem formulation that is sufficiently general for the purposes of this dissertation is the following.

From a set of admissible controls, find the control $\underline{u}^*(t)$ that minimizes the functional

$$J(\underline{u}) = \Phi(\underline{x}(t_0), t_0, \underline{x}(t_f), t_f) + \int_{t_0}^{t_f} F(\underline{x}, \underline{u}, t)\,dt \qquad \text{(II-31)}$$

subject to the differential constraints

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, t) \quad , \quad \underline{x}(t_0) = \underline{x}_0 \qquad \text{(II-32)}$$

and the terminal constraints or boundary conditions

$$\underline{\Omega}(\underline{x}(t_f), t_f) = 0 \,. \qquad \text{(II-33)}$$

In the above, $\underline{x}$ is an n-vector of state variables to be controlled, $\underline{f}$ is an n-vector of nonlinear expressions defining the dynamical system (see 3, pp. 163-168) to be controlled, $\underline{u}$ is an m-vector of control functions on the interval $[t_0, t_f]$, $\underline{\Omega}$ is a p-vector of linear or nonlinear expressions constraining the terminal conditions of the dynamical system where $p \leq n+1$, F is a scalar function, and t_0, t_f are the initial and final times which may or may not be specified. The cost functional II-31 is in the Bolza form. However if $\Phi = 0$, the functional takes the Lagrange form, and if $F = 0$, it takes the Mayer form.

The term admissible requires that the components of a control vector

$\underline{u}$ at any given time $t \in [t_o, t_f]$ be chosen from a convex set of real m-tuples and that they are piecewise continuous functions of time on that interval. The set of admissible controls could be specified as a closed and bounded set for all times t, a definition that includes problems with bounded controls or control variable constraints.

The possibility of inequality or equality constraints involving the state variables for $t_o < t < t_f$ is not included in the problem statement given here because, except by using penalty functions, the CG method has not been extended to that class of problems and is not attempted here.

The functions ϕ, F, $\underline{f}$, and $\underline{\Omega}$, are considered to be real-valued functions, and thus J is a real valued functional mapping $\underline{u}(t)$ to the real line. It is convenient to form the Hamiltonian function defined as

$$H(\underline{x}, \underline{u}, \underline{\lambda}, t) = F + \underline{\lambda}^T \underline{f} \tag{II-34}$$

where $\underline{\lambda} = \underline{\lambda}(t)$ is an n-vector of real adjoint or costate variables on $[t_o, t_f]$, and the superscript T indicates the transpose.

Although the necessary conditions for optimality can be derived under weaker regularity requirements (62), F, $\underline{f}$, ϕ, and $\underline{\Omega}$ will be assumed to be continuous and possess continuous first and second partial derivatives with respect to all their arguments (class C^2) since quadratic expansions of $J(\underline{u})$ will be needed. In addition, it will be assumed that the Hamiltonian H is of class C^2 with respect to its arguments (50).

The Bolza form of the cost functional J has been chosen in the original formulation. However, certain techniques of numerical optimization are more easily derived or applied when the Lagrange or Mayer form is used. Simple methods exist for transforming any one of the three forms

into either of the other two. These transformations produce problems with the same solutions but may alter significantly the ease or difficulty with which the solution is obtained using a particular solution method.

The question of existence and uniqueness of an optimal control is avoided here as in most treatments of numerical techniques by assuming that the optimal control problems to be solved by the methods are 'well posed' in the sense that they possess unique solutions. It should be stressed that all solution techniques that make use of any of the necessary conditions for optimality apply only to problems for which solutions exist since the derivation of the necessary conditions presupposes the existence of an optimal solution.

In this thesis, application of the CG method to optimal control problems is done under one additional restriction to the definition previously presented. It is assumed that the optimal control is an element of the space containing all piecewise continuous functions that are elements of an unbounded set. The optimality condition $\frac{\partial H}{\partial \underline{u}}\Big|_{\underline{u}=\underline{u}^*}=0$ is not given by Pontryagin et al. (62) as a necessary condition since the maximum principle is derived for a closed control space. However, for problems where the Hamiltonian is of class C^1 with respect to its arguments and the control space is unbounded, relative minima of J will occur when $\underline{u}^*(t)$ satisfies

$$\frac{\partial H}{\partial \underline{u}}\Big|_{\underline{u}=\underline{u}^*} = 0 \qquad \text{(II-35)}$$

and the other optimality conditions (7). Recently Pagurek and Woodside (58) have reported success with a computational modification to the

logic which applies to bounded control problems. However, the modifications necessitated by terminal state constraints are studied here under the assumption that all piecewise continuous controls are admissible.

Lasdon et al. (50) extend the CG method by considering all controls that are elements of a Hilbert space $\mathcal{H}$ with the inner product

$$\langle \underline{\ell}(t), \underline{m}(t) \rangle = \int_{t_0}^{t_f} \underline{\ell}^T(t)\, \underline{m}(t)\, dt \tag{II-36}$$

and the associated ℓ_2 norm

$$\| \underline{\ell}(t) \|^2 = \langle \underline{\ell}(t), \underline{\ell}(t) \rangle \quad . \tag{II-37}$$

The necessary conditions for optimality given by Pontryagin's Maximum Principle and the Weierstrass condition of the calculus of variations indicate that finding the minimum of the functional $J(\underline{u})$ is equivalent to minimizing the Hamiltonian function defined by Equation II-34 over the set of admissible controls. The minimization must be done, however, subject to the constraints that the state Equations II-32 and the following costate equations are satisfied for each iteration.

$$\dot{\underline{\lambda}} = -\frac{\partial H}{\partial \underline{x}} \tag{II-38}$$

$$\underline{\lambda}(t_f) = \left.\frac{\partial \Phi}{\partial \underline{x}}\right|_{t_f} + \left(\left.\frac{\partial \underline{\Omega}}{\partial \underline{x}}\right|_{t_f}\right)^T \underline{\mu} \tag{II-39}$$

where $\underline{\mu}$ is a p-vector of constant Lagrange multipliers. Lasdon et al. (50) consider only problems without terminal state constraints. For problems of that type, the boundary conditions on the costates do not involve $\underline{\Omega}$. In addition, they consider only problems having fixed initial and final times and problems having scalar control. These assumptions are not particularly restrictive, but each simplifies the necessary conditions

and makes the logic of the iteration process more transparent. The control problem to which Lasdon applies the CG method is now restated.

$$\text{minimize} \quad J = \Phi(\underline{x}(t_f)) \tag{II-40}$$

$$\text{subject to} \quad \dot{\underline{x}} = f(\underline{x}, u, t) \tag{II-41}$$

$$\underline{x}(t_o) = \underline{c} \tag{II-42}$$

The conditions that are necessary for the optimality of a control $u(t)$ are:

$$\dot{\underline{x}} = f(\underline{x}, u, t) \tag{II-43}$$

$$\underline{x}(t_o) = \underline{c} \tag{II-44}$$

$$\dot{\underline{\lambda}} = -\frac{\partial H}{\partial \underline{x}} = -\frac{\partial \underline{f}}{\partial \underline{x}}^T \underline{\lambda} \tag{II-45}$$

$$\underline{\lambda}(t_f) = \frac{\partial \Phi}{\partial \underline{x}}(t_f) \tag{II-46}$$

$$g(u^*) = \left.\frac{\partial H}{\partial u}\right|_{u=u^*} = 0 \tag{II-47}$$

It should be emphasized that the search of the Hilbert space of controls for the optimal control $u^*(t)$ is restricted to the controls satisfying the conditions II-41, II-42, II-43, II-45, and II-46. A discussion of this fact is given in Reference (35). The condition II-47 holds only at the minimum of $J(u)$ i.e. when $J(u) = J(u^*) = \min_u J$. The expression

$$g(u) = \frac{\partial H}{\partial u} \tag{II-48}$$

is the gradient to the Hamiltonian and points in the 'direction' of increasing J. This can be seen by examining the first variation in J

given by

$$\delta J = \frac{\partial \Phi}{\partial \underline{x}}\Big|_{t_f} \delta \underline{x}_f + \int_{t_o}^{t_f} \delta F \, dt \, . \tag{II-49}$$

The notation δJ represents the first-order approximation to $J(u) - J(\hat{u})$ where $\hat{u}$ is a given nominal control function. Using the definition of the Hamiltonian from II-34 and requiring the satisfaction of the state Equations II-32 results in

$$\delta F = \delta(H - \underline{\lambda}^T \underline{f}) = \delta(H - \underline{\lambda}^T \dot{\underline{x}}) \tag{II-50}$$

$$= \frac{\partial H}{\partial u} \delta u + \frac{\partial H}{\partial \underline{x}}^T \delta \underline{x} - \underline{\lambda}^T \delta \dot{\underline{x}} \tag{II-51}$$

or

$$\delta J = \frac{\partial \Phi}{\partial \underline{x}}\Big|_{t_f} \delta x_f + \int_{t_o}^{t_f} \left[\frac{\partial H}{\partial u} \delta u + \frac{\partial H}{\partial \underline{x}}^T \delta \underline{x} - \underline{\lambda}^T \delta \dot{\underline{x}} \right] dt \tag{II-52}$$

Integrating by parts gives

$$\delta J = \frac{\partial \Phi}{\partial \underline{x}}\Big|_{t_f} \delta \underline{x}_f - \underline{\lambda}^T \delta \underline{x}\Big|_{t_o}^{t_f} + \int_{t_o}^{t_f} \left[\frac{\partial H}{\partial u} \delta u + \left(\frac{\partial H}{\partial \underline{x}}^T + \dot{\lambda}^T \right) \delta \underline{x} \right] dt \, . \tag{II-53}$$

However, $\delta \underline{x}(t_o) = 0$ because the initial conditions are fixed. Using the optimality conditions II-38 and II-39, II-53 becomes

$$\delta J = \int_{t_o}^{t_f} \left[\frac{\partial H}{\partial u} \delta u \right] dt \quad .$$

If the variation of the control u is along a direction of search s, i.e.

$$\delta u = s \, \delta \alpha \tag{II-54}$$

where α is a scalar, then the derivative of J along s is

$$\frac{dJ}{d\alpha} = \int_{t_o}^{t_f} \left[\frac{\partial H}{\partial u} s \right] dt \quad . \tag{II-55}$$

Equation II-55 is the inner product of the direction of search and $\frac{\partial H}{\partial u}$. Thus, $\frac{\partial H}{\partial u}$ plays the role of the gradient vector in the finite-dimensional analysis.

Lasdon's algorithm is given as follows:

1. For $i=0$ guess an initial control function $u_o(t)$.
2. Integrate the state system II-41, II-42 from t_o to t_f.
3. Integrate the costate system II-45, II-46 from t_f to t_o.
4. Calculate

$$\frac{\partial H}{\partial u} = g(u_i(t)) = g_i \quad . \tag{II-56}$$

5. Calculate β_i using

$$\beta_i = \frac{\int_{t_o}^{t_f} g(u_i(t))^2 \, dt}{\int_{t_o}^{t_f} g(u_{i-1}(t))^2 \, dt} = \frac{\langle g_i, g_i \rangle}{\langle g_{i-1}, g_{i-1} \rangle} \quad . \tag{II-57}$$

If $i=0$, $\beta_o = 0$.

6. Calculate the direction of search

$$s_i(t) = -g_i(t) + \beta_i \, s_{i-1}(t) \quad . \tag{II-58}$$

7. Let

$$u_{i+1}(t) = u_i(t) + \alpha_i \, s_i(t) \tag{II-59}$$

and determine α_i by performing a one-dimensional minimization, i.e.

$$J(u_i + \alpha_i s_i) \leq J(u_i + \gamma s_i) \quad \forall \gamma > 0 \; . \tag{II-60}$$

8. Increase i and repeat from 2 until the minimum is reached.

Step 5 indicates that Lasdon chose to use the method of calculating β_i that is valid in finite dimensional problems for a quadratic objective function. An alternate formula analogous to expression II-4 has also been derived. A comparison of the applicability and accuracy of the two formulae for β_i is given in the next section. This comparison also serves to illustrate the convergence properties of the CG method on control problems with no terminal state constraints.

Numerical Solutions of Unconstrained Optimal Control Problems

In this thesis, all automatic computations reported were performed on the IBM 360 model 65 digital computer using the FORTRAN IV language and double-precision arithmetic with accuracy of approximately sixteen decimal digits. Computation times quoted are times used by the central processing unit (CPU) during the execution of the program logic. Although the CPU time is the best measure of the computing effort required, it is not precisely reproducible on identical programs due to the multi-programming feature of the system.

All integrations were performed using fourth-order numerical integration methods. During initial studies, variable stepsizes were used, but experience soon revealed that a fixed stepsize caused very little degration in accuracy and increased the computation speeds by as much as a factor of two. In addition, the use of fixed stepsizes greatly reduces the programming effort since trajectories computed from forward integrations are stored at the same time points as those obtained from backward integrations. Each one-dimensional minimization required

in a solution reported here was based upon a cubic polynomical approximation to the contour of the function or functional along the direction of search. Both function values and derivative values were used to determine the polynomial. After a satisfactory approximation was made, the minimum of the polynomial was chosen as the optimum stepsize. This procedure has been used extensively in finite-dimensional problems (59) and has proven satisfactory here for control problems as well.

A control problem with linear dynamics and quadratic cost was given by Hsieh (37) and solved using the CG method by Lasdon et al. (50) as an example. Their method was duplicated for the purpose of checking the computer program, and the results are presented here to illustrate the convergence of the method. Initially, the parameter β_i was determined using II-57. When that formula for calculating β_i is used, the method will be called the simplified conjugate gradient method (SCG). A statement of problem P-1 is:

P-1. Minimize

$$J(u) = \int_0^1 [(x_1-1)^2 + x_2^2 + 0.005u^2]\,dt \tag{II-61}$$

subject to

$$\dot{x}_1 = x_2 \tag{II-62}$$

$$\dot{x}_2 = -x_2 + u \tag{II-63}$$

$$x_1(0) = 1 \tag{II-64}$$

$$x_2(0) = -1 \tag{II-65}$$

Subscripts appearing on variables that do not have the vector notation refer to vector elements (e.g. x_2, x_3) whereas subscripts appearing

with the vector notation refer to iteration numbers (e.g. $\underline{x}_2$ is the second iterate of the vector $\underline{x}$).

This problem can be interpreted as that of controlling a unit mass sliding on a surface with a friction coefficient equal to the reciprocal of the gravitational constant. The mass has an initial velocity of -1 foot per second and its initial position is $x_1(0) =$ one foot. The objective is to find the control that minimizes a linear combination of the magnitudes of the velocity of the mass, the deviation of its position from $x_1 =$ one foot, and the cumulative control effort. The initial control guess was the unit function $u(t) = 1$. Table 1 gives the results of the solution as well as a comparison with a steepest descent solution obtained using the SCG program with $\beta_i = 0 \ \forall i$. Figure 1 shows the convergence of the control iterates. After four iterations of the SCG method, the control shows a high degree of agreement with the steepest descent solution given by Hsieh (37) after twenty-four steps from an initial control of $u_0(t) = 1$.

The calculation of β_i as the inner product of the current gradient with itself divided by the inner product of the previous gradient with itself was valid in finite dimensions for objective functions that were quadratic. It is instructive to examine the conditions under which the same simplified expression for β_i is valid in the function space version of the SCG method. The Euclidean vector space E_n in which the search for the n-dimensional vector $\underline{x}^*$ took place is replaced in the control problem by the Hilbert space $\mathcal{H}$ of control functions. Contours of constant values of a quadratic objective function are ellipsoids in E_n. The analogous situation in function space is for the 'contours' of constant

Table 1. Convergence of the unconstrained CG method on problem P-1

Iteration Number i	CG		SD	
	$J(u_i)$	$\langle g_i, g_i \rangle$	$J(u_i)$	$\langle g_i, g_i \rangle$
0	.2685	$.6294 \times 10^{-1}$	.2685	0.6294×10^{-1}
1	.1707	$.2350 \times 10^{-1}$	.1707	0.2350×10^{-1}
2	$.8741 \times 10^{-1}$	$.1138 \times 10^{-2}$	.1256	0.1380×10^{-1}
3	$.7211 \times 10^{-1}$	$.1790 \times 10^{-3}$	.1041	0.6091×10^{-2}
4	$.7139 \times 10^{-1}$	$.3837 \times 10^{-3}$	0.9306×10^{-1}	0.3739×10^{-2}
5	$.7034 \times 10^{-1}$	$.2636 \times 10^{-3}$	0.8678×10^{-1}	0.2322×10^{-2}
6	$.7003 \times 10^{-1}$	$.8390 \times 10^{-4}$	0.8283×10^{-1}	0.1469×10^{-2}
7	$.6979 \times 10^{-1}$	$.1464 \times 10^{-3}$	0.8008×10^{-1}	0.1251×10^{-2}
8	$.6959 \times 10^{-1}$	$.2653 \times 10^{-4}$	0.7807×10^{-1}	0.7723×10^{-3}
9			0.7647×10^{-1}	0.7864×10^{-3}
10			0.7526×10^{-1}	0.4691×10^{-3}
11			0.7424×10^{-1}	0.5174×10^{-3}
12			0.7345×10^{-1}	0.3045×10^{-3}
13			0.7277×10^{-1}	0.3466×10^{-3}
14			0.7224×10^{-1}	0.2056×10^{-3}
15			0.7178×10^{-1}	0.2359×10^{-3}
20			0.7047×10^{-1}	$.7327 \times 10^{-4}$
25			0.6990×10^{-1}	$.3978 \times 10^{-4}$

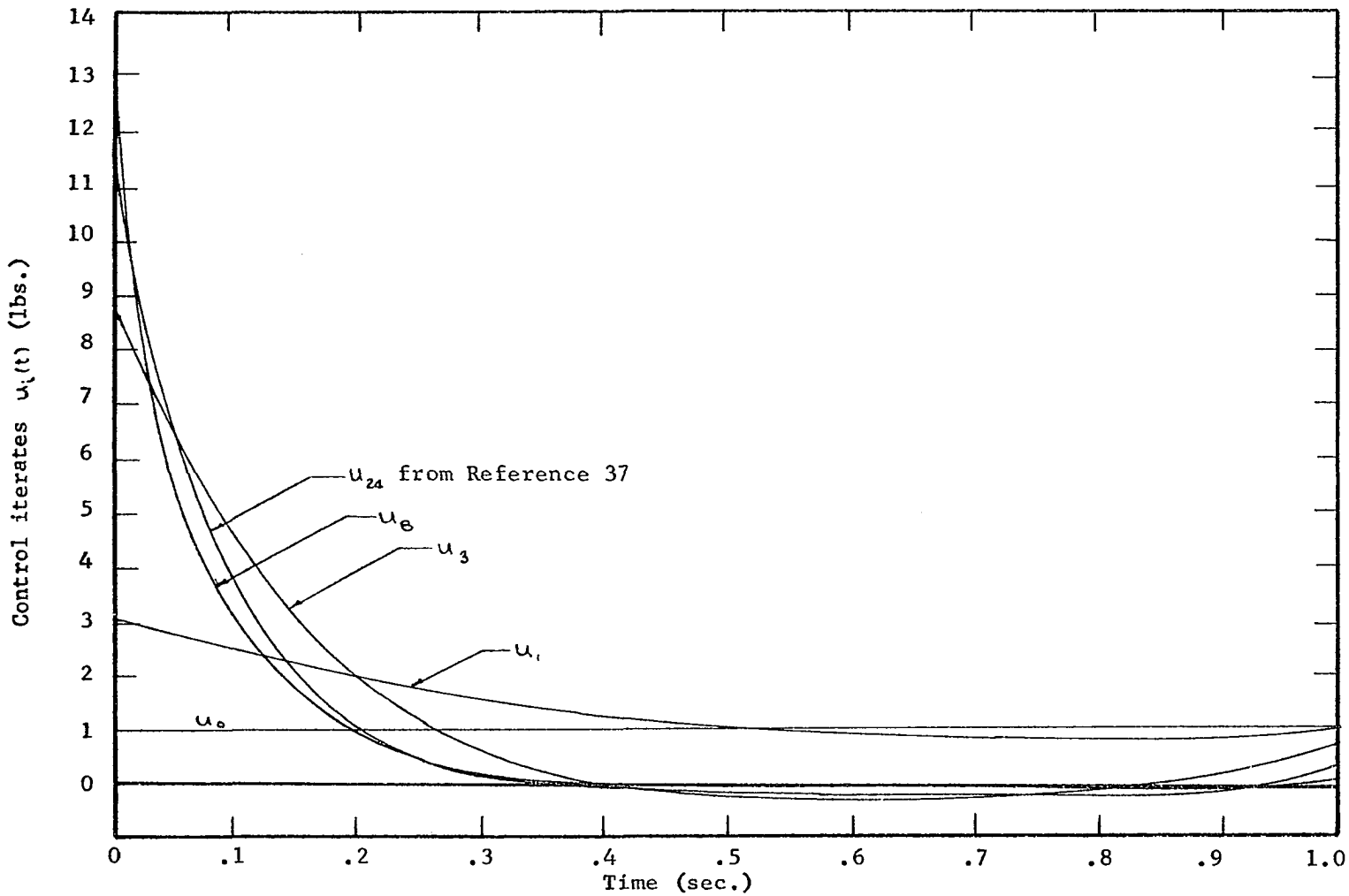


Figure 1. Control iterates of the CG solution to problem P-1

cost to be quadratic in the control space. If the system dynamics (Equations II-32) of the control problem are linear i.e. if

$$\dot{\underline{x}} = D(t)\,\underline{x}(t) + B(t)\,\underline{u}(t) \tag{II-66}$$

$$\underline{x}(t_0) = \underline{x}_0 \tag{II-67}$$

then

$$\underline{x}(t) = \Phi(t,t_0)\,\underline{x}_0 + L[\underline{u}(t)] \tag{II-68}$$

where $\Phi(t,t_0)$ is the transition matrix for the system II-66 and $L[\cdot]$ is a linear operator defined by

$$L[\underline{u}(t)] = \int_{\tau_0}^{t} \Phi(t,\lambda)\,B(\lambda)\,\underline{u}(\lambda)\,d\lambda \quad . \tag{II-69}$$

A control problem having a cost functional J that is quadratic in the state variables $\underline{x}$ and the control variables $\underline{u}$ is quadratic in the control space $\mathcal{H}$ only if the states can be related to the control through a linear transformation such as II-68. Nonlinear dynamic equations do not in general permit a linear relationship between the states and the control. Therefore a quadratic cost functional of the form

$$J = \frac{1}{2}\langle \underline{x}(t_f), K\underline{x}(t_f)\rangle + \frac{1}{2}\int_{t_0}^{t_f}\left[\langle \underline{x}(t), Q(t)\,\underline{x}(t)\rangle + \langle \underline{u}(t), R(t)\,\underline{u}(t)\rangle\right]dt \tag{II-70}$$

where

K is a positive semidefinite $n \times n$ matrix,

$Q(t)$ is a positive semidefinite $n \times n$ matrix, and

$R(t)$ is a positive definite $m \times m$ matrix

does not produce quadratic 'contours' in the control space unless the dynamics are linear.

