

INFORMATION TO USERS

This dissertation was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.
2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.
3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.
4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

University Microfilms

300 North Zeeb Road
Ann Arbor, Michigan 48106
A Xerox Education Company

73-9479

RAJTORA, Stanley Gene, 1945-
APPLICATION OF PROJECTION OPERATOR TECHNIQUES
TO OPTIMAL CONTROL PROBLEMS.

Iowa State University, Ph.D., 1972
Engineering, aeronautical

University Microfilms, A XEROX Company, Ann Arbor, Michigan

Application of projection operator techniques
to optimal control problems

by

Stanley Gene Rajtora

A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of
The Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Majors: Aerospace Engineering
Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department s

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1972

PLEASE NOTE:

Some pages may have

indistinct print.

Filmed as received.

University Microfilms, A