The purpose of the previous argument was to determine the class of optimal control problems for which the simplified formula II-57 applies. Problems such as the problem of Hsieh with linear dynamics and quadratic cost II-70 constitute that class.

The alternate means of calculating the parameter β_i for the finite-dimensional CG method is given by II-4. The CG method obtained by using II-4 will be referred to as the pure conjugate gradient method (PCG). Methods for determining β_i using the function space analog of II-4 have been derived by Sinnott and Luenberger (69) by Tripathi and Narendra (72, 73) and Pagurek and Woodside (58). A similar derivation is given in Appendix A. The matrix N in the finite-dimensional version is replaced in Hilbert space by the linear operator $\hat{N}$. The function $\hat{N} s_{i-1}$ can be determined after integrating two sets of auxiliary equations. β_i is then determined from

$$\beta_i = \frac{\langle g_i, \hat{N} \underline{s}_{i-1} \rangle}{\langle \underline{s}_i, \hat{N} \underline{s}_{i-1} \rangle} \qquad \text{(II-71)}$$

which requires two quadrature integrations to evaluate the indicated inner products.

Recently some numerical results comparing the two methods of determining β_i have been published for optimal control problems by Pagurek and Woodside (58). Although two examples of quadratic costs with linear dynamics are given, the control variables are bounded in each case. Thus the argument that the two methods are equivalent for this class of problems is not directly tested.

The auxiliary differential equations necessary for the calculation of

using II-71 were programmed for the unit mass problem P-1 presented previously. Since that problem has quadratic cost and linear dynamics, the two methods were expected to give similar results. Table 2 shows that the cost functional and the gradient magnitudes were reduced comparably by the two methods. The numerical values of β_i differ considerably after the second iteration, but this difference is thought to be a result of numerical procedures that cause the first non-zero values of β to differ slightly. This small difference initially leads to different sequences of search points in the control space. A direct comparison of the numerical values of β is not valid unless the steps being compared are taken from exactly the same points in the control space. However, it can be seen that the overall convergence rates of the two methods are very nearly the same. This tends to confirm the validity of the SCG method for control problems with linear dynamics and quadratic cost.

A comparison of the run times from Table 2 shows nearly a 25% increase when using the SCG method. Pagurek reports an increase of 20%. The additional programming complexity and the substantial increase in the running time that result using the PCG method are sufficient to justify the use of the SCG method on all quadratic problems with linear dynamics, and on any other problems where the approximation is reasonable.

An extensive study of the accuracy of the simplified β formula on problems that are not quadratic with linear dynamics has not been done. However, some data are provided here as a result of comparative solutions of an unconstrained problem with a quadratic cost but with nonlinear dynamics. A statement of problem P-2 in Bolza form follows:

Table 2. Comparison of the SCG and PCG methods for problem P-1

Iteration Number	$J(u_i)$		$\langle g_i, g_i \rangle \times 10^2$		β_i	
	SCG	PCG	SCG	PCG	SCG	PCG
0	0.2685	0.2685	6.294	6.294		
1	0.1707	0.1707	2.350	2.350	0	0
2	0.0874	0.08750	0.1138	0.1219	0.3734	0.3612
3	0.07211	0.07778	0.0179	0.2788	0.0484	0.0578
4	0.07139	0.07218	0.0684	0.0215	0.1573	2.548
5	0.07034	0.07078	0.0264	0.0154	3.819	0.1241
6	0.07003	0.07044	0.0039	0.0112	0.3856	0.8301
7	0.06979	0.06962	0.0146	0.00147	0.3182	0.5962
8	0.06959	0.06956	0.00265	0.00182	1.744	0.1496
			SCG	PCG		
	Execution times:		13.3 sec.	16.5 sec.		

P-2. Minimize $$J(u) = 2x_1(5)^2 + 2x_2(5)^2 + \int_0^5 [x_1^2 + x_2^2 + u^2]\,dt \qquad \text{(II-72)}$$

subject to $$\dot{x}_1 = (1 - x_1^2 - x_2^2)x_1 - x_2 + u \qquad \text{(II-73)}$$

$$\dot{x}_2 = x_1 \qquad \text{(II-74)}$$

$$x_1(0) = 0 \qquad \text{(II-75)}$$

$$x_2(0) = 2 \qquad \text{(II-76)}$$

This problem was given by Schley and Lee (68) who point out that the uncontrolled dynamics exhibit a limit cycle on the unit circle in the state space. The objective is to find a control that (1) eliminates the limit cycle character by keeping that states near the origin and (2) has

small magnitude in the integral squared sense.

Figure 2 shows the optimal controls obtained numerically and the results given in (68). Table 3 gives a comparison of the SCG and the PCG methods. The rate of convergence was significantly increased by using the PCG formulation. After seven iterations of the PCG method, the functional value was less than after twenty-four iterations of the SCG method. In addition, the PCG method reduced the gradient magnitudes more rapidly and more uniformly.

Lasdon et al. (50) report an oscillation in the magnitude of the gradient on a different numerical problem. The phenomenon is exhibited in the solution of this example problem and is decidedly more pronounced for the simplified β formula. An intuitive explanation for this oscillation might be that the SCG method excludes the possibility of negative values of β. Geometrically, this says that each new direction of search cannot have a component along the negative previous direction of search. Quadratic contours of constant cost are everywhere convex and thus should never require a direction of search that has a component back along the previous line of search. A simple two-dimensional illustration of this point is given in Figure 3.

The restriction that $\beta_i \geq 0$ prevents 'acute angle turns' which on a non-quadratic problem could force small stepsizes and require more iterations. The solution of this example problem using the PCG method and a Lagrange form of the cost functional produced nine negative values of β in twenty-four steps.

Although the use of the accurate formula for calculating β requires additional programming effort and longer execution times, a significant

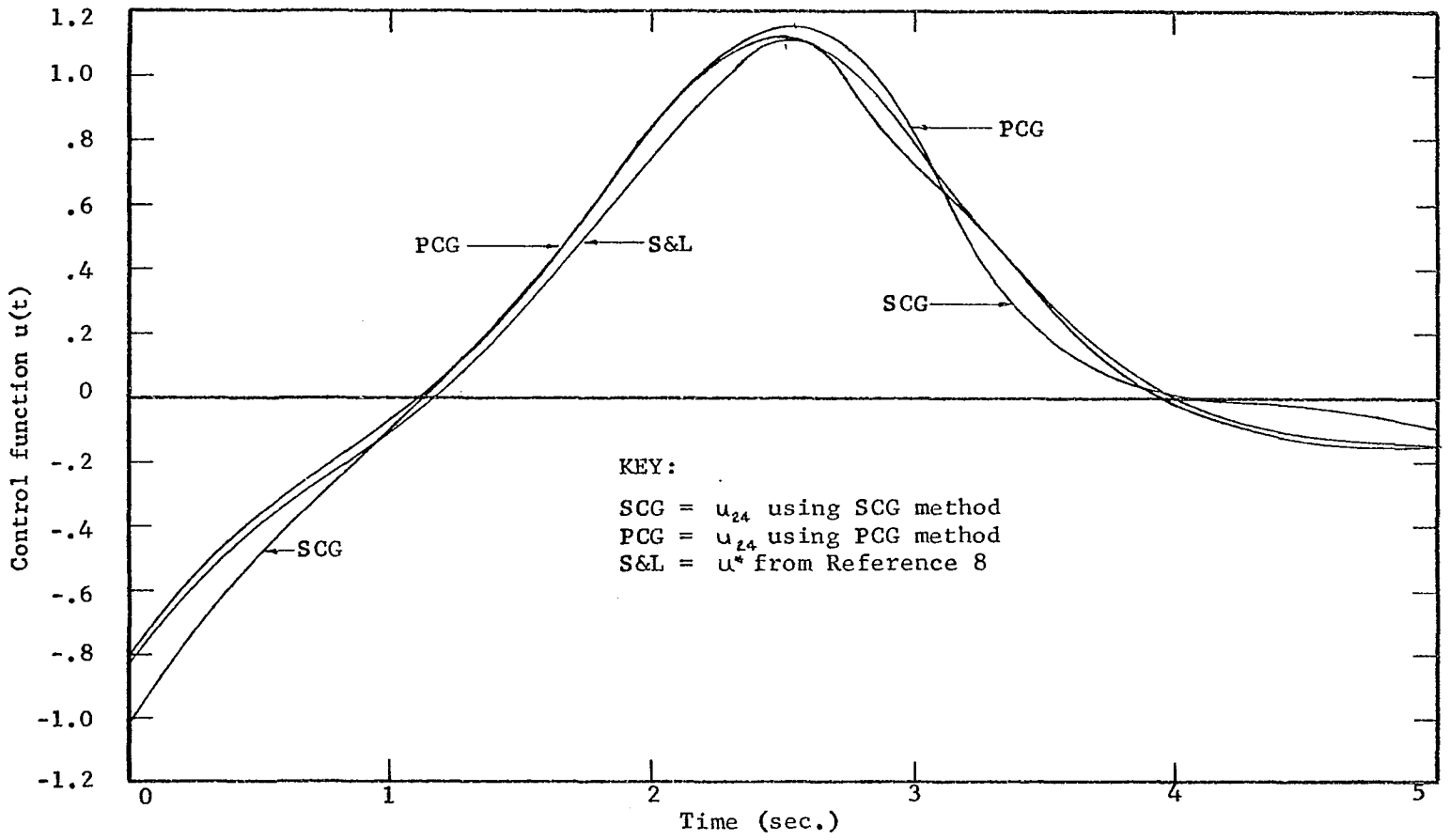


Figure 2. Numerical solutions of problem P-2

Table 3. Comparison of the SCG and PCG solutions of problem P-2

Iteration Number	SCG		PCG	
	$J(u_i)$	$\langle g_i, g_i \rangle$	$J(u_i)$	$\langle g_i, g_i \rangle$
0	10.9118	14.2126	10.9118	14.2126
1	9.0917	9.7058	9.0917	9.7058
2	8.3582	7.6621	8.6532	16.1310
3	8.0449	2.7313	8.0386	3.4232
4	7.8789	5.5149	7.9311	11.5300
5	7.8602	3.8629	7.7525	5.2700
6	7.8428	2.0676	7.6593	1.1127
7	7.8343	3.2297	7.5981	5.5360
8	7.8164	8.1103	7.5457	0.6831
9	7.7815	15.7724	7.5134	2.0770
10	7.7252	21.6517	7.4964	1.0802
11	7.6675	19.7288	7.4808	0.2352
12	7.6359	12.2861	7.4775	0.1489
13	7.6192	7.0847	7.4748	0.1087
14	7.6084	5.4209	7.4733	0.04921
15	7.5985	6.1297	7.4724	0.03817
16	7.5863	7.0796	7.4719	0.02663
17	7.5740	5.2600	7.4714	0.01666
18	7.5667	2.4002	7.4711	0.01465
19	7.5636	0.9067	7.4708	0.009810
20	7.5623	0.4469	7.4707	0.004966
21	7.5615	0.4741	7.4706	0.001526
22	7.5602	0.9835	7.4706	0.003007
23	7.5571	2.7723	7.4705	0.0007414
24	7.5478	7.0189	7.4705	0.001005

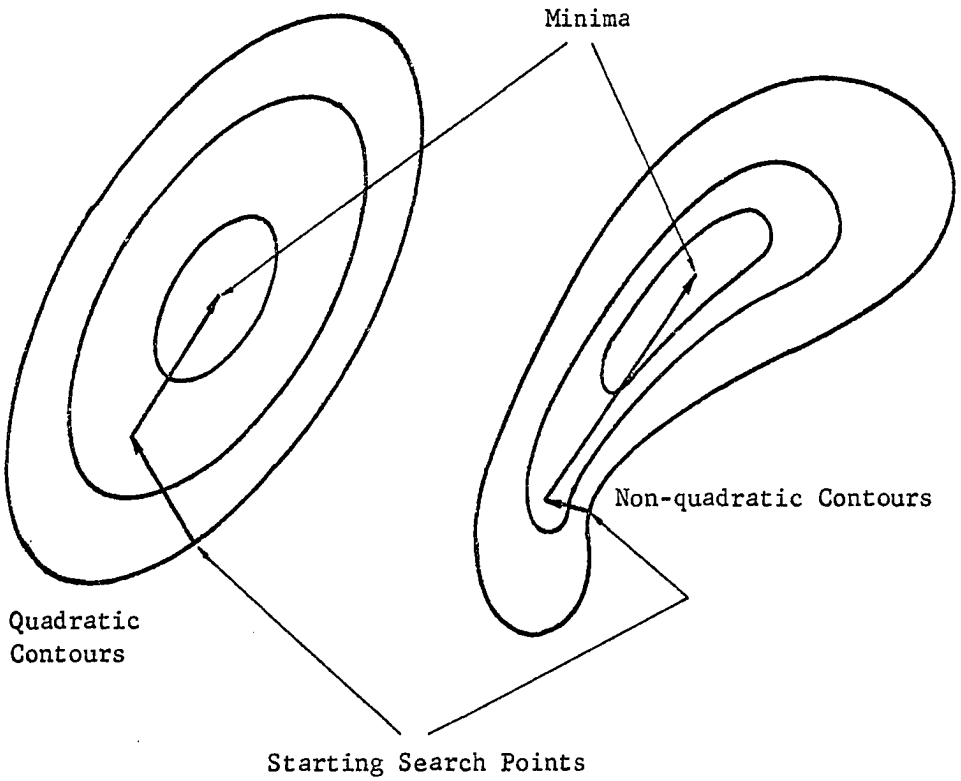


Figure 3. Contours of constant cost in a two-dimensional space

performance advantage appears to be realized when the problem is not purely quadratic. A comparison of execution times is given in Table 4 which indicates that the use of the SCG method rather than the PCG method on this class of problems results in an execution time increase factor of approximately 2.

Table 4. Comparison of execution times on problem P-2 using the SCG and PCG methods

	SCG	PCG
Execution time for 24 iterations	29.3 sec.	40.2 sec.
Average time per step	1.22 sec.	1.67 sec.
Estimated time for eight steps	----	14 sec.

In addition to the primary objective of comparing the PCG and the SCG methods, this example problem was used to investigate the influence of transforming the cost functional from Bolza to Lagrange form. The auxiliary equations for calculating β that are given by Tripathi and Narendra (72, 73) apply to the Bolza form and use different boundary conditions than those used by Sinnott and Luenberger (69) who considered only the Lagrange problem. The theoretical equivalence of the two systems of auxiliary equations can be shown. Computational advantages of one form over the other are of interest however.

The conversion from Bolza to Lagrange form is accomplished by

defining a new state variable x_{n+1} which satisfies the equation $\dot{x}_{n+1} = 0$. The derivative of the term of the Bolza cost functional that depends only upon the final states is placed under the integral along with the constant function x_{n+1}. Proper choice of the initial conditions on x_{n+1} produces a Lagrangian formulation that is equivalent to the Bolza problem. (See Reference 3, pp. 300-301.) The Lagrange form of example problem P-2 becomes:

P-3. Minimize

$$J(u) = \int_0^5 [x_1^2 + x_2^2 + u^2 - 4x_1^4 + 4x_1^2 - 4x_1^2 x_2^2 + 4x_1 u + x_3]\, dt \tag{II-75}$$

subject to

$$\dot{x}_1 = (1 - x_1^2 - x_2^2)x_1 - x_2 + u \tag{II-76}$$

$$\dot{x}_2 = x_1 \tag{II-77}$$

$$\dot{x}_3 = 0 \tag{II-78}$$

$$x_1(0) = 0 \tag{II-79}$$

$$x_2(0) = 2 \tag{II-80}$$

$$x_3(0) = \frac{1}{5}[2x_1(0) + 2x_2(0)] = \frac{8}{5} \tag{II-81}$$

Table 5 presents the PCG solutions of the problem in both Lagrange and Bolza forms. The comparable convergence rates seem to substantiate the theoretical equivalence of the two sets of auxiliary equations given by Sinnott and Tripathi. However, the execution time for twenty-four iterations on the Lagrange form was 87% greater than the execution time for the Bolza form. This increase is almost certainly explained by the introduction of the new state variable. The conclusion that the

Table 5. Comparison of PCG solutions of the Lagrange problem P-2 and the Bolza problem P-3

Iteration Number i	Lagrange (P-2)		Bolza (P-3)	
	$J(u_i)$	$\langle g_i, g_i \rangle$	$J(u_i)$	$\langle g_i, g_i \rangle$
0	10.9117	14.0311	10.9118	14.2126
1	9.5063	14.2618	9.0917	9.7058
2	8.1717	9.5370	8.6532	16.1310
3	7.7346	6.3717	8.0386	3.4232
4	7.7128	3.8221	7.9311	11.5300
5	7.5924	1.7674	7.7525	5.2700
6	7.5862	1.0587	7.6593	1.1127
7	7.5245	1.4299	7.5981	5.5360
8	7.5197	1.4837	7.5457	0.6831
9	7.4823	0.2257	7.5134	2.0770
10	7.4821	0.1352	7.4964	1.0802
11	7.4742	0.1492	7.4808	0.2352
12	7.4740	0.1192	7.4775	0.1489
13	7.4721	0.02477	7.4748	0.1087
14	7.4720	0.02430	7.4733	0.04921
15	7.4715	0.01562	7.4724	0.03817
16	7.4713	0.02774	7.4719	0.02663
17	7.4709	0.006439	7.4714	0.01666
18	7.4709	0.006425	7.4711	0.01465
19	7.4709	0.006732	7.4708	0.009810
20	7.4708	0.005984	7.4707	0.004966
21	7.4707	0.002601	7.4706	0.001526
22	7.4707	0.001640	7.4706	0.003007
23	7.4706	0.001394	7.4705	0.0007414
24	7.4706	0.001011	7.4705	0.001005

mathematical description of the optimal control problem should be made as simple as possible is probably a valid generalization from the results of this example problem. Further conclusions regarding the convergence rates of the alternate cost functional formulations cannot be made on the basis of the results obtained thus far.

CHAPTER III. SOLUTION OF OPTIMAL CONTROL PROBLEMS WITH TERMINAL STATE CONSTRAINTS USING THE CONJUGATE GRADIENT METHOD WITH PENALTY FUNCTIONS

Characteristics of the Penalty Function Method

Most optimal control problems are constrained by one or more algebraic relationships involving the state variables at the terminal time. Although these constraints are merely boundary conditions for the variational problem, they create complications of a computational nature for any of the direct solution techniques. Modifications to either the problem format or to the computational algorithm are required. This chapter deals with the penalty function method as a means of adapting the CG method to optimal control problems with terminal state constraints. The terminal time is assumed to be specified explicitly. The terminal conditions may be linear or nonlinear algebraic relations of the form

$$\underline{\Omega}\,(\underline{x}(t_f)) = \underline{0} \tag{III-1}$$

where $\underline{\Omega}$ is a p-vector with $p \leq n$.

Unlike the other two methods presented in subsequent chapters of this thesis, the penalty function approach is an alteration of the form of the optimal control problem itself rather than a modification of the numerical technique used to solve it. The constrained problem is approximated by one or more unconstrained problems by adding to the cost functional a positive measure of the constraint violation. If the constrained problem has the form given by Equations II-31, II-32, and II-33, the related unconstrained problem has the following form:

P-4. Minimize $$\hat{J}(\underline{u}) = \Phi(\underline{x}(t_o), t_o\ \underline{x}(t_f), t_f) + \int_{t_o}^{t_f} F(\underline{x}, \underline{u}, t)\,dt + \underline{\Psi}^T W \underline{\Psi} \qquad \text{(III-2)}$$

$$= J + \underline{\Psi}^T W \underline{\Psi} \qquad \text{(III-3)}$$

subject to

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}, t) \qquad \text{(III-4)}$$

$$\underline{x}(t_o) = \underline{x}_o \qquad \text{(III-5)}$$

where W is a $p \times p$ positive definite matrix of penalty constants, and $\underline{\Psi}$ is the constraint violation, i.e. $\underline{\Psi} = \underline{\Omega}(\underline{x}(t_f))$ when $\underline{x}(t_f)$ does not satisfy the constraint.

It can be seen that any violation of the constraint Equation III-1 adds a positive increment to $\hat{J}$. Minimization of $\hat{J}$ should then drive $\underline{\Psi}$ to zero as well as $J(\underline{u})$ to $J(\underline{u}^*)$.

The penalty function approach attempts to make controls that produce larger constraint violations lie on contours of higher cost in the control space than those producing smaller constraint violations. An unconstrained relative minimum of $\hat{J}$ is constructed at the constrained minimum of J. From a computational point of view, the geometric nature of the relative minimum of $\hat{J}$ is important. The choice of the penalty constants in W influences the 'shape' of the cost functional throughout the entire control space. In a typical optimal control problem with nonlinear dynamics, the effect of the penalty term in the control space is difficult or impossible to assess without numerical experimentation. Thus the choice of the penalty constants is often arbitrary and must be altered on the basis of the success or failure of trial solutions.

Large penalty constants affect the solution process by causing large gradient components for controls that produce large constraint violations. The boundary conditions on the adjoint variables involve the penalty constants explicitly. The effect of their presence can cause the cost functional $\hat{J}$ to form a very 'steep-sided valley' along the locus of those controls that produce constraint satisfaction. The gradients to these surfaces point in directions across the valley. Such gradients often cause direct methods to make slow progress along the valley and toward the minimum. Even the conjugate direction methods which deflect the directions of search so that they become more nearly parallel to the axis of the valley may converge slowly because the differences in the gradient magnitudes in the different directions may lead to the accumulation of roundoff errors.

Mehra and Bryson (56) discuss difficulties encountered in using the CG method with penalty functions on nonlinear problems having more than two terminal constraints. They state that the eigenvalues of the linearized dynamical system often differ greatly in magnitude when penalty functions are used, a fact that leads to slow convergence of a gradient method. Lasdon, Mitter and Waren (50) report poor convergence of the gradient magnitudes when using the penalty function approach with the CG method on a simple rocket launch problem. Numerical solutions of this problem using penalty functions are given later in this chapter.

Some of the difficulties encountered using penalty functions can be avoided by replacing a single solution attempt by a sequence of solutions involving increased weighting of the constraint violation. Each new subproblem is started from the solution to the previous subproblem or from

an estimate derived from solutions to the previous subproblems. This method has been studied extensively by Fiacco and McCormick (21,22,23) who call the procedure the sequential unconstrained minimization technique (SUMT). Under certain convexity requirements, they prove that when applied to a constrained function of several variables, the method produces a sequence of solutions that converges to the constrained minimum. The same procedure can be applied to the control problem and has been used successfully in this study. Unfortunately, the choice of the penalty constants used in each unconstrained subproblem must still be made arbitrarily at first and modified on the basis of experience with each problem. Fiacco and McCormick (21) have suggested several criteria for choosing both the initial values of the penalty constants and the amount of their increase between subproblems. However these estimates are derived for finite-dimensional problems with inequality constraints. In practice, the arbitrary choice of the penalty constants for control problems has not been especially difficult.

Numerical Solutions Using Conjugate Gradient Methods With Penalty Functions

Two numerical examples are presented here to demonstrate the use of penalty functions with the conjugate gradient method. The first is the rocket launch problem given by Lasdon, Mitter and Waren (50). The objective is to maximize the horizontal velocity of a rocket under the assumptions of a constant gravitational acceleration of 32 ft./sec.^{2}, two-dimensional vacuum flight, and a constant thrust acceleration of twice the gravitational acceleration. The control variable is the thrust

direction with respect to the horizontal. The thrust time is specified as 100 seconds and the terminal boundary conditions require a vertical velocity of zero at an altitude of 100,000 feet. After nondimensionalizing, the mathematical problem statement becomes:

P-5. Minimize $J = -x_3(1)$ (III-6)

subject to $\dot{x}_1 = x_2$ (III-7)

$\dot{x}_2 = 6.4 \sin u - 3.2$ (III-8)

$\dot{x}_3 = 6.4 \cos u$ (III-9)

with $x_1(0) = 0$ (III-10)

$x_2(0) = 0$ (III-11)

$x_3(0) = 0$ (III-12)

$x_1(1) = 1$ (III-13)

$x_2(1) = 0$ (III-14)

Introduction of penalty functions to account for the terminal state constraints gives the new cost functional

$$\hat{J} = -x_3(1) + 100\, P_1 (x_1(1)-1)^2 + 0.1\, P_2\, x_2(1)^2 \qquad \text{(III-15)}$$

where W has been chosen to be

$$\begin{pmatrix} 100\, P_1 & 0 \\ 0 & 0.1\, P_2 \end{pmatrix}$$

Problem P-5 admits an analytical solution of the form $u^*(t) = \tan^{-1}(b - ct)$ although the constants b and c must be determined by solving simultaneous transcendental equations numerically. The 'exact' solution based on the

values $b = 4.8412$ and $c = 0.06319$ is included in the results given here for comparison purposes. The solution given in Tables 6 and 7 was obtained using the SUMT approach with the PCG method starting from an initial control of $u_0(t) = \frac{\pi}{2} - \frac{t}{100}$. The solution method differs from that apparently used by Lasdon et al. (50) in that a sequence of subproblems was solved with each subproblem using the PCG method instead of the SCG method. Four unconstrained problems were solved using the values for the penalty constants given in Table 8.

An oscillation in the magnitude of the gradient was mentioned by Lasdon et al. (50) and was also observed in the SUMT-PCG solution given here. Investigation of several causes led ultimately to the opinion that the nature of the cost functional for this problem was very irregular along many directions of search. Figure 4 is a plot obtained by computing both the value of the functional $\hat{J}$ and its slope along the direction of search at various points along the direction of search. BETA represents the original stepsize estimate used to initiate the one-dimensional minimization and is based upon the optimum stepsize obtained from the previous iteration. The extremely non-unimodal character of the contour suggests the reason for the oscillation of the gradient magnitudes. Several contours were observed that had very small negative slopes at the search points and relative minima at extremely small stepsizes in relation to those that occurred on other iterations. The appearance of more than one relative minimum makes the location of the proper minimum a difficult task for automated logic. In contrast, Figure 5 shows another profile from the same unconstrained subproblem which exhibits unimodal character over an even larger range of stepsizes than given in Figure 4.

Table 6. SUMT-PCG solution of the rocket launch problem P-5

Time (sec.)	Numerical Solution for $u^*(t)$ (rad.)	'Exact' Solution $u^*(t)$ (rad.)
0.0	1.3646	1.3670
5.0	1.3545	1.3532
10.0	1.3407	1.3375
15.0	1.3234	1.3193
20.0	1.3023	1.2981
25.0	1.2771	1.2732
30.0	1.2471	1.2434
35.0	1.2107	1.2073
40.0	1.1659	1.1627
45.0	1.1091	1.1066
50.0	1.0347	1.0343
55.0	0.9347	0.9387
60.0	0.7984	0.8097
65.0	0.6153	0.6331
70.0	0.3802	0.3960
75.0	0.1020	0.1019
80.0	-0.1923	-0.2104
85.0	-0.4666	-0.4869
90.0	-0.6938	-0.7017
95.0	-0.8658	-0.8598
100.0	-0.9896	-0.9756

Table 7. SUMT-PCG solution of the rocket launch problem P-5

Time (sec.)	Numerical Solution $x_1(t)$ (ft.)	$x_2(t)$ (ft./sec.)	'Exact' Solution $x_1(t)$ (ft.)	$x_2(t)$ (ft./sec.)
0.0	0	0	0	0
5.0	383	153.1	383	157.9
10.0	1,529	305.1	1,528	304.8
15.0	3,432	456.1	3,429	455.5
20.0	6,087	605.5	6,080	604.6
25.0	9,484	753.0	9,472	751.7
30.0	13,612	897.9	13,593	896.2
35.0	18,457	1039.4	18,429	1037.4
40.0	23,999	1176.3	23,959	1173.9
45.0	30,210	1306.9	30,157	1304.1
50.0	37.052	1428.1	36,985	1425.1
55.0	44,468	1535.0	44,385	1532.3
60.0	52,365	1619.4	52,272	1618.2
65.0	60,603	1668.0	60,512	1670.4
70.0	68,954	1661.6	68,887	1669.0
75.0	77,091	1578.6	77,070	1588.7
80.0	84,587	1404.2	84,611	1411.1
85.0	90,982	1140.0	91,025	1140.7
90.0	95,867	803.8	95,902	800.5
95.0	98,938	418.1	98,954	414.6
100.0	99,997	1.5	100,000	0.0
$x_3(100) = 3510.1$ (ft./sec.)			$x_3^*(100) = 3508.1$ (ft./sec.)	

Table 8. Penalty constants for the SUMT-PCG solution of the rocket launch problem P-5

Subproblem Number	P_1	P_2
1	2	5
2	20	50
3	200	500
4	2000	5000
Penalty constants from Reference (50)		
	$P_1 = 200$	$P_2 = 500$

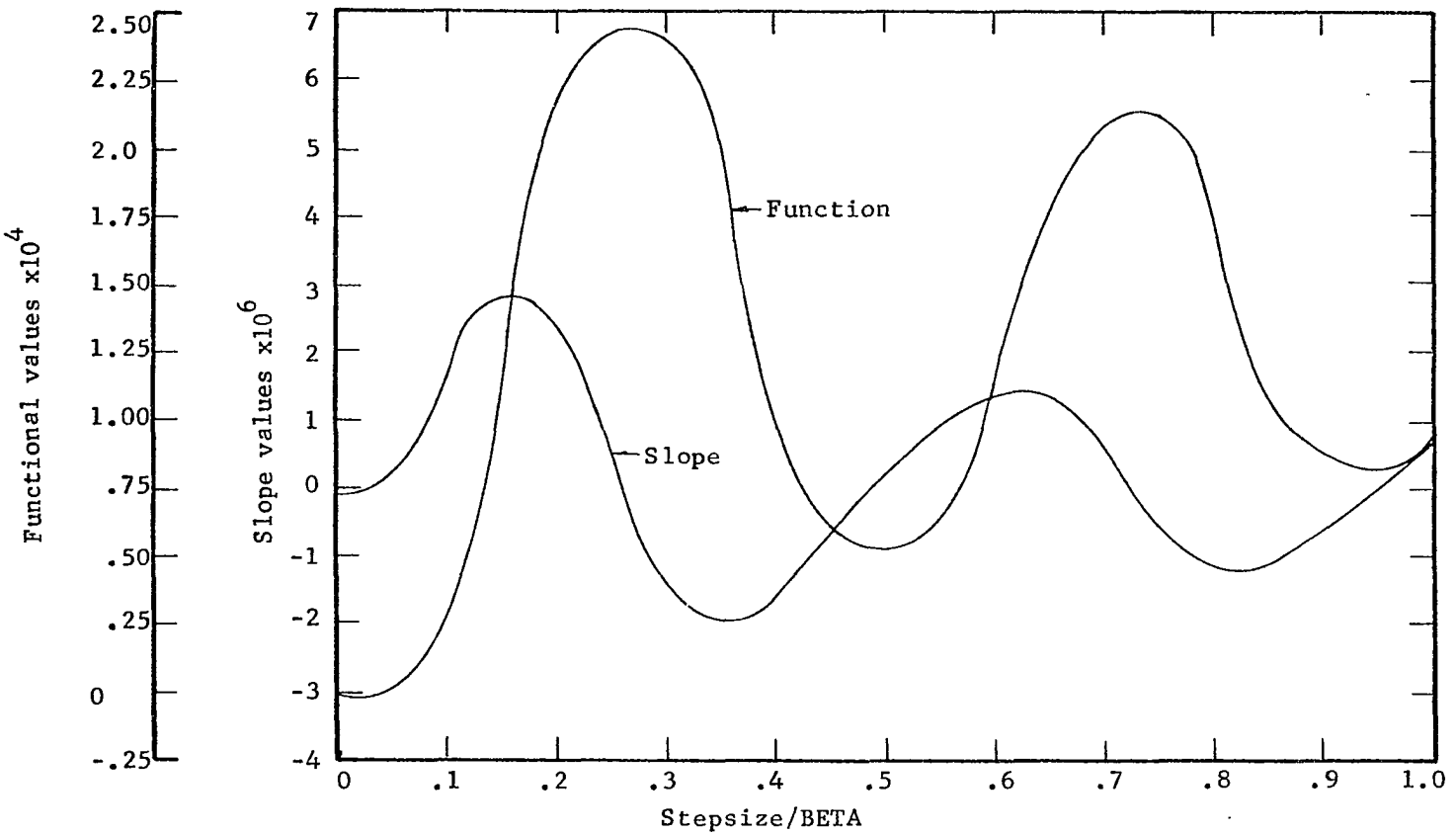


Figure 4. Functional contour along one direction of search in problem P-5

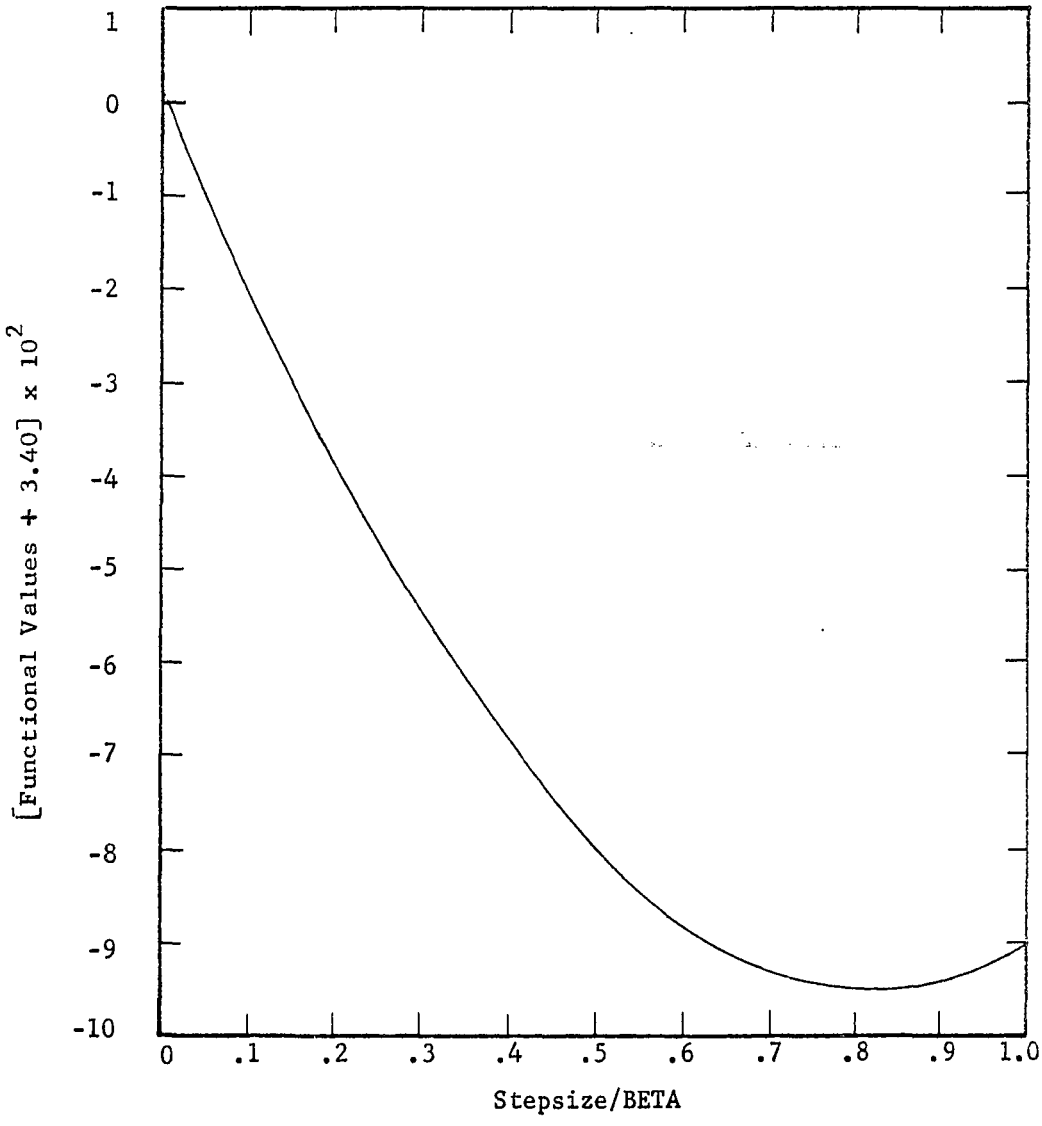


Figure 5. Functional contour along one direction of search in problem P-5

This problem and subsequent experience indicates that a reasonably sophisticated one-dimensional minimization method should be implemented. The procedure should bracket the relative minimum with a small enough interval that the functional contours are at least unimodal between the bracket points. A polynomial fit would be accurate in such a case. A Fibonacci search technique (70), however, depends upon a unimodal character of the function along the direction of search and for the optimal control problem may have difficulty in determining the minimum.

A second numerical example of the penalty function approach using the PCG method is given by the solution of the Van der Pol problem (8,71).

P-6. Minimize $$J(u) = \frac{1}{2}\int_0^5 [x_1^2 + x_2^2 + u^2]\,dt \tag{III-16}$$

subject to $$\dot{x}_1 = x_2 \tag{III-17}$$

$$\dot{x}_2 = -x_1 + (1 - x_1^2)x_2 + u \tag{III-18}$$

$$x_1(0) = 1 \tag{III-19}$$

$$x_2(0) = 0 \tag{III-20}$$

$$\Omega(\underline{x}(5)) = -x_1(5) + x_2(5) - 1 = 0 \tag{III-21}$$

The penalty term was of the form $\frac{1}{2}P[\Omega(\underline{x}(5))]^2$, and the initial control iterate was $u_0(t) = 0$, $0 \leq t \leq 5$. Table 9 gives the values of the functional and the constraint violation resulting from each subproblem.

The solution to the Van der Pol problem presented in Table 9 was obtained after several trial sequences of penalty constants were tried. A single unconstrained solution was made using a penalty constant value

Table 9. SUMT-PCG solution of the Van der Pol problem P-6

Subproblem Number	Penalty Constant, P	$J = \hat{J} - \frac{1}{2} P \Psi^2$	$\Psi = \Omega(\xi(5))$	Number of Steps Taken
1	10	1.65837	-0.05565	8
2	50	1.68340	-0.01171	2
3	250	1.68527	-0.002367	5
4	1,250	1.68633	-0.000462	2
5	6,250	1.68652	-0.000092	2

of P = 250. Again, the initial control was $u_o(t) = 0$. The results are given in Table 10. It is evident from the data that the SUMT approach with penalty function converges more rapidly than the solution of a single unconstrained problem with a relatively large penalty constant. The fixed penalty constant method did not produce an accurate solution after 38 iterations. However, the SUMT method converged satisfactorily in a total of 19 iterations. The CPU times also reveal the greater efficiency of the SUMT approach.

It should be noted that the penalty function approach may be used for nonlinear as well as linear constraints. A fixed penalty constant solution of the Van der Pol problem was accomplished using the nonlinear constraint given in Chapter IV by Equation IV-59. The presence of the nonlinear constraint presented no additional complications.

Table 10. Penalty function solutions using fixed penalty constant and SUMT methods on problem P-6

Iteration Number	SUMT			Fixed Penalty Constant		
	Penalty Constant	$\hat{J}$	Ψ	Penalty Constant	$\hat{J}$	Ψ
1	10	7.8901	0.06054	250	8.0154	0.00184
2		2.1427	-0.03214		7.5215	0.03647
3		2.0813	-0.10543		7.1919	0.05554
4		2.0456	-0.02881		6.8867	0.06806
5		1.7207	-0.12491		6.5547	0.07231
6		1.6824	-0.06012		3.1067	-0.01820
7		1.6745	-0.06622		2.8850	-0.03530
8		1.6739	-0.05565		2.7197	-0.0434
9	50	1.6882	-0.01258		2.5183	-0.0474
10		1.6868	-0.01171		2.1027	-0.00089
11	250	1.6898	-0.02451		2.0720	0.00038
12		1.6872	-0.00187		2.0687	-0.00452
13		1.6871	-0.00240		2.0604	-0.00024
14		1.6860	-0.00214		2.0575	-0.00447
15		1.6860	-0.00237		2.0496	0.00003
16	1,250	1.6865	-0.00047		2.0466	-0.00454
17		1.6865	-0.00046		2.0215	-0.00382
18	6,250	1.6865	-0.00010		2.0116	0.00670
19		1.6865	-0.00009		1.9683	-0.00568
27					1.6882	-0.00238
38					1.6859	-0.00235
CPU time		30.1 sec.			52.5 sec.	

In spite of the fact that the penalty function approach can produce satisfactory solutions to many constrained problems, the disadvantages of the method that arise in conjunction with the ordinary gradient methods seem equally apparent with the conjugate gradient methods. Specifically, the characteristics of the technique that are not appealing are first, that in order to obtain the solutions to most constrained problems, an entire sequence of unconstrained problems must be solved, and second, the proper values of the penalty constants must be found by experience before an efficient solution is obtained. This latter problem stems from the unknown effect on the geometry of the cost functional due to arbitrary weighting of terminal errors in the state space. Large elements of the penalty matrix often cause poor convergence. However, if the SUMT approach is used with small penalty constants for the first several subproblems, the number of unconstrained solutions necessary to obtain the constrained solution may become excessive. Care should be taken to use the most efficient method possible to solve each subproblem. The advantage of the PCG method over the SCG method, for example, is magnified considerably when a sequence of solutions is being computed.

CHAPTER IV. SOLUTION OF OPTIMAL CONTROL PROBLEMS WITH TERMINAL STATE CONSTRAINTS USING THE CONJUGATE GRADIENT METHOD WITH A PROJECTION TECHNIQUE

Theoretical Basis of the Projection Method

In this chapter an adaptation of the conjugate gradient method is made which is equivalent to the method used by Bryson and Denham (12,18) in adapting the steepest descent technique to control problems with terminal state constraints. The method was suggested and implemented by Sinnott and Luenberger (69) for a class of control problems with linear terminal constraints. Some alterations in the method they present are given here, and the method is shown to be applicable to problems with nonlinear constraints as well.

The basic procedure of the methods discussed in this chapter is to project the gradient vectors onto a linear constraint manifold and then to determine the directions of search from the projected gradients in the standard CG fashion. Rosen (64,65,66) developed the gradient projection method for linear and nonlinear programming problems. Recently Kelley and Speyer (46) have combined the projection technique with Davidon's method for solving finite-dimensional problems. The function space projection techniques of Bryson and Sinnott and Luenberger (69) are analogous to Rosen's finite dimensional work. Although the derivation of the projection equations is outlined in Reference (69), a more complete derivation is presented here. This derivation leads to a necessary alteration in the equations given by Sinnott and Luenberger.

Consider the p constraint equations in a Euclidean n-space given by

$$A\underline{x} = 0 \tag{IV-1}$$

where A is a $p \times n$ matrix of constants and $\underline{x}$ is an n-vector of problem variables. As in Reference 66, let Q be the set of all $\underline{x}$ that satisfy (IV-1) and let $\tilde{Q}$ be its orthogonal complement. Q is an n-p dimensional space whereas $\tilde{Q}$ is p-dimensional. The gradient to the ith constraint hyperplane is given by the vector

$$\begin{pmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{pmatrix}$$

where a_{ij} is the ith row and the jth column element of A. Any vector in $\tilde{Q}$ is orthogonal to all p hyperplanes given by IV-1 and thus can be represented by a linear combination of the rows of A. If a change $\Delta\underline{x}$ in the vector $\underline{x}$ is sought which is orthogonal to Q, then $\Delta\underline{x}$ can be written as a linear combination of the rows of A, i.e.

$$\Delta\underline{x} = c_1 \begin{pmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1n} \end{pmatrix} + c_2 \begin{pmatrix} a_{21} \\ a_{22} \\ \vdots \\ a_{2n} \end{pmatrix} + \cdots c_p \begin{pmatrix} a_{p1} \\ a_{p2} \\ \vdots \\ a_{pn} \end{pmatrix} \tag{IV-2}$$

$$= A^T \underline{c} \tag{IV-3}$$

Thus if $\underline{x}$ does not lie in Q but $\underline{x}+\Delta\underline{x}$ does we have

$$A\underline{x} = \underline{\Psi} \tag{IV-4}$$

where $\underline{\Psi}$ is the constraint error vector and

$$A(\underline{x} + \Delta\underline{x}) = \underline{0} \tag{IV-5}$$

or

$$A\Delta\underline{x} = -\underline{\Psi} \tag{IV-6}$$

$$AA^T\underline{c} = -\underline{\Psi} \tag{IV-7}$$

so that

$$\underline{c} = -[AA^T]^{-1}\underline{\Psi} \quad . \tag{IV-8}$$

From IV-3 we obtain

$$\Delta\underline{x} = -A^T[AA^T]^{-1}\underline{\Psi} \tag{IV-9}$$

so

$$\underline{x} + \Delta\underline{x} = \underline{x} - A^T[AA^T]^{-1}\underline{\Psi} \tag{IV-10}$$

$$= \underline{x} - A^T[AA^T]^{-1}A\underline{x} \quad . \tag{IV-11}$$

Therefore if $\bar{\underline{z}}$ is the projection of a vector $\underline{z}$ onto the linear constraint manifold then

$$\bar{\underline{z}} = \underline{z} - A^T[AA^T]^{-1}A\underline{z} \quad . \tag{IV-12}$$

The function space analog of IV-12 can now be derived. Linearization of the system dynamics given by

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, t) \tag{IV-13}$$

about a nominal control produces the system

$$\Delta\dot{\underline{x}} = \underline{f}_{\underline{x}}\Delta\underline{x} + \underline{f}_{\underline{u}}\Delta\underline{u} \tag{IV-14}$$

$$\Delta\underline{x}(t_0) = 0 \tag{IV-15}$$

where

$$\underline{f}_{\underline{x}} = \frac{\partial \underline{f}}{\partial \underline{x}}, \qquad \underline{f}_{\underline{u}} = \frac{\partial \underline{f}}{\partial \underline{u}} \quad . \tag{IV-16}$$

The solution form for this system is

$$\Delta \underline{x}(t) = \int_{t_0}^{t} \Phi(t,\tau)\, \underline{f}_{\underline{u}}(\tau)\, \Delta \underline{u}(\tau)\, d\tau \tag{IV-17}$$

$$= S[\Delta u(t)] \tag{IV-18}$$

where S is the linear operator defined by IV-17 and $\Phi(t,\tau)$ is the state transition matrix for the linear system IV-14. Suppose the optimal control problem has terminal state constraints of the form

$$A\, \underline{x}(t_f) = 0 \tag{IV-19}$$

where A is a $p \times n$ constant matrix. If $\underline{x}(t_f)$ does not satisfy the constraint but $(\underline{x}+\Delta\underline{x})_{t_f}$ does, i.e. if

$$A\, \underline{x}(t_f) = \underline{\Psi} \tag{IV-20}$$

and

$$A(\underline{x}(t_f) + \Delta\underline{x}(t_f)) = 0\,, \tag{IV-21}$$

then

$$A\, \Delta\underline{x}(t_f) = -\underline{\Psi}\,, \tag{IV-22}$$

or

$$\int_{t_0}^{t_f} A\, \Phi(t_f,\tau)\, \underline{f}_{\underline{u}}(\tau)\, \Delta u(\tau)\, d\tau = -\underline{\Psi}\,. \tag{IV-23}$$

We desire that Equation IV-23 be the analog of Equation IV-6. To establish the analogy, we may regard IV-6 as

$$-\underline{\Psi} = \begin{pmatrix} \langle A_{R1}, \Delta\underline{x}\rangle \\ \langle A_{R2}, \Delta\underline{x}\rangle \\ \vdots \\ \langle A_{Rp}, \Delta\underline{x}\rangle \end{pmatrix} \tag{IV-24}$$

where A_{Rj} is the jth row of A. Now using the inner product given by II-36, IV-23 may be written as

$$-\underline{\Psi} = \begin{pmatrix} \langle (A\,\Phi(t_f,t)\,\underline{f}_{\underline{u}}(t))_{R1}, \Delta\underline{u}(t)\rangle \\ \langle (A\,\Phi(t_f,t)\,\underline{f}_{\underline{u}}(t))_{R2}, \Delta\underline{u}(t)\rangle \\ \vdots \\ \langle (A\,\Phi(t_f,t)\,\underline{f}_{\underline{u}}(t))_{Rp}, \Delta\underline{u}(t)\rangle \end{pmatrix} \qquad \text{(IV-25)}$$

The analogy between IV-24 and IV-25 is complete if:

1. A in IV-6 is replaced by

$$\tilde{A} = A\,\Phi(t_f,t)\,\underline{f}_{\underline{u}}(t) \qquad \text{(IV-26)}$$

2. $\underline{\chi}$ is replaced by $\Delta\underline{u}(t)$
3. A multiplied by a member of the $\underline{\chi}$-space is replaced by the inner products of the rows of $\tilde{A}$ with the analogous member of the $\underline{u}$-space.

The analog of IV-9 becomes

$$\Delta\underline{u}(t) = -\tilde{A}^{T} \begin{bmatrix} \langle \tilde{A}_{R1}, \tilde{A}_{R1}\rangle & \langle \tilde{A}_{R1}, \tilde{A}_{R2}\rangle & \cdots \\ \vdots & & \\ \langle \tilde{A}_{Rp}, \tilde{A}_{R1}\rangle & \cdots & \langle \tilde{A}_{Rp}, \tilde{A}_{Rp}\rangle \end{bmatrix}^{-1} \underline{\Psi} \qquad \text{(IV-27)}$$

It is important to note that in IV-27 $\underline{\Psi}$ is a p-vector of constants. The projection $\bar{\underline{z}}(t)$ of an element $\underline{z}(t)$ is given by

$$\bar{\underline{z}}(t) = \underline{z}(t) + \Delta\underline{z}(t) \qquad \text{(IV-28)}$$

$$= z(t) - \tilde{A}^{T}\left[\int_{t_0}^{t_f} \tilde{A}\tilde{A}^{T}\,dt\right]^{-1}\left[\int_{t_0}^{t_f} \tilde{A}\,\underline{z}(t)\,dt\right] \qquad \text{(IV-29)}$$

The final factor in IV-29,

$$\int_{t_0}^{t_f} \tilde{A}\, \underline{z}(t)\, dt = \langle \tilde{A}, \underline{z}(t) \rangle ,$$

represents a transformation from the Hilbert space containing $\underline{z}(t)$ to the real Euclidean space $\mathcal{E}_p$. It is a p-vector of constants just as $\underline{\Psi}$ is a p-vector of constants in IV-27. Therefore, IV-29 is the proper analog to IV-12. This result differs from that obtained in (69) which is

$$\bar{\underline{z}}(t) = \underline{z}(t) - \tilde{A}^T \left[\int_{t_0}^{t_f} \tilde{A}\, \tilde{A}^T \right]^{-1} \tilde{A}\, \underline{z}(t) \qquad \text{(IV-30)}$$

The final factor in IV-30 is not a constant p-vector and does not satisfy the third point in the analogy established earlier. However, the equivalence of the projection formula given in IV-29 to that given by Bryson (12,56) is easily established. The differences lie only in notation.

Application of the Projection Theory to the Conjugate Gradient Method

With the substitution of IV-29 for IV-30, the algorithm given by Sinnott and Luenberger is the following:

1. Choose an initial control function $\underline{u}_0(t)$
2. Integrate forward the state equations

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, t) \quad , \quad \underline{x}(t_0) = \underline{x}_0 \qquad \text{(IV-31)}$$

and the auxiliary equations

$$\dot{P}(t) = \underline{f}_{\underline{u}}\, \underline{f}_{\underline{u}}^T + \underline{f}_{\underline{x}} P + P \underline{f}_{\underline{x}}^T \quad , \quad P(t_0) = 0 \qquad \text{(IV-32)}$$

3. Integrate backward the adjoint equations

$$\dot{\underline{\lambda}}(t) = -\underline{f}_{\underline{x}}^T \underline{\lambda} - F_{\underline{x}}^T \quad , \quad \underline{\lambda}(t_f) = \Phi_{\underline{x}}(t_f) \qquad \text{(IV-33)}$$

and the auxiliary equations

$$\dot{\Lambda}(t) = -f_{\underline{x}}^{T}\Lambda, \qquad \Lambda(t_f) = A^{T} \tag{IV-34}$$

4. Compute the gradient function

$$g_i(t) = \frac{\partial H}{\partial \underline{u}} = f_{\underline{u}}^{T}\Delta + F_{\underline{u}}^{T} \tag{IV-35}$$

5. Project the gradient via the formula

$$\bar{g}_i(t) = g_i(t) - f_{\underline{u}}^{T}\Lambda\hat{M}\int_{t_0}^{t_f}\Lambda^{T} f_{\underline{u}}\, g_i\, dt \tag{IV-36}$$

where

$$\hat{M} = \left[A\,P(t_f)\,A^{T}\right]^{-1} \tag{IV-37}$$

6. Project the previous direction of search

$$\bar{\xi}_{i-1}(t) = \xi_{i-1} - f_{\underline{u}}^{T}\Lambda\hat{M}\int_{t_0}^{t_f}\Lambda^{T} f_{\underline{u}}\,\xi_{i-1}\, dt \tag{IV-38}$$

7. Calculate the conjugate gradient parameter β_i

$$\beta_i = \frac{\langle \bar{g}_i, \hat{N}\,\bar{\xi}_{i-1}\rangle}{\langle \bar{\xi}_{i-1}, \hat{N}\,\bar{\xi}_{i-1}\rangle} \tag{IV-39}$$

where

$$\hat{N}\,\bar{\xi}_{i-1} = f_{\underline{u}}^{T}\eta + F_{\underline{u}\underline{x}}\,y + F_{\underline{u}\underline{u}}\,\bar{\xi}_{i-1} \tag{IV-40}$$

$$\dot{y}(t) = f_{\underline{x}}\,y + f_{\underline{u}}\,\bar{\xi}_{i-1}, \qquad y(t_0) = 0 \tag{IV-41}$$

$$\dot{\eta}(t) = -f_{\underline{x}}\,\eta - F_{\underline{x}\underline{x}}\,y - F_{\underline{x}\underline{u}}\,\bar{\xi}_{i-1}, \qquad \eta(t_f) = \left.\frac{\partial^2\phi}{\partial\underline{x}^2}\right|_{t_f} y(t_f) \tag{IV-42}$$

$$\text{if } i = 0, \quad \beta_0 = 0 \tag{IV-43}$$

8. Calculate the direction of search

$$\underline{s}_i(t) = -\bar{g}_i + \beta_i \bar{s}_{i-1} \tag{IV-44}$$

9. Perform a one-dimensional minimization to determine α_i i.e.

$$J(\underline{u}_i + \alpha_i \underline{s}_i) \leqslant J(\underline{u}_i + \gamma \underline{s}_i) \quad \forall \gamma > 0$$

10. Take a forward step 'parallel' to the constraint.

$$\hat{\underline{u}}_{i+1} = \underline{u}_i + m_i \alpha_i \underline{s}_i \tag{IV-45}$$

m_i is a stepsize adjustment parameter discussed below.

11. Compute $\underline{\Psi}_i$

$$\underline{\Psi}_i = A \hat{\underline{x}}_i(t_f) \tag{IV-46}$$

where $\hat{\underline{x}}_i(t)$ is obtained from $\hat{\underline{u}}_{i+1}(t)$

12. Compute the control correction in a direction orthogonal to the constraint

$$\Delta \hat{\underline{u}}_i(t) = -\underline{f}_u^T \Lambda \hat{M} \underline{\Psi}_i \tag{IV-47}$$

13. Correct the control

$$\underline{u}_{i+1}(t) = \hat{\underline{u}}_{i+1} + n_i \Delta \hat{\underline{u}}_i \tag{IV-48}$$

where n_i is another stepsize adjustment parameter discussed below.

14. Repeat from step 2 until the minimum is reached.

The algorithm above represents an adaptation of the PCG method. The choice of this conjugate gradient method instead of the SCG method is justified by the superiority of the PCG method for unconstrained problems.

The stepsize adjustment parameters appearing in steps 10 and 13 are

necessary because of the linearization of the system dynamics in Equation IV-14. If changes in the control are too large for the linearization to be valid, the projection equations become inaccurate. Similar problems are present when the projection technique is used with the steepest descent method. With the latter method however, the solution is merely to restrict stepsizes to the degree necessary for the linearization to be valid. After a forward step has been taken in step 10, the constraint violation can be compared with that before the step was taken. If the direction of search is 'parallel' to the constraint, and the linearity assumptions have not been violated, $\underline{\Psi}$ values should be nearly equal (56). If however, the constraint violations are significantly different, linearization has been violated, and a smaller stepsize is necessary. The parameter m_i must be reduced from its original value of 1.0. Similarly, a linearization check can be made on the correction step by comparing the actual change in the constraint violation $\Delta\underline{\Psi}$ after step 13 to the quantity $A\Delta\underline{x}(t_f)$ which is calculated by integrating the linearized state Equations IV-14. The adjustment parameter n_i is reduced if a significant difference occurs between $\Delta\underline{\Psi}$ and $A\Delta\underline{x}(t_f)$. The stepsize adjustment philosophy just described is similar to that used with steepest descent and will be referred to here as stepsize adjustment policy I (SAP-I).

Although Mehra and Bryson (56) suggest its use with the CG method, they report no computational experience using SAP-I with that technique. A very basic difference between steepest descent and the CG method raises doubts about the merits of using SAP-I with the latter technique. With steepest descent, stepsizes are usually chosen by an automated trial

process. Changes in the control history are constrained isoperimetrically, i.e. the control change $\Delta \underline{u}(t)$ must satisfy the constraint

$$dP^2 = \int_{t_0}^{t_f} \Delta \underline{u}^T W \Delta \underline{u} \, dt \tag{IV-49}$$

where W is a weighting matrix and dP^2 is a positive constant. The quantity dP^2 is chosen small enough so that the linearized theory of the projection method remains valid. The theory of the conjugate gradient method however does not permit alteration of the stepsize. The rapid convergence of the method depends upon completing a one-dimensional minimization along each direction of search. Thus an adjustment of the stepsize to accommodate the linearity assumptions could be expected to degrade the rate of convergence of the method. If the stepsizes chosen automatically by the CG logic do not violate the linearity assumptions however, the projection method becomes compatible with the CG method. Presumably, the linearity assumption becomes better as the control approaches the optimal control.

A different stepsize adjustment policy is suggested here for use with conjugate gradient methods. Instead of reducing the stepsize adjustment parameter m_i until linear approximations are accurate, this policy reduces m_i from 1.0 only if

1. The 'correction' in 13 leads to greater rather than smaller constraint violation, or
2. The value of the cost functional after correcting the control in 13 becomes larger than the cost before the forward step was taken (step 10).

This stepsize adjustment philosophy is referred to as SAP-II. Other authors[a,b] who have proposed the use the projection method have not reported an automatic policy for choosing m_i and n_i. The advantage of SAP-II is that larger stepsizes can be taken, thus permitting rapid convergence. A disadvantage is that even though the correction equation (IV-48) may be sufficiently accurate to move the control toward the constraint rather than away from it, the approach could be very slow because of a poor approximation produced by the linearized state equations. To improve this deficiency, several small corrections can be used in SAP-II instead of one single attempt to reduce $\underline{\Psi}$. This can be accomplished by choosing n_i such that $0 < n_i < 1.0$, and then executing steps 12 and 13 alternately until the constraint violation is within acceptable limits. Logic flow diagrams of the alternate stepsize adjustment policies are given as Figures 6 and 7.

Numerical Solutions Using the Conjugate Gradient-Projection Method

Both SAP-I and SAP-II were used to solve the Van der Pol problem P-6 presented in Chapter III. Acceptable linearity violations using SAP-I were set at 10%. That is, if the constraint violations that occurred before and after step 10 differed by more than 10%, the parameter

---

[a]Mehra, R. K. The Analytic Sciences Corporation, Reading, Massachusetts. Private communication regarding the choice of values for the stepsize parameters. June 6, 1969.

[b]Luenberger, D. G. Stanford University, Stanford, California. Private Communication regarding the choice of values for the stepsize parameters. July, 1969.

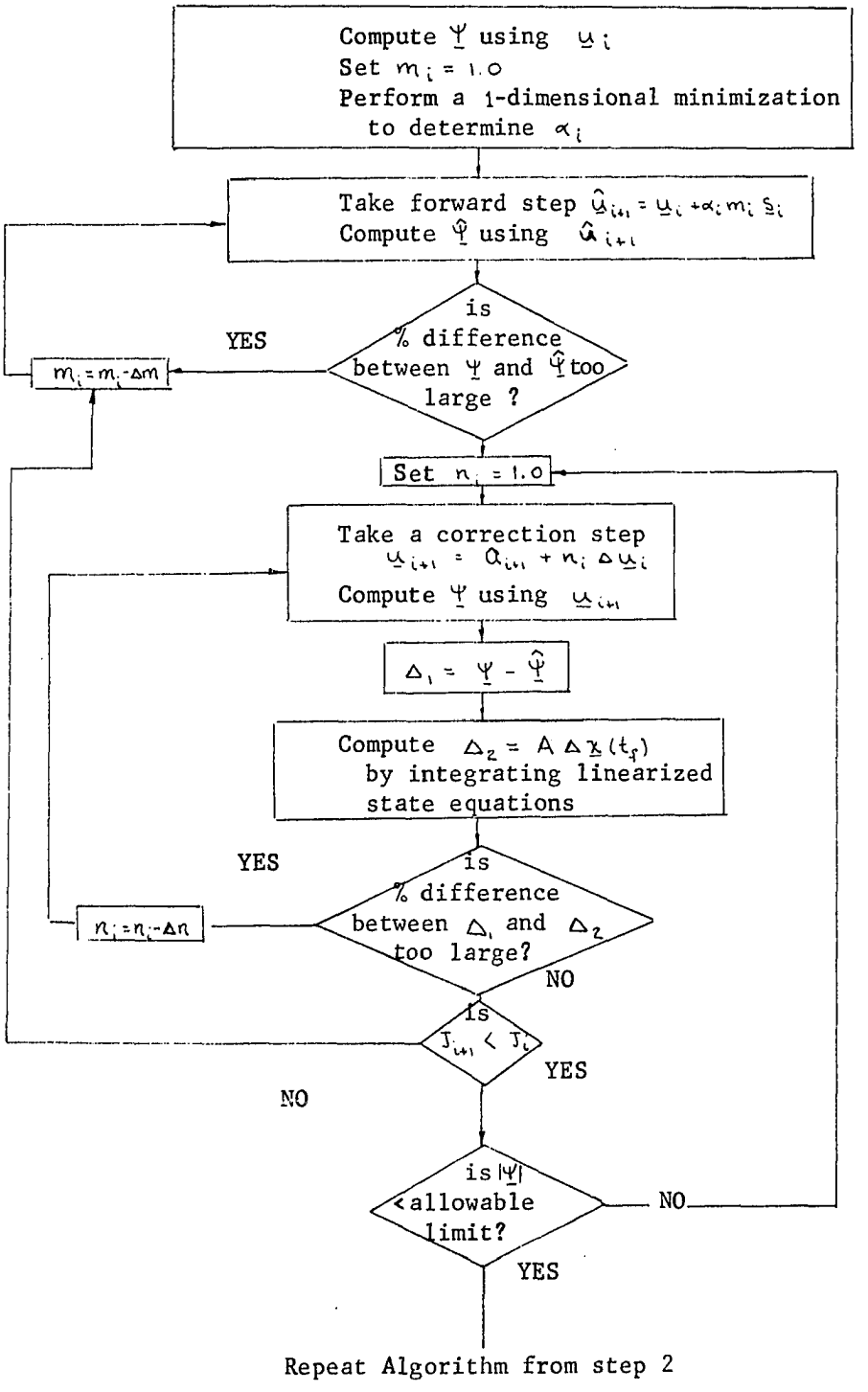
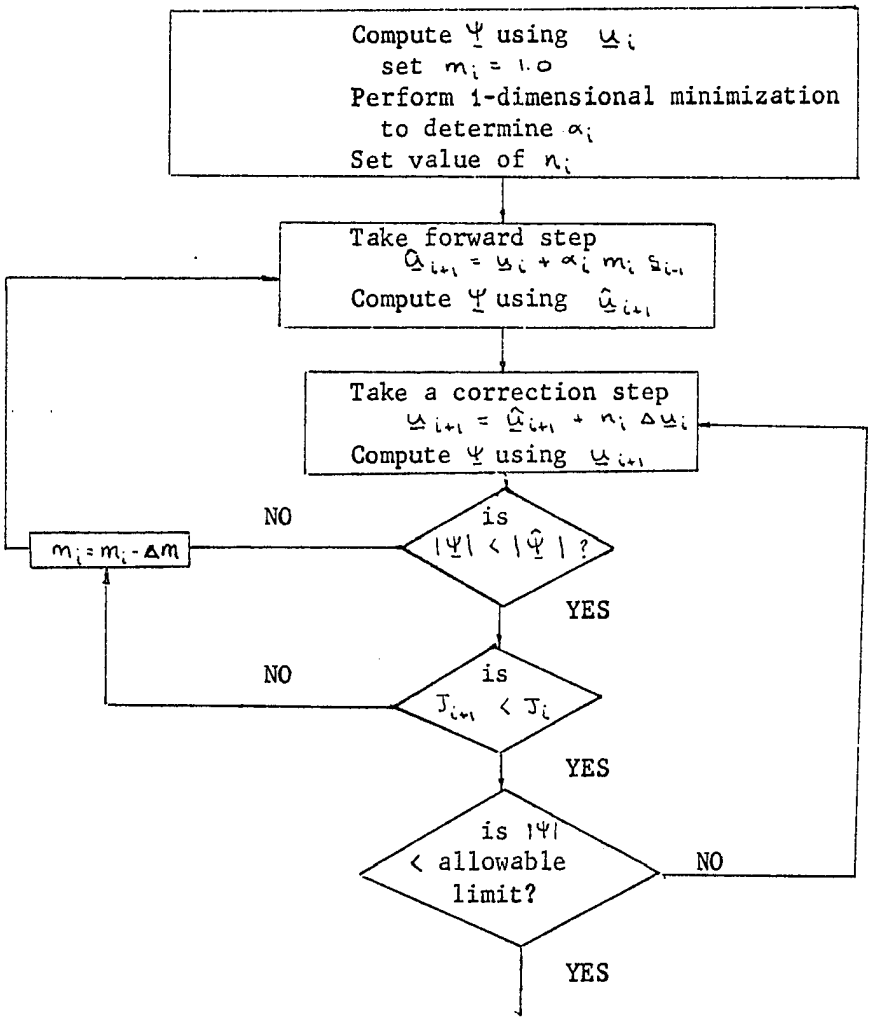


Figure 6. Logic flow diagram for SAP-I

Step 8



Repeat Algorithm from step 2

Figure 7. Logic flow diagram for SAP-II

m_i was reduced and step 10 repeated. Similarly, if the control change $\Delta\hat{u}_i$ in step 13 caused the change in the constraint violation $\Delta\underline{\Psi}$ to differ by more than 10% from $A\Delta\underline{x}(t_f)$, the parameter n_i was reduced and step 13 repeated. Table 11 gives the results for the first ten iterations as well as Sinnott and Luenberger's results (69). No information is available concerning the values of m_i and n_i used by those authors. The solution was started with the control $u_o(t) = 0$. Figure 8 gives the control iterates and the optimal control given by Birta and Trushel (8).

It can be seen that stringent limits on the linearity checks cause stepsize restrictions that are unnecessarily severe. The convergence of the SAP-II solution, which used full steps both in the forward direction and on the correction steps, compares favorably with the solutions given in Chapter III and in References (8) and (69). It should be noted that the maximum linearity violation that occurred using SAP-I was less than approximately 100% on the forward step and less than 30% on the correction step. Relaxation of the violation limits to those levels would have produced results identical to the SAP-II solution.

Both solutions given in Table 11 used Equation IV-29. When Equation IV-30 was used, the convergence was extremely slow.

Although the terminal state constraints for this problem are linear in the state space, they are not linear in the control space. This is due to the nonlinear dynamics that define the relationship between the control function and the state trajectories. Figure 9 shows the percentage change in Ψ along the direction of search from the second iteration on the Van der Pol problem using SAP-I. Enforcing adherence

Table 11. Solution of the Van der Pol problem P-6 using the CG method with projection

Iteration Number	SAP-I		SAP-II		Solution from Reference (69)	
	J	Ψ	J	Ψ	J	Ψ
0	7.4781	0.6313	7.4781	0.6313	7.4780	0.6313
1	4.2684	0.5932	2.2228	-0.04083	2.6584	0.1457
2	4.8778	-0.09158	1.7276	-0.006006	2.4580	-0.0153
3	4.4264	-0.005444	1.7108	$\lvert\Psi\rvert < 10^{-4}$	2.2338	0.00827
4	4.3015	$\lvert\Psi\rvert < 10^{-4}$	1.7012	$\lvert\Psi\rvert < 10^{-7}$	1.8287	-0.0331
5	3.9808	$\lvert\Psi\rvert < 10^{-4}$	1.6995	$\lvert\Psi\rvert < 10^{-7}$	1.7850	-0.000503
6	3.8744	$\lvert\Psi\rvert < 10^{-5}$	1.6933	$\lvert\Psi\rvert < 10^{-7}$	1.6944	0.00426
7	3.5774	$\lvert\Psi\rvert < 10^{-3}$	1.6921	$\lvert\Psi\rvert < 10^{-7}$	1.6874	-0.000368
8	3.4892	$\lvert\Psi\rvert < 10^{-5}$	1.6883	$\lvert\Psi\rvert < 10^{-6}$	1.6861	$\lvert\Psi\rvert < 10^{-4}$
9	3.2393	$\lvert\Psi\rvert < 10^{-3}$	1.6881	$\lvert\Psi\rvert < 10^{-7}$	1.6860	$\lvert\Psi\rvert < 10^{-4}$
10	3.1650	$\lvert\Psi\rvert < 10^{-5}$	1.6869	$\lvert\Psi\rvert < 10^{-7}$	1.6853	$\lvert\Psi\rvert < 10^{-4}$

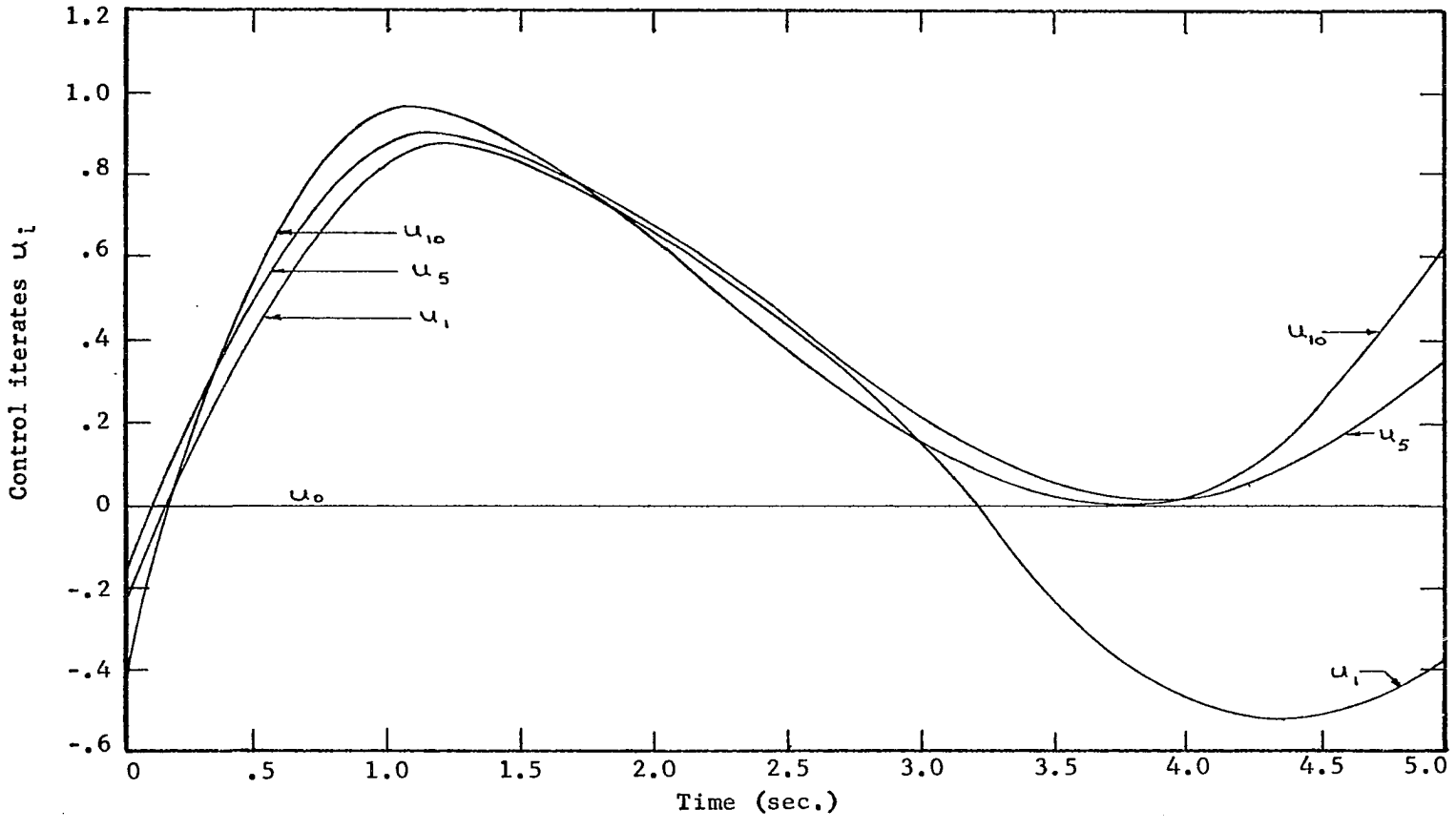


Figure 8. Control iterates in the solution of problem P-6

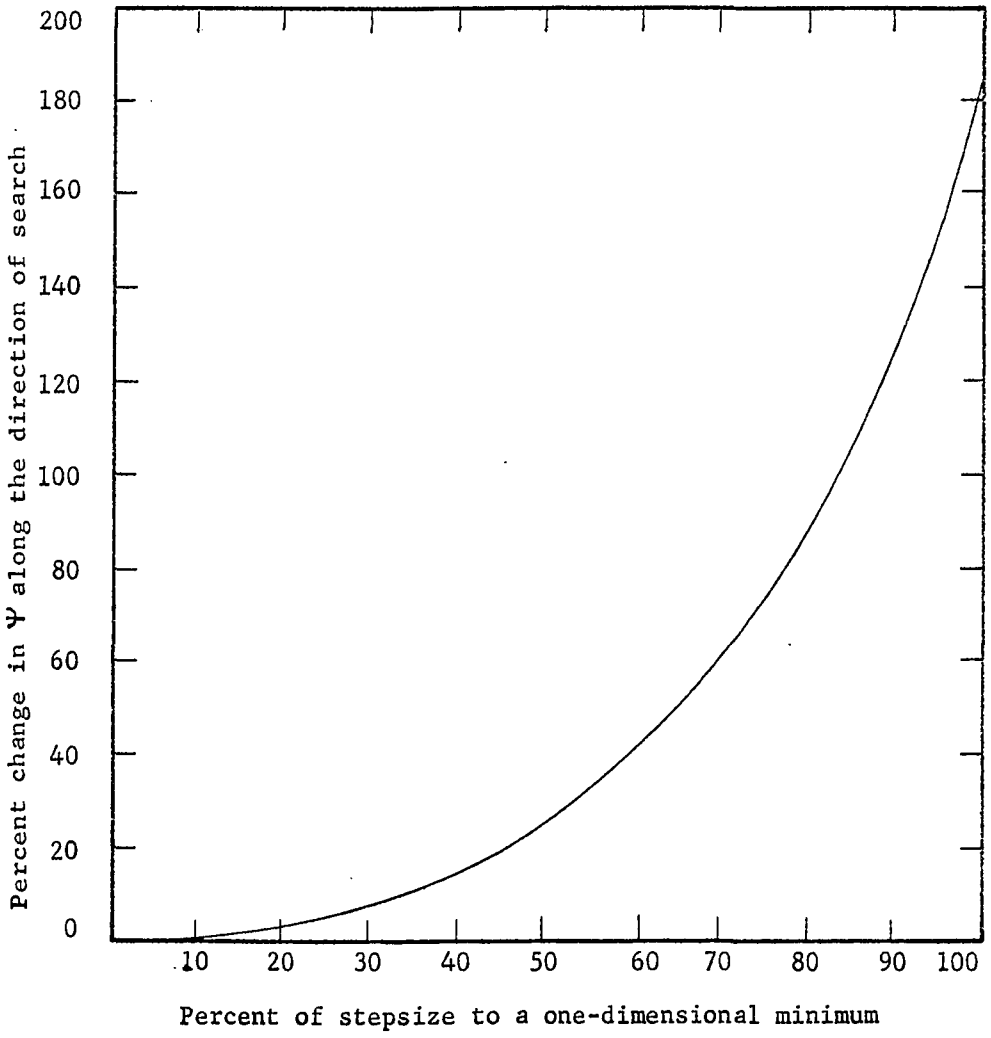


Figure 9. Constraint nonlinearity in problem P-6

to the 10% linearity violation limit would have allowed a maximum of 32% of the optimum stepsize along the direction of search. The fact that the curve approaches zero slope for very small stepsizes is in agreement with the notion that the direction of search is 'parallel' to the constraint at the point from which the step is initiated. However, the constraint violations do not remain the same for all controls along the direction of search but instead increase rapidly as larger stepsizes are chosen.

To demonstrate the point that linear system dynamics as well as linear constraints are sufficient to make the projection and correction equations valid for large stepsizes, a very simple control problem is presented. A mathematical statement of the problem is:

P-7. Minimize $J = \phi(\underline{x}(1)) = -x_2(1) + x_3(1)$ (IV-50)

subject to $\dot{x}_1 = x_2$ (IV-51)

$\dot{x}_2 = -x_2 + u$ (IV-52)

$\dot{x}_3 = u^2$ (IV-53)

$x_1(0) = x_2(0) = x_3(0) = 0$ (IV-54)

$x_1(1) = 5$ (IV-55)

where x_1 = position

x_2 = velocity

and x_3 = variable used to create the Mayer formulation.

Problem P-7 may be thought of as a physical problem whose objective is to maximize the velocity of a unit mass by determining the forcing function $u(t)$ that must act on it for one second. The final position of the mass is constrained and the integral squared force is penalized.

The mass slides on a surface with viscous friction, and the friction coefficient is the reciprocal of the gravitational constant. The physical system for this problem is identical to that of problem P-1. However, the objective here differs from that of problem P-1.

The projection method starting from the control $u_0(t) = 50 - 50t$ converged to the analytical solution in essentially one step. Table 12 gives the values of the cost functional J and the terminal position of the mass $x_1(1)$ using controls $u_0(t)$ and $u_1(t)$. The simplicity of the problem accounts in part for the rapid convergence. This single-step solution does demonstrate however that application of the projection method to this problem with linear dynamics and a linear constraint produces an initial direction of search that is parallel to the constraint. This is evidenced by the fact that the constraint violation after the control change of step 10 was virtually eliminated by the control correction of step 13 with $n_i = 1.0$.

Table 12. Solution of the unit mass problem P-7 with the CG-projection method

Iteration Number i	J	$x_1(t_f)$	α_i
0	820.12	11.787941	
1	142.74	5.000067	0.49881
	$J^* = 143.78$		

Extension of the Method to Problems With Nonlinear Terminal Constraints

The previous argument suggests that the projection method could be applied to optimal control problems having nonlinear terminal constraints

$$\underline{\Omega}\,(\underline{x}(t_f)) = 0, \tag{IV-56}$$

where $\underline{\Omega}$ is a p-vector, merely by linearizing the constraint expression about the current search point. This gives

$$\underline{\Omega}(\underline{x}(t_f)) \cong \underline{\Omega}(\underline{x}_i(t_f)) + \left.\frac{\partial \underline{\Omega}}{\partial \underline{x}}\right|_{\underline{x}_i(t_f)} \Delta \underline{x} \quad . \tag{IV-57}$$

Then $\left.\frac{\partial \underline{\Omega}}{\partial \underline{x}}\right|_{\underline{x}_i(t_f)}$ replaces the matrix A, and IV-22 becomes

$$-\underline{\Psi} = \left.\frac{\partial \underline{\Omega}}{\partial \underline{x}}\right|_{\underline{x}_i(t_f)} \Delta \underline{x}(t_f) \tag{IV-58}$$

where $\underline{x}_i(t_f)$ is determined using the control about which the state dynamics are linearized. Although such a procedure adds another approximation to the problem, it does not cause additional difficulties conceptually since in problems with nonlinear dynamics, the constraint is nonlinear in the control space whether its form in the state space is linear or not. It should be noted that no linearity requirement on the constraints is made when the projection method is used with the steepest descent method (12).

For the purposes of numerical example, the terminal constraints on the Van der Pol problem were modified as shown in Figure 10. The linear constraint was replaced by a parabolic constraint which was tangent to the linear constraint at the solution point to the original problem. The constraint relation takes the form

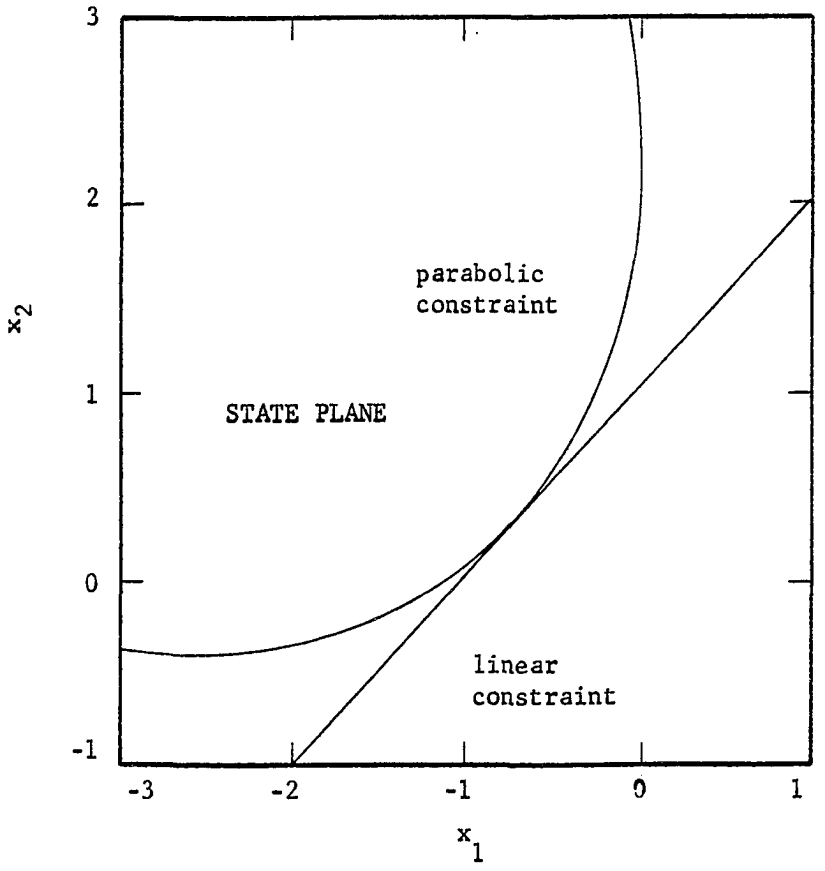


Figure 10. Terminal constraint curves for problem P-6

$$\Omega(\underline{x}(t_f)) = -\frac{1}{2K}\left[x_1(t_f)^2 + x_2(t_f)^2\right] - \frac{1}{K}\left[x_1(t_f)\,x_2(t_f)\right]$$
$$+ x_1(t_f)\left[c_1 + \frac{p_1+p_2}{K}\right] + x_2(t_f)\left[-c_1 + \frac{p_1+p_2}{K}\right]$$
$$c_1 p_2 - c_1 p_1 - \frac{1}{2K}\left[p_1^2 + p_2^2\right] - \frac{1}{K}p_1 p_2$$

(IV-59)

where K is a constant determining the curvature of the parabola, p_1 and p_2 are the coordinates of the vertex of the parabola and $c_1 = \frac{\sqrt{2}}{2}$. The Van der Pol problem with the nonlinear constraint given by IV-59 will be called problem P-8. The numerical solution to the Van der Pol problem with the nonlinear constraint of Equation IV-59 is given in Table 13. Stepsize adjustment policy II was used in the solution. A penalty function solution of this problem produced an optimal control and a minimum value of the cost functional that were virtually the same values as those obtained here and elsewhere for the Van der Pol problem with the linear constraint. Computationally, the use of the nonlinear constraint instead of the linear constraint caused no apparent additional difficulties. In both solutions, the stepsize parameters m_i and n_i were not reduced by SAP-II from their original values of 1.0. The convergence rates were also comparable. These solution data tend to substantiate the claim that the projection method with a stepsize adjustment policy that is workable for problems with nonlinear dynamics is applicable to problems with terminal constraints that are either linear or nonlinear. It can be speculated however, that the linear approximations to the constraints could, in general, cause the method to take smaller step sizes and make less accurate corrections than it would if the constraints had a linear representation in the state space.

Table 13. Solution of the Van der Pol problem P-8

Iteration Number	J	Ψ
0	7.4781	-0.7270
1	2.0979	-0.1931
2	2.0767	-0.1956
3	1.8710	-0.01368
4	1.8412	-0.001153
5	1.8323	-0.000132
6	1.6935	-0.001025
7	1.6907	$\lvert\Psi\rvert < 10^{-4}$
8	1.6905	0.000156
9	1.6862	$\lvert\Psi\rvert < 10^{-5}$
10	1.6862	$\lvert\Psi\rvert < 10^{-4}$

$K = -4 \qquad p_1 = -.2293 \qquad p_2 = .7707$

CHAPTER V. A MODIFIED CONJUGATE GRADIENT METHOD FOR SOLVING CONSTRAINED MINIMIZATION PROBLEMS

Development and Application of the Method in Finite-Dimensional Spaces

Both the penalty function and the projection methods have properties that detract from either their theoretical or computational attractiveness. These disadvantages have been discussed in the previous chapters. Neither technique is a totally satisfactory solution to the problem of adapting basic conjugate gradient methods to control problems with terminal state constraints. This chapter reports another attempt to adapt the CG ideas to a solution method for that class of problems. Although the method also possesses disadvantages principally of a computational nature, it avoids some of the difficulties encountered in using the other two methods. In addition, it represents an approach that is substantially different from other methods reported in the literature.

The use of conjugate gradient methods on unconstrained control problems is an extension to function space of an algorithm originally presented as a means of minimizing a function on a finite-dimensional vector space. Similarly, the method reported here is an extension of a scheme to solve constrained finite-dimensional minimization problems. Both the finite and the infinite dimensional versions will be referred to as the modified conjugate gradient method (MCG). Explanation of the method is most lucid in terms of a finite-dimensional minimization.

Application of the method to optimal control problems follows with only minor alterations to the algorithm.

The penalty function approach to constrained problems involves adding to the objective function $f(\underline{x})$ some positive measure of the constraint violation, for example

$$\hat{f}(\underline{x}) = f(\underline{x}) + \underline{\Psi}^T W \underline{\Psi} \tag{V-1}$$

where $\underline{x}$ is an n-vector of independent variables.

$$\underline{\Psi} = \underline{\Omega}(\underline{x}) \neq \underline{0} \tag{V-2}$$

is the constraint violation for $\underline{x} \neq \underline{x}^*$,

$$\underline{\Omega}(\underline{x}^*) = \underline{0} \tag{V-3}$$

is a set of p constraint equations, and W is a $p \times p$ positive definite matrix of penalty constants. A solution method that attempts to minimize the function $\hat{f}$ requires the choice of values for the penalty constants in W. As pointed out in Chapter III, the choice of W may be difficult due to the sensitivity of the method to the geometric properties of $\hat{f}$. To avoid this problem, one might consider the classical Lagrangian function defined as

$$\tilde{f} = f + \underline{\Lambda}^T \underline{\Omega}(\underline{x}) \tag{V-4}$$

where $\underline{\Lambda}$ is a p-vector of constant Lagrange multipliers. The appearance of V-4 is similar to V-1 since when $\underline{x}$ does not satisfy the constraint $\underline{\Omega}(\underline{x}) = \underline{\Psi}$. However the mathematical differences are significant. Although Thrasher (71) has shown a relation between Lagrange multipliers and penalty constants for certain problems involving control variable inequality constraints, in general the penalty constants can seldom be related to Lagrange multipliers. The expression $\underline{\Omega}(\underline{x})$ appears in the

Lagrangian $\tilde{f}$ instead of an arbitrary positive measure of $\underline{\Psi}$, the constraint violation. The classical approach regards $\underline{\Lambda}$ as p additional unknowns and uses the constraint Equation V-3 to provide p additional equations. The values of $\underline{x}$ minimizing f subject to $\underline{\Omega}(\underline{x}^*)=0$ are among the zeroes of the system

$$\frac{\partial \tilde{f}}{\partial \underline{x}} = \underline{0} \tag{V-5}$$

$$\frac{\partial \tilde{f}}{\partial \underline{\Lambda}} = \underline{0} \tag{V-6}$$

Precise values of the components of $\underline{\Lambda}$ are obtained as by-products in the solution of this system. Although the zeroes of V-5 and V-6 may not be unique, the $\underline{\Lambda}$ corresponding to the solution point $\underline{x}^*$ is unique. Thus from a computational point of view, it might be advantageous to consider the application of a CG method to the Lagrangian $\tilde{f}$ in perference to the function $\hat{f}$ which involves the arbitrary penalty constants in W.

Forsythe (27) points out that care must be taken in attempting to apply a direct descent technique such as steepest descent or conjugate gradients to a Lagrangian function. Difficulties can result from the fact that the vector $\underline{x}^*$ that minimizes f subject to the constraints $\underline{\Omega}(\underline{x})=0$ may lie at a stationary point of the Lagrangian $\tilde{f}$ which is neither a relative maximum nor minimum in the n-dimensional space. That is, at the constrained minimum, the following relationship may not hold

$$\tilde{f}(\underline{x}^*, \underline{\Lambda}^*) \leqslant \tilde{f}(\underline{x}, \underline{\Lambda}^*) \tag{V-7}$$

Stated in still another way, the matrix of second partial derivatives $B = \frac{\partial^2 \tilde{f}}{\partial \underline{x}^2}(\underline{x}^*, \underline{\Lambda}^*)$ may not be sign definite. In that case, a descent

method could have difficulty locating the constrained minimum. Forsythe (27) and Arrow and Solow (2) point out however, that if B is positive definite for all vectors $\underline{x}$ satisfying the linearized constraint conditions

$$\left.\frac{\partial \underline{\Omega}}{\partial \underline{x}}\right|_{\underline{x}=\underline{x}^*} (\underline{x}-\underline{x}^*) = \underline{0}, \tag{V-8}$$

the constrained minimum represents a relative minimum in the $(n-p)$-dimensional space given by the intersection of the p constraint equations $\underline{\Omega}(\underline{x}) = 0$. Therefore, a descent technique avoids the difficulty of seeking inflection points if the search is carried out within the constraint space. This situation is roughly analogous to that which would be obtained if the constraint equations could be solved explicitly for p of the independent variables. Substitution of these relations into the unconstrained objective function would result in a function of $n-p$ variables to be minimized without side constraints.

Searching within the constraint space is attempted in the projection methods by projecting the search directions onto the support hyperplane to the constraint space, searching along this projected direction of search and finally correcting any constraint violations that occur as a result of the 'curvature' of the constraint due to its nonlinearity. Since the projection and correction relations are based on linearization, stepsizes must be restricted to preserve their validity. As pointed out in Chapter IV, any method such as the conjugate gradient method that often seeks to take large stepsizes is in basic conflict with the philosophy of the projection methods for problems with insufficient linearity.

A method of adapting the CG method is sought which avoids the necessity of choosing arbitrary weighting or penalty constraints, which searches entirely within the constraint space, and which is not restricted to small stepsizes for nonlinear problems. The algorithm given by Equations II-2-II-9 in Chapter II will be applied formally to the Lagrangian V-4 which is considered to be a function of the components of $\underline{x}$ only. The ρ components of $\underline{\Lambda}$ are parameters that are computed at the point in the algorithm where a one-dimensional minimization is done in the unconstrained version. Specifically the MCG algorithm is stated as follows:

1. For $i = 0$, guess an initial state vector $\underline{x}_0$
2. Calculate the gradient vector $g_i(\underline{\Lambda}_i)$ at $\underline{x}_i$

$$g_i(\underline{\Lambda}_i) = \nabla f(\underline{x}_i) + \frac{\partial \underline{\Omega}}{\partial \underline{x}}^T \underline{\Lambda}_i \tag{V-9}$$

3. Calculate the parameter β_i using Equation II-30

$$\beta_i(\underline{\Lambda}_i) = \frac{\langle g_i(\underline{\Lambda}_i), g_i(\underline{\Lambda}_i) \rangle}{\langle g_{i-1}, g_{i-1} \rangle} \tag{V-10}$$

$$= \frac{1}{\langle g_{i-1}, g_{i-1} \rangle} \left[\langle \nabla f, \nabla f \rangle + 2 \langle \frac{\partial \underline{\Omega}}{\partial \underline{x}}^T \underline{\Lambda}_i, \nabla f \rangle + \langle \frac{\partial \underline{\Omega}}{\partial \underline{x}}^T \underline{\Lambda}_i, \frac{\partial \underline{\Omega}}{\partial \underline{x}}^T \underline{\Lambda}_i \rangle \right] \tag{V-11}$$

for $i = 0$, $\beta_0 = 0$

Note that $\beta_i(\underline{\Lambda}_i)$ is given by a quadratic function in the ρ parameters $\underline{\Lambda}$.

4. Calculate the direction of search

$$\underline{s}_i(\underline{\Lambda}_i) = -g_i(\underline{\Lambda}_i) + \beta_i(\underline{\Lambda}_i)\, \underline{s}_{i-1} \tag{V-12}$$

5. Solve the system

$$\frac{\partial \tilde{f}}{\partial \alpha_i}(\underline{x}_i + \alpha_i \underline{s}_i(\underline{\Lambda}_i)) = 0 \tag{V-13}$$

$$\underline{\Omega}(\underline{x}_i + \alpha_i \underline{s}_i(\underline{\Lambda}_i)) = \underline{0} \tag{V-14}$$

for α_i and $\underline{\Lambda}_i$.

The system given by V-13 and V-14 replaces the one-dimensional minimization. Solution of the system however, demands that the new vector $\underline{x}_{i+1}$ lie within the constraint space and also represent a stationary point of $\tilde{f}$ with respect to α_i.

6. Increase i and repeat from step 2 until the constrained minimum is reached.

Steps 2, 3 and 4 are done algebraically before numerical implementation of the process. The result is that after choosing an initial guess for $\underline{x}_0$, system V-13 is solved immediately. Once the α_i and $\underline{\Lambda}_i$ are determined, $\underline{s}_i(\underline{\Lambda}_i)$ is calculated and stored for use as the previous direction of search. The algebraic relations used in V-11, and V-12, and V-13 are problem dependent. Once they are determined by the user, they are invariant throughout the automated process. Solution of system V-13 will be called the 'inner loop' of the MCG method. In general, the inner loop is a $(p+1)$-dimensional system of nonlinear equations and requires the use of a numerical solution technique applicable to that class of problems. In the examples given in this chapter, the unconstrained SCG method has been used to minimize $\frac{\partial \tilde{f}}{\partial \alpha_i}^2 + \underline{\Omega}^2$ and thus to satisfy Equations V-13 and V-14. It should be stressed however, that the CG method need not be used in the inner loop to preserve the conjugate gradient formulae used in the derivation. In fact, in solving

the Equations V-13 and V-14, the user is free to exploit any peculiarities of the system that result from the specific problem being considered. For example, Pierson (59) shows by numerical investigation that shorter run times result from using Davidon's method instead of the CG method when the number of independent variables is 'small'. For most applications, the number of constraints p will be small. Once the values of α_i and Λ_i are obtained, the new search point $\underline{x}_{i+1}$ is obtained from the relation $\underline{x}_{i+1} = \underline{x}_i + \alpha_i \underline{s}_i$. The inner loop is then re-entered to determine the next stepsize α_i and the new values of Λ_i. It can be seen that computationally the outer loop consists only of computing and storing the direction of search and computing the search points. Thus the procedure results in a sequence of inner loop solutions of Equations V-13 and V-14. This is similar to the penalty function method except that no arbitrary constants must be chosen and the system to be solved is $(p+1)$-dimensional instead of n-dimensional.

In any conjugate gradient algorithm, setting the parameter β_i equal to zero for all i results in the optimum steepest descent method. Similarly, if β_i is zero for all i, the MCG method looks like a modified steepest descent method (MSD). McIntyre (53) discusses a very similar modification to steepest descent which permits constraints between the independent variables. The primary difference is that each search point in McIntyre's method satisfies a linearlized approximation of the constraint conditions whereas the MSD method presented here guarantees that each search point satisfies the original nonlinear constraint conditions exactly. Although linearization of the constraint permits the

derivation of an explicit formula for the Lagrange multipliers (see Equation 2.2.52, 53, p. 23); the expression is an approximation which is valid only when stepsizes are taken that are small enough to preserve the accuracy of the linearized constraint equations. This restriction on stepsize does not exist in the MSD method or the MCG method. What is required however is the existence and uniqueness of the solution to Equations V-13 and V-14. Since the form of the system is problem dependent, the satisfaction of these conditions will be assumed for all values of $\underline{x}$ in E_n.

McIntyre also gives a modification to the second order Newton's method that is applicable to the Lagrangian function. Linearization of the constraint equations is again necessary. However, it seems reasonable to expect that the MCG method given previously might provide convergence rates lying between those shown by the first and second order methods discussed by McIntyre.

When the MSD method ($\beta_i = 0$) is considered, the directions of search chosen are the negative gradients to the Lagrangian $\tilde{f}$ given by V-4. Since the method guarantees the satisfaction of the constraint at every step of the iteration, a decrease in the value of $\tilde{f}$ produced by a step in the direction of its negative gradient represents the same decrease in the function f regardless of the values of the Lagrange multipliers. Therefore it can be argued that the MSD method produces a monotonic decrease in the function f while causing the constraints $\underline{\Omega}(\underline{x})$ to be satisfied by each iterate.

The previous argument is not valid for the MCG algorithm in which

$\beta_i \neq 0,\ i > 0$ since the directions of search are no longer the negative gradients of $\tilde{f}$ and do involve the values of the Lagrange multipliers Λ. Unfortunately a proof of the conjugacy of the directions of search in the $(n-p)$-dimensional constraint space has not yet been produced. It is therefore theoretically undetermined whether the formal application of the CG formulae for determining the directions of search will produce more rapid convergence than the use of the MSD formulae using $\beta_i = 0 \ \forall i$.

Two finite-dimensional numerical solutions are presented to illustrate the MCG method and its convergence properties. The objective of the first is to find the coordinates on a parabola that minimize the distance to the point $(1, 0)$ in the x-y plane, i.e.

P-9. Minimize

$$f = (x-1)^2 + y^2 \tag{V-15}$$

subject to

$$\Omega = y^2 - 4x = 0 . \tag{V-16}$$

The Lagrangian is then

$$\tilde{f} = (x-1)^2 + y^2 - \Lambda(y^2 - 4x) . \tag{V-17}$$

Solution by the classical method of Lagrange multipliers gives

$$x^* = 0$$
$$y^* = 0$$
$$\Lambda^* = -\tfrac{1}{2}$$

The numerical solution given by the MCG method is presented in Table 14. The starting point was chosen as $x = -0.5$, $y = 1.0$ and does not satisfy the parabolic constraint. In the solution, each point after the initial point satisfied the constraints to within the limits set for

Table 14. Solution of problem P-9 using the MCG method

Iteration Number i	x_i	y_i	λ_i	α_i
0	-0.5	1.0		
1	.03124	0.3535	-.5756	0.7617
2	$-.1074\text{x}10^{-4}$	$-.4472\text{x}10^{-2}$	-.5129	0.8193
3	$.5664\text{x}10^{-5}$	$-.8091\text{x}10^{-3}$	-.5000	0.8191
4	$.1514\text{x}10^{-5}$	$-.2648\text{x}10^{-4}$	-.5000	0.8191
5	$-.4832\text{x}10^{-7}$	$-.4791\text{x}10^{-5}$	-.5000	0.8191

solution of the inner loop equations.

Another example with higher dimensionality is given here for the purpose of comparing the MCG and the MSD methods. The objective is to find the point on a cylindrical surface that lies the minimum distance from a fixed point $x = 1$, $y = 3$, $z = 2$, i.e.

P-10. Minimize

$$f = (x-1)^2 + (y-3)^2 + (z-2)^2 \tag{V-18}$$

subject to

$$\Omega = (x-1)^2 + y^2 - 4 = 0 \quad . \tag{V-19}$$

Solution by the classical method of Lagrange multipliers gives

$$x^* = 1$$
$$y^* = 2$$
$$z^* = 2$$
$$\lambda^* = \frac{1}{2}$$

Table 15 gives the MCG solution of the problem. The initial point in the search was $x = 2$, $y = 5$, $z = 7$ which is not on the constraint.

The same problem was solved with the MSD algorithm by letting $\beta_i = 0$. The comparison of the methods is given in Table 16. The quantity $\|e_i\|^2$ is defined by

$$\|e_i\|^2 = (x - x^*)^2 + (y - y^*)^2 + (z - z^*)^2 \qquad \text{(V-20)}$$

Although more extensive theoretical and computational investigation is necessary to establish the effect of using the CG equations formally to determine the directions of search in the manner indicated, the results of this finite-dimensional example suggest that the MCG method could provide more rapid convergence rates than the MSD method. It should be noted that if the MCG method produces a β_i that causes the function value to increase along any direction of search, the problem can be corrected by setting β_i equal to zero as is done in the standard conjugate gradient methods for unconstrained problems. This procedure makes the direction of search correspond to the negative gradient of the Lagrangian and thus a reduction in both $\tilde{f}$ and f is guaranteed. Limited computational experience with the finite-dimensional algorithm has not yet produced an 'uphill' direction of search using the MCG method. Problem P-10 was also solved by using penalty functions and a sequence of unconstrained subproblems. The results of both the MCG and the SUMT solution are given in Table 17.

Application of the Method to Constrained Optimal Control Problems

The extension of the modified conjugate gradient method to function space and thus its applicability to optimal control problems is now

Table 15. Solution of problem P-10 using the MCG method

Iteration Number i	x_i	y_i	z_i	λ_i
0	2.0	5.0	7.0	
1	0.8442	1.9939	2.3783	0.2504
2	1.0378	1.9996	2.0312	0.4925
3	0.9992	2.0000	2.0016	0.5001
4	1.0002	2.0000	2.0003	0.5000
5	1.0000	2.0000	2.0000	0.5000

Table 16. Comparison of the MCG and MSD methods on problem P-10

Iteration Number i	MCG $\|e_i\|^2$	MSD $\|e_i\|^2$
0		
1	5.54×10^{-1}	5.54×10^{-1}
2	4.45×10^{-4}	3.02×10^{-3}
3	1.89×10^{-5}	1.21×10^{-4}
4	8.13×10^{-7}	5.11×10^{-6}
5	1.36×10^{-9}	2.10×10^{-7}
6	8.29×10^{-11}	8.63×10^{-9}
7	3.59×10^{-12}	3.55×10^{-10}
8	7.68×10^{-15}	1.46×10^{-11}

Table 17. Comparison of MCG and SUMT solutions of problem P-10

Iteration Number	MCG $\|e_i\|^2$	Penalty function-SUMT Subproblem Number	$\|e_i\|^2$
0		0	
1	5.54×10^{-1}	1	3.15×10^{-5}
2	4.45×10^{-4}	2	1.93×10^{-8}
3	1.89×10^{-5}	3	3.36×10^{-8}
4	8.127×10^{-7}	4	9.70×10^{-10}
5	1.36×10^{-9}	5	2.49×10^{-12}
6	8.29×10^{-11}	6	1.46×10^{-12}
7	3.59×10^{-12}	7	1.07×10^{-13}
8	7.68×10^{-15}	8	1.46×10^{-13}
Execution Time	1.17 sec.		1.06 sec.

presented. The problem statement below is in Mayer form to simplify the algorithm developed subsequently. We wish to use the MCG method to minimize

$$J = \Phi(\underline{x}(t_f)) \tag{V-21}$$

subject to

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, t) \tag{V-22}$$

$$\underline{x}(t_0) = \underline{x}_0 \tag{V-23}$$

and

$$\underline{\Omega}(\underline{x}(t_f)) = 0 . \tag{V-24}$$

where $\underline{\Omega}$ is a p-vector of terminal state constraints relations and is fixed. Let

$$\tilde{J} = \tilde{\Phi}(\underline{x}(t_f)) = \Phi(\underline{x}(t_f)) + \underline{\lambda}^T \underline{\Omega} \tag{V-25}$$

where $\underline{\lambda}$ is a p-vector of constant Lagrange multipliers. Minimizing $\tilde{\Phi}(\underline{x}(t_f))$ while satisfying V-21-V-24 also minimizes Φ subject to the same constraints. As shown in Chapter II, the gradient $\underline{g}$ to the functional $\tilde{J}$ is given by

$$\underline{g} = \frac{\partial H}{\partial \underline{u}} = \underline{\lambda}^T(t) \frac{\partial \underline{f}}{\partial \underline{u}}(t) \tag{V-26}$$

where H is the Hamiltonian and $\underline{\lambda}(t)$ is the vector of adjoint or costate variables satisfying

$$\dot{\underline{\lambda}} = -\frac{\partial \underline{f}}{\partial \underline{x}}^T \underline{\lambda} \tag{V-27}$$

$$\underline{\lambda}(t_f) = \frac{\partial \tilde{\Phi}}{\partial \underline{x}}(t_f) = \frac{\partial \Phi}{\partial \underline{x}}(t_f) + \frac{\partial \underline{\Omega}}{\partial \underline{x}}(t_f)\underline{\lambda} . \tag{V-28}$$

The MCG algorithm for this control problem is the following:

1. Choose an initial control function $\underline{u}_o(t)$
2. Integrate the state system V-22, V-23 forward from t_o to t_f
3. Calculate the transition matrix $\Phi(t,\tau)$ for the linear homogeneous system V-27. Since the boundary conditions for V-27 are given at t_f, let $\tau = t_f - t$ and write V-27 as

$$\frac{d\underline{\lambda}_i}{d\tau} = \frac{\partial \underline{f}}{\partial \underline{x}}(\tau)^T \underline{\lambda}_i(\tau) \tag{V-29}$$

The transition matrix for the above is computed

from (see 3, pp. 127-128)

$$\dot{\Phi} = \frac{\partial \underline{f}}{\partial \underline{x}}^{T} \Phi , \qquad \Phi(\tau = 0) = I . \tag{V-30}$$

The solution of V-27 can then be written in terms of the unknown $\underline{\lambda}_i$ as

$$\underline{\lambda}_i(\tau; \underline{\lambda}_i) = \Phi(\tau, t_f) \left[\frac{\partial \Phi}{\partial \underline{x}} + \frac{\partial \underline{\Omega}}{\partial \underline{x}} \underline{\lambda}_i \right]_{\underline{x}(t_f)} . \tag{V-31}$$

4. Determine the gradient $\underline{g}_i(t; \underline{\lambda}_i)$ from

$$\underline{g}_i(t; \underline{\lambda}_i) = \Phi(t_f - t, t_f) \left[\frac{\partial \Phi}{\partial \underline{x}} + \frac{\partial \underline{\Omega}}{\partial \underline{x}} \underline{\lambda}_i \right]_{\underline{x}(t_f)} \frac{\partial \underline{f}}{\partial \underline{u}}(t) \tag{V-32}$$

5. Determine $\beta_i(\underline{\lambda}_i)$ from

$$\beta_i(\underline{\lambda}_i) = \frac{\langle \underline{g}_i(t; \underline{\lambda}_i), \underline{g}_i(t; \underline{\lambda}_i) \rangle}{\langle \underline{g}_{i-1}(t), \underline{g}_{i-1}(t) \rangle} \tag{V-33}$$

6. Determine the new control in terms of $\underline{\lambda}_i$ and α_i as

$$\underline{u}_{i+1}(t; \alpha_i, \underline{\lambda}_i) = \underline{u}_i + \alpha_i \left[-\underline{g}_i + \beta_i \underline{s}_{i-1} \right] \tag{V-34}$$

7. Solve the 'inner loop' equations

$$\underline{\Omega}(\underline{x}_{i+1}(t_f)) = 0 \tag{V-35}$$

$$\frac{\partial \tilde{J}}{\partial \alpha_i}(\underline{u}_{i+1}) = 0 \tag{V-36}$$

for α_i and $\underline{\lambda}_i$, where $\underline{x}_{i+1}$ results from integrating the state equations, V-22 and initial conditions V-23. When $\underline{\lambda}_i$ and α_i are known numerically, $\underline{u}_{i+1}(t)$ is a known function of time.

8. Repeat from step 2 until the constrained minimum is reached.

Steps 3, 4, 5 and 6 are, to a large extent, executed analytically as the problem is cast into the proper form for the algorithm. For

example, an analytic expression for $\underline{\lambda}(t, \underline{\Lambda})$ can be written in terms of the unknown $\underline{\Lambda}$ and the elements of Φ. From this, the analytic expression for $\underline{g}(t, \underline{\Lambda})$ is determined for the particular problem being solved. Similarly the expression for $\beta(\underline{\Lambda})$ can be written as a quadratic expression in the elements of $\underline{\Lambda}$. The coefficients of the expression are quadrature integrals involving known functions of time. Note that the denominator of V-33 involves no unknown quantities and is in fact already calculated from the previous β calculation. Finally, an expression for $\underline{u}_{i+1}$ in terms of the elements of Φ, the quadrature integrals from the β expression, and the unknowns α_i and $\underline{\Lambda}_i$ is derived. After the integration of V-30 to determine the transition matrix and the evaluation of the quadrature terms arising from V-33, the new control becomes a function of α_i and $\underline{\Lambda}_i$ only. The inner loop solution must then be accomplished.

Many methods of solving the inner loop Equations V-35 and V-36 require minimizing an expression of the form

$$\underline{\Omega}(\underline{x}_{i+1})^T \underline{\Omega}(\underline{x}_{i+1}) + K\left(\frac{\partial \tilde{J}}{\partial \alpha_i}(\underline{u}_{i+1})\right)^2$$

where K is some positive constant. If the method chosen requires first derivative information, then expressions for the quantities

$$\frac{\partial^2 \tilde{J}}{\partial \alpha_i^2}(\underline{u}_{i+1}) \quad \text{and} \quad \frac{\partial^2 \tilde{J}}{\partial \alpha_i \partial \Lambda_{ij}}(\underline{u}_{i+1}) \quad j = 1, 2, \cdots p$$

as well as $\frac{\partial \tilde{J}}{\partial \alpha_i}(\underline{u}_{i+1})$ must be derived. Here Λ_{ij} represents the jth component of the ith iterate of $\underline{\Lambda}$. To obtain these quantities, the following expressions can be derived:

$$\frac{\partial \tilde{J}}{\partial \alpha_i}(\underline{u}_{i+1}) = \left[\frac{\partial \tilde{\Phi}}{\partial \underline{x}_{i+1}} \frac{\partial \underline{x}_{i+1}}{\partial \alpha_i}\right]_{t_f} \tag{V-37}$$

$$\frac{\partial^2 \tilde{J}}{\partial \alpha_i^2} = \left[\frac{\partial^2 \tilde{\Phi}}{\partial \underline{x}_{i+1}} \frac{\partial \underline{x}_{i+1}}{\partial \alpha_i}^2 + \frac{\partial \tilde{\Phi}}{\partial \underline{x}_{i+1}} \frac{\partial^2 x_{i+1}}{\partial \alpha_i^2} \right]_{t_f} \tag{V-38}$$

$$\frac{\partial^2 \tilde{J}}{\partial \alpha_i \partial \lambda_{ij}} = \left[\frac{\partial \Phi}{\partial \underline{x}_{i+1}} \frac{\partial^2 \underline{x}_{i+1}}{\partial \alpha_i \partial \lambda_{ij}} + \frac{\partial^2 \tilde{\Phi}}{\partial \underline{x}_{i+1}^2} \frac{\partial \underline{x}_{i+1}}{\partial \lambda_{ij}} \frac{\partial x_{i+1}}{\partial \alpha_i} \right]_{t_f} \quad j = 1, 2, \cdots p \tag{V-39}$$

However $\frac{\partial \underline{x}_{i+1}}{\partial \alpha_i}$, $\frac{\partial^2 \underline{x}_{i+1}}{\partial \alpha_i^2}$ and $\frac{\partial^2 \underline{x}_{i+1}}{\partial \alpha_i \partial \lambda_{ij}}$ can be obtained by using first order perturbation analyses on Equations V-22. For example,

$$\frac{\partial}{\partial \alpha_i} \dot{\underline{x}}_{i+1} = \frac{\partial \underline{f}}{\partial \alpha_i} = \frac{\partial \underline{f}}{\partial \underline{x}_{i+1}} \frac{\partial \underline{x}_{i+1}}{\partial \alpha_i} + \frac{\partial \underline{f}}{\partial \underline{u}_{i+1}} \frac{\partial \underline{u}_{i+1}}{\partial \alpha_i} \tag{V-40}$$

but

$$\frac{\partial \underline{u}_{i+1}}{\partial \alpha_i} = \underline{s}_i(t; \lambda_i) \tag{V-41}$$

$$= -\underline{g}_i + \beta_i \underline{s}_{i-1} \quad . \tag{V-42}$$

Interchanging the order of differentiation in V-40 gives

$$\frac{d}{dt} \frac{\partial \underline{x}_{i+1}}{\partial \alpha_i} = \frac{\partial \underline{f}}{\partial \underline{x}_{i+1}} \frac{\partial \underline{x}_{i+1}}{\partial \alpha_i} + \frac{\partial \underline{f}}{\partial \underline{u}_{i+1}} \underline{s}_i$$

Since the initial conditions on χ are fixed,

$$\frac{\partial \underline{x}_{i+1}}{\partial \alpha_i}(t_o) = 0 \tag{V-43}$$

represent the proper initial conditions for the system. Similar analyses lead to the following equations in which iteration subscripts have been dropped to simplify notation:

$$\frac{d}{dt}\left(\frac{\partial \underline{x}}{\partial \lambda_j} \right) = \frac{\partial \underline{f}}{\partial \underline{x}} \frac{\partial \underline{x}}{\partial \lambda_j} + \frac{\partial \underline{f}}{\partial \underline{u}} \frac{\partial \underline{u}}{\partial \lambda_j} \qquad j = 1, 2, \cdots p \tag{V-44}$$

$$\frac{d}{dt}\left(\frac{\partial^2 \underline{x}}{\partial \alpha_i \partial \lambda_j}\right) = \frac{\partial^2 \underline{f}}{\partial \underline{x}^2}\frac{\partial \underline{x}}{\partial \alpha}\frac{\partial \underline{x}}{\partial \lambda_j} + \frac{\partial^2 \underline{f}}{\partial \underline{x}\partial \underline{u}}\frac{\partial \underline{x}}{\partial \alpha}\frac{\partial \underline{u}}{\partial \lambda_j} + \frac{\partial \underline{f}}{\partial \underline{x}}\frac{\partial^2 \underline{x}}{\partial \alpha \partial \lambda_j}$$

$$+ \frac{\partial^2 \underline{f}}{\partial \underline{u}^2}\frac{\partial \underline{u}}{\partial \alpha}\frac{\partial \underline{u}}{\partial \lambda_j} + \frac{\partial^2 \underline{f}}{\partial \underline{u}\partial \underline{x}}\frac{\partial \underline{u}}{\partial \alpha}\frac{\partial \underline{x}}{\partial \lambda_j}$$

$$+ \frac{\partial \underline{f}}{\partial \underline{u}}\frac{\partial^2 \underline{u}}{\partial \alpha \partial \lambda_j} \qquad \text{(V-45)}$$

$$\frac{d}{dt}\left(\frac{\partial^2 \underline{x}}{\partial \alpha^2}\right) = \frac{\partial^2 \underline{f}}{\partial \underline{x}^2}\left(\frac{\partial \underline{x}}{\partial \alpha}\right)^2 + \frac{\partial \underline{f}}{\partial \underline{x}}\frac{\partial^2 \underline{x}}{\partial \alpha^2} + \frac{\partial \underline{f}}{\partial \underline{u}^2}\underline{s}^2 \qquad \text{(V-46)}$$

The initial conditions are all zero for these perturbation equations.

Birta and Trushel (8) comment that the 'analytic gradient' approach used above may be less satisfactory than a determination of the needed partial derivatives by a finite difference method. He states that the perturbation equations can, under certain circumstances, become 'ill-conditioned' making accurate numerical integration of them difficult. Insufficient computational experience was gained in this study to allow comment on that point. However, the determination of the inner loop gradient components involves considerable computing effort whether the needed partials derivatives are calculated using the perturbation equations or by repeated integration of the states equations V-22. Since the inner loop is traversed so frequently in the computation procedure, improved computing efficiency demands great care in choosing the method of solution of Equations V-35 and V-36. For example, numerical minimization methods which require only function evaluations might prove to be more efficient than gradient methods. However, the purpose here is not to suggest a method to solve the inner loop, but to present the MCG logic which does

not depend upon the inner loop algorithm.

Step 3 of the MCG algorithm for the optimal control problem requires the calculation of an $n \times n$ dimensional transition matrix. This is not unlike the projection method of Sinnott and Luenberger (69). The projection method requires the solution of the matrix equation $\dot{Q} = -\frac{\partial f}{\partial \underline{x}}^T Q$ where Q is an $n \times p$ matrix defined by $Q = \hat{\Phi}(t_f, t)^T A^T$. $\hat{\Phi}$ is the transition matrix for the linearized system equations and A^T is an $n \times p$ matrix of constants. In addition another $n \times n$ matrix differential equation must be solved and a $p \times p$ matrix inverted each time a projection or correction is made from a new point in the control space. If several small corrections are made to improve the performance of the method, these calculations must be repeated several times for each forward step. The same comments can be made of course regarding the projection method when used with steepest descent. By comparison then, the calculation of a transition-matrix once for each step of the MCG method does not represent excessive computational effort.

The adjoint system given by Equations V-27 and V-28 is, in general, a linear time-varying system since the matrix $\frac{\partial f}{\partial \underline{x}}$ is evaluated using the current control. The appearance of the transition matrix for that system does not represent an approximation. This is in contrast to the projection methods that make use of the transition matrix of the approximate linearized state equations.

To demonstrate the application of the method to an optimal control problem, the MCG solution of the unit mass problem P-7 stated in Chapter IV is presented here. Appendix B contains a summary of the equations used to implement the MCG algorithm for this problem. The simplicity of the

problem made it unnecessary to perform a numerical computation of the transition matrix for the adjoint system. The results of the MCG solution are given in Table 18. The problem admits an analytic solution which is also presented in Table 18 for comparison purposes. Convergence to the optimal control is virtually complete after one iteration. However the value of the Lagrange multiplier is not accurate until after a second iteration is completed. The first solution of the inner loop was accomplished in three steps from an initial guess of $\alpha = 1.0$, $\lambda = 0$. The second inner loop solution used only one step and was started from the values of α and λ obtained from the first solution.

The first stepsize chosen by the MCG method was 0.5000 and the stepsize used by the projection method in solving the problem in one step was 0.4988. Since both methods give excellent representations of the optimal control after one step, the first direction of search chosen by the MCG method is very nearly the same as that chosen by the projection method. This is an expected result for a problem with linear dynamics and linear constraints. If however, the MCG and projection methods were applied to a nonlinear problem, the directions of search would not be explected to be the same. Geometrically, the projection method chooses a direction of search that lies in linear surface which is tangent to the constraint at the current search point in the control space. On a nonlinear problem, a step in this plane produces constraint violation. In contrast, the MCG method produces only those controls that satisfy the constraints. Its directions of search therefore would not lie in linear surfaces that are tangent to the constraint in the control space.

Table 18. Solution of problem P-7 using the MCG method

Time (sec.)	u_0	u_1	u_2	u^*
0	50	18.6115	18.6114	18.6112
0.1	45	17.5030	17.5028	17.5027
0.2	40	16.2778	16.2777	16.2776
0.3	35	14.9238	14.9237	14.9236
0.4	30	13.4274	13.4273	13.4272
0.5	25	11.7737	11.7736	11.7735
0.6	20	9.9460	9.9459	9.9458
0.7	15	7.9260	7.9260	7.9260
0.8	10	5.6937	5.6937	5.6936
0.9	5	3.2266	3.2266	3.2266
1.0	0	0.50000	0.50000	0.50000
		$\chi_1(1) = 5.0000$	$\chi_1(1) = 5.0000$	$\chi_1^*(1) = 5.0000$
		$\alpha = 0.50000$	$\alpha = .50000$	
		$\mathcal{L} = -0.00374$	$\mathcal{L} = -58.3034$	$\mathcal{L}^* = -58.3031$

CHAPTER VI. COMPARATIVE DISCUSSION AND RECOMMENDATIONS FOR ADDITIONAL INVESTIGATION

The purpose of this study has been to identify and investigate the theoretical and computational characteristics of various adaptations of the conjugate gradient method to optimal control problems with terminal state constraints. Three such adaptations have been discussed. The penalty function technique and the projection technique have been suggested by others and refined or extended here, whereas the modified conjugate gradient method is original to this study.

Comparative results that are not presented elsewhere in the dissertation are given here. The Van der Pol problem P-6 with linear constraints has been solved in this thesis using both the penalty function-SUMT approach and the projection method. The results are given in Chapters III and IV. The execution times required for the solutions were 30.1 seconds and 37.8 seconds for the penalty function and the projection methods, respectively. The penalty function run was made after several previous solutions to the problem had given some insight into the choice of the penalty constants. Other choices could produce significantly longer or perhaps even shorter execution times.

The linear unit mass problem P-7 given in Chapters IV and V offers another comparative result. The projection method solved the problem in one iteration and 2.0 seconds of execution time. The MCG method solved the problem to the same degree of accuracy in one iteration and 13.9 seconds of execution time. This comparison is not particularly informative however, since the example problem is in different respects an

advantageous choice for both methods. A more accurate comparison of these two methods must await the solution of a more challenging problem using the MCG method and the further refinement of that technique.

Other comparisons of computational techniques and algorithm variations have already been reported in the chapters concerning the particular methods. These include a comparison of the PCG and the SCG methods in Chapter II, a comparison of the Lagrange and Bolza formulation of the same problem in Chapter II, a comparison of the SUMT and fixed penalty constant approaches in Chapter III, a comparison of the MSD and MCG method for finite-dimensional problems in Chapter V, and a comparison of the MCG and the SUMT methods in Chapter V.

With regard to the ease of implementation, the penalty function approach has the advantage of programming simplicity. The method works well for problems with a small number of constraints and in those cases where acceptable penalty constants can be chosen from a wide range of positive constants. The projection method is theoretically more sophisticated and requires more programming effort. Because of the conflict between the large stepsizes chosen by the conjugate gradient method and the small stepsizes required by the linear theory used in the projection equations, a reasonably sophisticated stepsize adjustment policy must be implemented. Limited computational experience suggests that relaxation of the linearity requirements in favor of larger stepsizes results in more rapid convergence. The allowable constraint violation for each step of the projection method must be set arbitrarily by the user. The algorithm works most efficiently when the acceptable constraint satisfaction is achieved on approximately the same iteration

as the minimum of the functional. It is usually impossible to preset the allowable constraint violations properly. For this reason, the projection method like the penalty function method often requires several trial solutions before an efficient solution is obtained. The MCG method is the most difficult of the three methods to implement. It requires the user to derive several algebraic relationships that are necessary for the inner loop structure. The effort required depends to large extent upon the method chosen to solve the inner loop equations. However, once the implentation is complete, these are no arbitrary parameters that must be adjusted on the basis of experience. Each control iterate satisfies the constraints, and each stepsize chosen is used without adjustment. The method does require an initial guess of the values of the first stepsize and the first approximation to the Lagrange multipliers if a direct method of minimization is used in the inner loop.

Additional research on the topic of this dissertation could include a more detailed and conclusive comparison of the operating efficiencies of the three methods investigated here. Such a study requires the existence of software that has been developed to a reasonably efficient state. This software has evolved during this study and is now available.

Apart from comparative studies, additional research on the adaptation of the conjugate gradient methods to constrained problems could pursue one of several alternatives. The MCG method is in an embryonic state of development. A theoretical demonstration of the conjugacy of the direction of search within the constraint space would be a major contribution. Numerical solutions of more difficult problems are needed

to evaluate the method's performance. Techniques for solving the inner loop nonlinear algebraic system should be explored since the efficiency of the method relies heavily upon the efficiency of the inner loop calculations.

Applications of CG methods to control problems having variable endtimes, multiple control variables, control variable inequality constraints, or inequality constraints involving the state variables at times other than the final time have been limited. The superiority of the CG methods to the ordinary gradient methods on problems to which both apply is sufficient to encourage further attempts to adapt the conjugate gradient technique to a class of problems with greater generality than has been considered here.

LITERATURE CITED

1. Antosiewicz, H. A. and Rheinboldt, W. C. Conjugate-direction methods and the method of steepest descent. In Todd, John, ed. Numerical analysis and functional analysis. Pp. 501-517. New York, New York, McGraw-Hill Book Co., Inc. 1962.

2. Arrow, K. J. and Solow, R. M. Gradient methods for constrained maxima with weakened assumptions. In Arrow, K. J., Hurwicz, L., and Uzawa, H. Studies in Linear and Nonlinear Programming. Pp. 166-176. Stanford, California, Stanford University Press. 1958.

3. Athans, M. and Falb, P. L. Optimal control: an introduction to the theory and its applications. New York, N.Y., The McGraw-Hill Book Co. 1966.

4. Beckman, F. S. The solution of linear equations by the conjugate gradient method. In Ralston, A. and Wilf, H. S., eds. Mathematical Methods for Digital Computers. Vol. 1. Pp. 62-72. New York, New York, John Wiley and Sons, Inc. 1960.

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

6. Bellman, R. E. and Dreyfus, S. E. Applied dynamic programming. Princeton, New Jersey, Princeton University Press. 1962.

7. Berkovitz, L. D. Variational methods in problems of control and programming. Journal of Mathematical Analysis and Applications 8: 145-169. 1961.

8. Birta, L. G. and Trushel, P. J. The TEF/Davidon-Fletcher-Powell method in the computation of optimal controls. National Research Council Laboratory, Ottawa, Canada Memorandum AC-97. 1969.

9. Bliss, G. A. Lectures on the calculus of variations. Chicago, Illinois, University of Chicago Press. 1946.

10. Blum, E. K. Minimization of functionals with equality constraints. Society of Industrial Applied Mathematics Journal on Control Series A, 3: 299-316. 1965.

11. Breakwell, J. V., Speyer, J. L., and Bryson, A. E., Jr. Optimization and control of nonlinear systems using the second variation. Society of Industrial Applied Mathematics Journal on Control Series A, 1: 193-223. 1963.

12. Bryson, A. E., Jr. and Denham, W. F. Optimum programming problems with inequality constraints. II. Solutions by Steepest Descent. American Institute of Aeronautics and Astronautics Journal 2: 25-34. 1964.

13. Bryson, A. E., Jr., Denham, W. F., and Dreyfus, S. E. Optimal programming problems with inequality constraints. I. Necessary Conditions for Extremal Solutions. American Institute of Aeronautics and Astronautics Journal 1: 2544-2550. 1963.

14. Curry, H. B. The method of steepest descent for non-linear minimization problems. Quarterly of Applied Mathematics 2: 258-262. 1944.

15. Daniel, James W. The conjugate gradient method for linear and nonlinear operator equations. Society of Industrial Applied Mathematics Journal on Numerical Analysis 4: 10-26. 1967.

16. Daniel, James W. Convergence of the conjugate gradient method with computationally convenient modifications. Numerische Mathematik 10: 125-131. 1967.

17. Davidon, W. C. Variable metric method for minimization. Argonne National Laboratory Report ANL-5990 (Rev. 2) (Argonne National Lab., Argonne, Ill.) Revised February 1966.

18. Denham, W. F. and Bryson, A. E., Jr. Optimal programming problems with inequality constraints. II. Solution by Steepest Ascent. American Institute of Aeronautics and Astronautics Journal 2: 25-34. 1964.

19. Dunford, N. and Schwartz, J. T. Linear operators part I: general theory. New York, N.Y., Interscience Publishers, Inc. 1958.

20. Engeli, M., Ginsburg, T. H., Rutishauser, H., Stiefel, E. Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems. Mitteilungen aus dem Institut für angewandte Mathematik, Birkhäuser Verlag, Basel. Nr. 8. 1959.

21. Fiacco, A. V. and McCormick, G. P. Computational algorithm for the sequential unconstrained minimization technique for nonlinear programming. Management Science 10: 601-617. 1964.

22. Fiacco, A. V. and McCormick, G. P. Extensions of SUMT for nonlinear programming: equality constraints and extrapolation. Management Science 12: 816-828. 1966.

23. Fiacco, A. V. and McCormick, G. P. The sequential unconstrained minimization technique for nonlinear programming, a primal-dual method. Management Science 10: 360-366. 1964.

24. Flanigan, P. D., Vitale, P. A. and Mendlesohn, J. A numerical investigation of several one-dimensional search procedures in nonlinear regression problems. Technometrics 2: 265-284. 1969.

25. Fletcher, R. and Powell, M. J. D. A rapidly convergent descent method for minimization. The Computer Journal 6: 163-168. 1963.

26. Fletcher, R. and Reeves, C. M. Function minimization by conjugate gradients. The Computer Journal 7: 149-154. 1964.

27. Forsythe, G. E. Computing constrained minimia with Lagrange multipliers. Journal of the Society of Industrial Applied Mathematics 3: 173-178. 1955.

28. Goldfarb, D. Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints. SIAM J. Applied Math. Submitted for publication 1968.

29. Goldfarb, D. and Lapidus, L. Conjugate gradient method for non-linear programming problems with linear constraints. Industrial and Engineering Chemistry Fundamentals 7: 142-151. 1968.

30. Hayes, R. M. Iterative methods of solving linear problems in Hilbert Space. National Bureau of Standards Applied Mathematics Series 39: 71-104. 1954.

31. Hestenes, M. R. The conjugate-gradient method for solving linear systems. Proceedings of the Symposia in Applied Mathematics 6: 83-102. New York, N.Y., McGraw-Hill Book Co., Inc. 1956.

32. Hestenes, M. R. and Stiefel, E. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards 49: 409-436. 1952.

33. Horwitz, L. B. An investigation of iteration techniques in function spaces and applications to optimal control problems. Unpublished Ph.D. thesis. New York, N.Y., New York University School of Engineering and Science. 1968.

34. Horwitz, L. B. and Sarachik, P. E. A computational technique for calculating the optimal control signal for a specific class of problems. Second Asilomar Conference on Circuits and Systems preprint, Pacific Grove, California, November 1968. Bell Telephone Laboratories, Inc., Whippany, New Jersey. 1969.

35. Horwitz, L. B. and Sarachik, P. E. Davidon's method in Hilbert space. Society of Industrial Applied Mathematics Journal on Applied Mathematics 16: 676-695. 1968.

36. Horwitz, L. B. and Sarachik, P. E. A survey of two recent iterative techniques for computing optimal control signals. Joint Automatic Control Conference preprint. 1969.

37. Hsieh, H. C. Synthesis of adaptive control systems by function space methods. In Leondes, C. T., ed. Advances in Control. Vol. 2. Pp. 118-206. New York, New York, Academic Press. 1965.

38. Jacobson, D. H. New second-order and first-order algorithms for determining optimal control: a differential dynamic programming approach. Journal of Optimization Theory and Applications 2, No. 6: 411-440. November 1968.

39. Jurovics, S. A. and McIntyre, J. E. Adjoint method and its application to optimum trajectory analyses. American Rocket Society Journal 32: 1354-1358. 1962.

40. Kawamura, K., and Volz, R. A. On the convergence of the conjugate gradient method in Hilbert space. Institute of Electrical and Electronics Engineers Transactions on Automatic Control AC-14(3): 296-297. 1969.

41. Kelley, H. J. Gradient theory of optimal flight paths. American Rocket Society Journal 30: 947-954. 1960.

42. Kelley, H. J. Guidance theory and extremal fields. Institute of Radio Engineers Transactions on Automatic Control. 1962: 75-82. October 1962.

43. Kelley, H. J. Method of gradients. In Leitmann, G., ed. Optimization Techniques. Pp. 206-254. New York, New York, Academic Press Inc. 1962.

44. Kelley, H. J., Kopp, R. E. and Moyer, H. G. A trajectory optimization technique based upon the theory of the second variation. In Szebehely, V. G., ed. Progress in Astronautics and Astrodynamics. Pp. 559-582. New York, New York, Academic Press Inc. 1964.

45. Kelley, H. J. and Myers, G. E. Conjugate direction methods for parameter optimization. Presented at the 18th Congress of the International Astronautical Federation, Belgrade, Yugoslavia. September 24-30, 1967. Analytical Mechanics Associates, Inc., Westbury, New York. 1967.

46. Kelley, H. J. and Speyer, J. L. Accelerated gradient projection. Colloquium on Optimization, preprint, Nice France, July 1969. Analytical Mechanics Associates, Inc., Westbury, New York. 1969.

47. Kolmogorov, A. N., and Fomin, S. V. Elements of the theory of functions and functional analysis. Metric and Normed Spaces. Translated from the Russian first edition. Vol. I. Rochester, N.Y., Graylock Press. 1957.

48. Kuhn, H. W. and Tucker, A. W. Proceedings of the second Berkeley symposium on mathematical statistics and probability. Pp. 481-493. Berkeley, California, University of California Press. 1951.

49. Lasdon, L. S. and Waren, A. D. An interior penalty method for inequality constrained optimal control problems. Institute of Electrical and Electronics Engineers Transactions on Automatic Control AC-12: 388-395. 1967.

50. Lasdon, L. S., Mitter, S., and Waren, A. D. The method of conjugate gradients for optimal control problems. Institute of Electrical and Electronics Engineers Transactions on Automatic Control AC-12: 132-138. 1967.

51. Mayne, D. Q. A second-order gradient method of optimizing non-linear discrete time systems. International Journal of Control 3: 85-95. 1966.

52. McGill, R. and Kenneth, P. Solution of variational problems by means of a generalized Newton-Raphson operator. American Institute of Aeronautics and Astronautics Journal 2: 1761-1766. 1964.

53. McIntyre, J. E. Guidance, flight mechanics and trajectory optimization. XIII. Numerical optimization methods. National Aeronautics and Space Administration, NASA CR-1012 (George C. Marshall Space Flight Center, Huntsville, Alabama). 1968.

54. McReynolds, S. R. The successive sweep method and dynamic programming. Journal of Mathematical Analysis and Applications 19: 565-598. 1967.

55. Mehra, R. K. Computation of the inverse Hessian matrix using conjugate gradient methods. Institute of Electrical and Electronics Engineers Proceedings 57: 225-226. 1969.

56. Mehra, R. K. and Bryson, A. E., Jr. Conjugate gradient methods with an application to V/STOL flight-path optimization. Journal of Aircraft 6: 123-128. 1969.

57. Myers, G. E. Properties of the conjugate gradient and Davidon methods. Journal of Optimization Theory and Applications 2: 209-219. 1968.

58. Pagurek, B. and Woodside, C. M. The conjugate gradient method for optimal control problems with bounded control variables. Automatica 4: 337-349. 1968.

59. Pierson, B. L. A discrete-variable approximation to optimal flight paths. Astronautica Acta 14: 157-169. 1967.

60. Pierson, B. L. Numerical solution of nonlinear optimal control problems by discrete variable approximation. Society of Industrial Applied Mathematics Fall Meeting, Philadelphia, Pennsylvania, October 1968. Department of Aerospace Engineering, Iowa State University, Ames, Iowa. 1968.

61. Pomentale, T. A new method for solving conditioned maxima problems. Journal of Mathematical Analysis and Applications 10: 216-220. 1965.

62. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V. and Mishchenko, E. F. The Mathematical Theory of Optimal Processes. New York, N.Y., John Wiley and Sons, Inc. 1962.

63. Porter, W. A. Modern foundations of systems engineering. New York, New York, The Macmillan Company. 1966.

64. Rosen, J. B. The gradient projection method for nonlinear programming. Part I. Linear constraints. Journal of the Society of Industrial Applied Mathematics 8: 181-217. 1960.

65. Rosen, J. B. The gradient projection method for nonlinear programming. Part II. Nonlinear constraints. Journal of the Society of Industrial Applied Mathematics 8: 514-532. 1961.

66. Rosen, J. B. Iterative solution of nonlinear optimal control problems. Society of Industrial Applied Mathematics Journal on Control 4: 223-244. 1966.

67. Russel, D. Penalty functions and bounded phase coordinate control. Society of Industrial Applied Mathematics Journal on Control. Series A, 2: 409-422. 1964.

68. Schley, C. H., Jr. and Lee, I. Optimal control computation by the Newton-Raphson method and the Riccati transformation. Institute of Electrical and Electronics Engineers Transactions on Automatic Control AC-12: 139-144. April 1967.

69. Sinnott, J. F., Jr. and Luenberger, D. G. Solution of optimal control problems by the method of conjugate gradients. Joint Automatic Control Conference Proceedings 1967: 566-574. 1967.

70. Spang, H. A., III. A review of minimization techniques for nonlinear functions. Society of Industrial Applied Mathematics Review 4: 343-365. 1962.

71. Thrasher, G. J. Optimal programming for aircraft trajectories utilizing a new strategy for handling state inequality constraints. Paper # 69-812 AIAA Aircraft Design and Operations Meeting, July 1969. New York, New York. 1969.

72. Tripathi, S. S. and Narendra, K. S. Conjugate direction methods for nonlinear optimization problems. Proceedings of the National Electronics Conference 1968: 125-129. December 1968.

73. Tripathi, S. S. and Narendra, K. S. Optimization using conjugate gradient methods. Yale University Dept. of Engineering and Applied Science, Dunham Laboratory Technical Report CT-27. May 1969.

74. Varaiya, P. P. Nonlinear programming and optimal control. University of California Electronics Research Laboratory. ERL Technical Memorandum M-129. 1965.

ACKNOWLEDGMENTS

The author wishes to express his gratitude to Dr. B. L. Pierson for his guidance and suggestions concerning the research reported in this dissertation. In addition, he wishes to thank his committee co-chairmen Dr. E. W. Anderson and Dr. C. J. Triska for their support and encouragement throughout the degree program.

The author is indebted to his wife Lynda for her generous assistance in preparing the many drafts of the manuscript and to Mrs. Marjorie Wibholm for typing the final manuscript.

This work was supported by the Office of Naval Research under Project THEMIS, Contract N1004-68A-0162, Iowa State University, Ames, Iowa.

APPENDIX A. DERIVATION OF THE AUXILIARY EQUATIONS FOR THE PCG METHOD IN FUNCTION SPACE

In order to apply the pure conjugate gradient method to optimal control problems a means of computing the quantity $\hat{N}\, \underline{s}_{i-1}(t)$ must be derived. $\hat{N}\, \underline{s}_{i-1}(t)$ is the analog of the expression $N\, \underline{s}_{i-1}$ which appears in Equation II-4 of the finite-dimensional algorithm. N is the Hessian matrix of the objective function for the finite-dimensional problem. $\hat{N}[\cdot]$ is the analogous second-order operator in a Taylor's series expansion of the cost functional J for the function space problem. After $\hat{N}[\cdot]$ is identified from the second-order expansion of J, certain auxiliary variables can be defined to aid in the computation of $\hat{N}\, \underline{s}_{i-1}(t)$.

Consider the Bolza cost functional

$$J = \phi(\underline{x}(t_f)) + \int_{t_0}^{t_f} F(\underline{x}, \underline{u}, t)\, dt \tag{A-1}$$

where $\underline{x}$ is n-dimensional and $\underline{u}$ is m-dimensional. Expanding $J(\underline{u})$ about a nominal control $\hat{u}(t)$ gives

$$\begin{aligned} \Delta J = J(\underline{u}) - J(\hat{\underline{u}}) \cong\ & \frac{\partial \phi}{\partial \underline{x}_f}^T \delta \underline{x}(t_f) + \frac{1}{2}\, \delta \underline{x}(t_f)^T \frac{\partial^2 \phi}{\partial \underline{x}_f^2} \delta \underline{x}(t_f) \\ & + \left\langle \frac{\partial F}{\partial \underline{x}}, \delta \underline{x} \right\rangle + \frac{1}{2} \left\langle \delta \underline{x}, \frac{\partial^2 F}{\partial \underline{x}^2} \delta \underline{x} \right\rangle + \left\langle \frac{\partial F}{\partial \underline{u}}, \delta \underline{u} \right\rangle \\ & + \frac{1}{2} \left\langle \delta \underline{u}, \frac{\partial^2 F}{\partial \underline{u}^2} \delta \underline{u} \right\rangle + \frac{1}{2} \left\langle \delta \underline{x}, \frac{\partial^2 F}{\partial \underline{x} \partial \underline{u}} \delta \underline{u} \right\rangle \\ & + \frac{1}{2} \left\langle \delta \underline{u}, \frac{\partial^2 F}{\partial \underline{u} \partial \underline{x}} \delta \underline{x} \right\rangle \end{aligned} \tag{A-2}$$

where all derivatives are evaluated along trajectories resulting from $\hat{\underline{u}}(t)$ and where $\hat{\underline{x}}_f = \hat{\underline{x}}(t_f)$. Linearization of the state equations $\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, t)$ results in

$$\delta \underline{x}(t_f) = \int_{t_o}^{t_f} \Phi(t_f,\tau) \frac{\partial \underline{f}}{\partial \underline{u}}(\tau)\, \delta \underline{u}(\tau)\, d\tau = T[\delta \underline{u}], \tag{A-3}$$

and

$$\delta \underline{x}(t) = \int_{t_o}^{t} \Phi(t,\tau) \frac{\partial \underline{f}}{\partial \underline{u}}(\tau)\, \delta \underline{u}(\tau)\, d\tau = S[\delta \underline{u}], \tag{A-4}$$

where $\Phi(t,\tau)$ represents the transition matrix for the linearized state equations. Equation A-2 may be rewritten as

$$\begin{aligned}
\Delta J \cong\ & \left\langle \frac{\partial \Phi}{\partial \underline{x}_f}, T[\delta \underline{u}] \right\rangle + \frac{1}{2} \left\langle T[\delta \underline{u}], \frac{\partial^2 \Phi}{\partial \underline{x}_f^2} T[\delta \underline{u}] \right\rangle \\
& + \left\langle \frac{\partial F}{\partial \underline{x}}, S[\delta \underline{u}] \right\rangle + \left\langle \frac{\partial F}{\partial \underline{u}}, \delta \underline{u} \right\rangle + \frac{1}{2} \left\langle S[\delta \underline{u}], \frac{\partial^2 F}{\partial \underline{x}^2} S[\delta \underline{u}] \right\rangle \\
& + \frac{1}{2} \left\langle \delta \underline{u}, \frac{\partial^2 F}{\partial \underline{u}^2} \delta \underline{u} \right\rangle + \frac{1}{2} \left\langle S[\delta \underline{u}], \frac{\partial^2 F}{\partial \underline{x} \partial \underline{u}} \delta \underline{u} \right\rangle \\
& + \frac{1}{2} \left\langle \delta \underline{u}, \frac{\partial^2 F}{\partial \underline{u} \partial \underline{x}} S[\delta \underline{u}] \right\rangle
\end{aligned} \tag{A-5}$$

$$\begin{aligned}
= & \left\langle \frac{\partial \Phi}{\partial \underline{x}_f}, T[\delta \underline{u}] \right\rangle + \left\langle \frac{\partial F}{\partial \underline{u}}, \delta \underline{u} \right\rangle + \left\langle \frac{\partial F}{\partial \underline{x}}, S[\delta \underline{u}] \right\rangle \\
& + \frac{1}{2} \left\langle \delta \underline{u}, T^* \frac{\partial^2 \Phi}{\partial \underline{x}^2} T[\delta u] \right\rangle + \frac{1}{2} \left\langle \delta \underline{u}, S^* \frac{\partial^2 F}{\partial \underline{x}^2} S[\delta u] \right\rangle \\
& + \frac{1}{2} \left\langle \delta \underline{u}, \frac{\partial^2 F}{\partial \underline{u}^2} \delta \underline{u} \right\rangle + \frac{1}{2} \left\langle \delta \underline{u}, S^* \frac{\partial^2 F}{\partial \underline{x} \partial \underline{u}} \delta \underline{u} \right\rangle \\
& + \frac{1}{2} \left\langle \delta \underline{u}, \frac{\partial^2 F}{\partial \underline{u} \partial \underline{x}} S[\delta \underline{u}] \right\rangle
\end{aligned} \tag{A-6}$$

where

$$S^*[\cdot] = \frac{\partial \underline{f}}{\partial \underline{u}}^T \int_t^{t_f} \Phi^T(\tau, t)\, (\cdot)\, d\tau \tag{A-7}$$

is the adjoint operator to $S[\cdot]$ and

$$T^*[\cdot] = \frac{\partial \underline{f}}{\partial \underline{u}}^T \Phi^T(t_f, t)\, (\cdot) \tag{A-8}$$

is the adjoint operator to T, i.e.

$$\langle \underline{y}, S[\underline{z}] \rangle = \langle S^*[\underline{y}], \underline{z} \rangle \tag{A-9}$$

and

$$\langle \underline{\xi}, T[\underline{z}] \rangle = \langle T^*[\underline{\xi}], \underline{z} \rangle \tag{A-10}$$

where $\underline{y}$ is an n-vector of time functions,

$\underline{z}$ is an m-vector of time functions,

and $\underline{\xi}$ is an n-vector of constants.

(See References 63,58).

The second order terms in A-6 can be expressed in a single term $\frac{1}{2}\langle \delta\underline{u}, \hat{N}\delta\underline{u}\rangle$ if $\hat{N}\delta\underline{u}$ is defined by

$$\hat{N}\,\delta\underline{u} = \left[T^{*}\frac{\partial^2\phi}{\partial\underline{x}_f^2}T + S^{*}\frac{\partial^2 F}{\partial\underline{x}^2}S + \frac{\partial^2 F}{\partial\underline{u}^2} + S^{*}\frac{\partial^2 F}{\partial\underline{x}\partial\underline{u}} + \frac{\partial^2 F}{\partial\underline{u}\partial\underline{x}}S\right]\delta\underline{u} \qquad \text{(A-11)}$$

With the operator $\hat{N}$ identified, $\hat{N}\,\underline{s}_{i-1}$ can be written as

$$\begin{aligned}\hat{N}\,\underline{s}_{i-1} = {} & \frac{\partial\underline{f}}{\partial\underline{u}}^{T}\Phi^{T}(t_f,t)\frac{\partial^2\phi}{\partial\underline{x}_f^2}\left[\int_{t_0}^{t_f}\Phi(t_f,\tau)\frac{\partial\underline{f}}{\partial\underline{u}}(\tau)\,\underline{s}_{i-1}(\tau)\,d\tau\right] \\ & + \frac{\partial\underline{f}}{\partial\underline{u}}^{T}\int_{t}^{t_f}\Phi^{T}(\lambda,t)\left[\frac{\partial^2 F}{\partial\underline{x}^2}(\lambda)\int_{t_0}^{\lambda}\Phi(\lambda,\tau)\frac{\partial\underline{f}}{\partial\underline{u}}(\tau)\,\underline{s}_{i-1}(\tau)\,d\tau\right]d\lambda \\ & + \frac{\partial\underline{f}}{\partial\underline{u}}^{T}\int_{t}^{t_f}\Phi^{T}(\tau,t)\frac{\partial^2 F}{\partial\underline{x}\partial\underline{u}}(\tau)\,\underline{s}_{i-1}(\tau)\,d\tau \\ & + \frac{\partial^2 F}{\partial\underline{u}\partial\underline{x}}\int_{t_0}^{t}\Phi(t,\tau)\frac{\partial\underline{f}}{\partial\underline{u}}(\tau)\,\underline{s}_{i-1}(\tau)\,d\tau \\ & + \frac{\partial^2 F}{\partial\underline{u}^2}\,\underline{s}_{i-1} \quad . \end{aligned} \qquad \text{(A-12)}$$

Let

$$\underline{y} = S[\underline{s}_{i-1}] = \int_{t_0}^{t}\Phi(t,\tau)\frac{\partial\underline{f}}{\partial\underline{u}}(\tau)\,\underline{s}_{i-1}(\tau)\,d\tau \quad . \qquad \text{(A-13)}$$

then

$$\dot{\underline{y}} = \int_{t_0}^{t}\dot{\Phi}(t,\tau)\frac{\partial\underline{f}}{\partial\underline{u}}(\tau)\,\underline{s}_{i-1}(\tau)\,d\tau + \frac{\partial\underline{f}}{\partial\underline{u}}(t)\,\underline{s}_{i-1}(t) \qquad \text{(A-14)}$$

But from the properties of the transition matrix

$$\dot{\Phi}(t,\tau) = \frac{\partial\underline{f}}{\partial\underline{x}}\Phi(t,\tau)\,, \qquad \text{(A-15)}$$

so that

$$\dot{\underline{y}} = \int_{t_0}^{t} \frac{\partial \underline{f}}{\partial \underline{x}}(t)\, \Phi(t,\tau) \frac{\partial \underline{f}}{\partial \underline{u}}(\tau)\, \underline{s}_{i-1}(\tau)\, d\tau + \frac{\partial \underline{f}}{\partial \underline{u}}(t)\, \underline{s}_{i-1}(t), \tag{A-16}$$

or

$$\dot{\underline{y}} = \frac{\partial \underline{f}}{\partial \underline{x}} \underline{y} + \frac{\partial \underline{f}}{\partial \underline{u}} \underline{s}_{i-1} \quad , \quad \underline{y}(t_0) = \underline{0} \quad . \tag{A-17}$$

Therefore, Equation A-12 may be written

$$\begin{aligned} \hat{N}\, \underline{s}_{i-1} = {} & \frac{\partial \underline{f}}{\partial \underline{u}}^{T} \Phi^{T}(t_f,t) \frac{\partial^2 \phi}{\partial \underline{x}_f^2} \underline{y}(t_f) \\ & + \frac{\partial \underline{f}}{\partial \underline{u}}^{T} \Big[\int_t^{t_f} \Phi^{T}(\tau,t) \frac{\partial^2 F}{\partial \underline{x}^2}(\tau)\, \underline{y}(\tau)\, d\tau \\ & \qquad + \int_t^{t_f} \Phi^{T}(\tau,t) \frac{\partial^2 F}{\partial \underline{x} \partial \underline{u}}(\tau)\, \underline{s}_{i-1}(\tau)\, d\tau \Big] \\ & + \frac{\partial^2 F}{\partial \underline{u} \partial \underline{x}}(t)\, \underline{y}(t) + \frac{\partial^2 F}{\partial \underline{u}^2} \underline{s}_{i-1}(t) \quad . \end{aligned} \tag{A-18}$$

Next define

$$\begin{aligned} \underline{\eta}(t) = {} & \Phi^{T}(t_f,t) \frac{\partial^2 \phi}{\partial \underline{x}_f^2} \underline{y}(t_f) + \int_t^{t_f} \Phi^{T}(\tau,t) \frac{\partial^2 F}{\partial \underline{x}^2}(\tau)\, \underline{y}(\tau)\, d\tau \\ & + \int_t^{t_f} \Phi^{T}(\tau,t) \frac{\partial^2 F}{\partial \underline{x} \partial \underline{u}}(\tau)\, \underline{s}_{i-1}(\tau)\, d\tau \, . \end{aligned} \tag{A-19}$$

Then

$$\begin{aligned} \dot{\underline{\eta}}(t) = {} & \frac{d}{dt}\left(\Phi^{T}(t_f,t) \right) \frac{\partial^2 \phi}{\partial \underline{x}_f^2} \underline{y}(t_f) + \int_t^{t_f} \frac{d}{dt}\left(\Phi^{T}(\tau,t) \right) \frac{\partial^2 F}{\partial \underline{x}^2}(\tau)\, \underline{y}(\tau)\, d\tau \\ & - \frac{\partial^2 F}{\partial \underline{x}^2}(t)\, \underline{y}(t) + \int_t^{t_f} \frac{d}{dt}\left(\Phi^{T}(\tau,t) \right) \frac{\partial^2 F}{\partial \underline{x} \partial \underline{u}}(\tau)\, \underline{s}_{i-1}(\tau)\, d\tau \\ & - \frac{\partial^2 F}{\partial \underline{x} \partial \underline{u}}(t)\, \underline{s}_{i-1}(t) \quad . \end{aligned} \tag{A-20}$$

Again using the properties of the transition matrix and an identity for the derivative of the inverse of a matrix we have

$$\frac{d}{dt}\left(\Phi^T(\tau,t)\right) = \left(\frac{d}{dt}\Phi(t,\tau)^{-1}\right)^T$$

$$= -\left(\Phi(\tau,t)\frac{\partial \underline{f}}{\partial \underline{x}}\Phi(\tau,t)^{-1}\Phi(\tau,t)\right)^T$$

$$= -\frac{\partial \underline{f}}{\partial \underline{x}}^T \Phi(\tau,t)^T \quad . \tag{A-21}$$

Therefore,

$$\dot{\underline{\eta}} = -\frac{\partial \underline{f}}{\partial \underline{x}}(t)^T \Big[\Phi^T(t_f,t)\frac{\partial^2 \Phi}{\partial \underline{x}_f^2}\underline{y}(t_f) + \int_t^{t_f}\Phi^T(\tau,t)\frac{\partial^2 F}{\partial \underline{x}^2}(\tau)\underline{y}(\tau)\,d\tau + \int_t^{t_f}\Phi^T(\tau,t)\frac{\partial^2 F}{\partial \underline{x}\partial \underline{u}}(\tau)\underline{s}_{i-1}(\tau)\,d\tau\Big]$$

$$-\frac{\partial^2 F}{\partial \underline{x}^2}(t)\underline{y}(t) - \frac{\partial^2 F}{\partial \underline{x}\partial \underline{u}}(t)\underline{s}_{i-1}(t) \tag{A-22}$$

or

$$\dot{\underline{\eta}} = -\frac{\partial \underline{f}}{\partial \underline{x}}^T(t)\underline{\eta}(t) - \frac{\partial^2 F}{\partial \underline{x}^2}(t)\underline{y}(t) - \frac{\partial^2 F}{\partial \underline{x}\partial \underline{u}}(t)\underline{s}_{i-1}(t). \tag{A-23}$$

From A-19,

$$\underline{\eta}(t_f) = \frac{\partial^2 \Phi}{\partial \underline{x}_f^2}\underline{y}(t_f) \quad . \tag{A-24}$$

In summary, by combining Equations A-18 and A-19, we have

$$\hat{N}\,\underline{s}_{i-1}(t) = \frac{\partial \underline{f}}{\partial \underline{u}}^T\underline{\eta}(t) + \frac{\partial^2 F}{\partial \underline{u}\partial \underline{x}}\underline{y}(t) + \frac{\partial^2 F}{\partial \underline{u}^2}\underline{s}_{i-1}(t) \tag{A-25}$$

where $\underline{y}(t)$ is obtained by a forward integration of the system

$$\dot{\underline{y}} = \frac{\partial \underline{f}}{\partial \underline{x}}\underline{y} + \frac{\partial \underline{f}}{\partial \underline{u}}\underline{s}_{i-1} \tag{A-26}$$

$$\underline{y}(t_o) = 0, \tag{A-27}$$

and $\eta(t)$ is obtained by a backward integration of the system

$$\dot{\underline{\eta}} = -\frac{\partial \underline{f}}{\partial \underline{x}}^{T} \eta - \frac{\partial^2 P}{\partial \underline{x}^2} y - \frac{\partial^2 F}{\partial \underline{x} \partial \underline{u}} \underline{s}_{i-1} \tag{A-28}$$

$$\eta(t_f) = \frac{\partial^2 \Phi}{\partial \underline{x}_f^2} y(t_f) \quad . \tag{A-29}$$

The parameter β_i is then obtained by the quotient of two quadratures:

$$\beta_i = \frac{\langle g_i , \hat{N} \underline{s}_{i-1} \rangle}{\langle \underline{s}_{i-1} , \hat{N} \underline{s}_{i-1} \rangle} \quad . \tag{A-30}$$

APPENDIX B

This appendix is a summary of the equations used to apply the MCG method to the unit mass control problem P-7 stated in Chapter IV.

The gradient as a function of λ_i is

$$q_i(\lambda_i) = -(\lambda_i + 1)\, e^{t-1} + \lambda_i + 2u_i \,. \tag{B-1}$$

Equation B-1 can be used to determine β_i as a function of λ_i. The resulting expression is

$$\beta_i(\lambda_i) = \frac{1}{\langle q_{i-1}, q_{i-1}\rangle}\left[\lambda_i^2 T_1 + \lambda_i T_2 + T_3\right] \tag{B-2}$$

where

$$T_1 = \int_0^1 \left[e^{2t-2} - 2e^{t-1} + 1 \right] dt \tag{B-3}$$

$$= -\tfrac{1}{2} - \tfrac{1}{2}e^{-2} + 2e^{-1} \tag{B-4}$$

$$T_2 = -1 - e^{-2} + 2e^{-1} - 4\int_0^1 u_i\,(e^{t-1} - 1)\, dt \tag{B-5}$$

$$T_3 = \tfrac{1}{2} - \tfrac{1}{2}e^{-2} - 4\int_0^1 u_i\,(e^{t-1} - u_i)\, dt \tag{B-6}$$

The partial derivatives needed for the gradient in the inner loop are obtained by integrating the equations

$$\dot{\underline{\ell}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \underline{\ell} + \begin{pmatrix} 0 \\ 1 \\ 2u_{i+1} \end{pmatrix} s_i(\lambda_i) \,, \quad \underline{\ell}(0) = 0 \tag{B-7}$$

$$\dot{\underline{m}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \underline{m} + \begin{pmatrix} 0 \\ 1 \\ 2u_{i+1} \end{pmatrix} \alpha_i \frac{\partial s_i(\lambda_i)}{\partial \lambda_i} \,, \quad \underline{m}(0) = 0 \tag{B-8}$$

$$\dot{\underline{n}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \underline{n} + \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} s_i(\lambda_i)^2 \,, \quad \underline{n}(0) = 0 \tag{B-9}$$

$$\dot{\underline{p}} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \underline{p} + \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} s_i(\mathcal{A}_i)\, \alpha_i \frac{\partial s_i(\mathcal{A}_i)}{\partial \mathcal{A}_i} \tag{B-10}$$

where

$$\underline{\ell} = \frac{\partial \underline{x}_{i+1}}{\partial \alpha_i} \tag{B-11}$$

$$\underline{m} = \frac{\partial \underline{x}_{i+1}}{\partial \mathcal{A}_i} \tag{B-12}$$

$$\underline{n} = \frac{\partial^2 \underline{x}_{i+1}}{\partial \alpha_i^2} \tag{B-13}$$

$$\underline{p} = \frac{\partial^2 \underline{x}_{i+1}}{\partial \alpha_i \partial \mathcal{A}_i} \tag{B-14}$$

and

$$u_{i+1} = u_i + \alpha_i \Big[(\mathcal{A}_i + 1) e^{t-1} - \mathcal{A}_i - 2u_i + (\mathcal{A}_i^2 T_1 + \mathcal{A}_i T_2 + T_3) \frac{s_{i-1}}{\langle q_{i-1}, q_{i-1} \rangle} \Big] \tag{B-15}$$

Terms involving e arise from the analytic solution of the adjoint equations

$$\dot{\lambda}_1 = 0 , \qquad \lambda_1(1) = \mathcal{A} \tag{B-16}$$

$$\dot{\lambda}_2 = -\lambda_1 + \lambda_2 , \qquad \lambda_2(1) = -1 \tag{B-17}$$

$$\dot{\lambda}_3 = 0 , \qquad \lambda_3(1) = 1 . \tag{B-18}$$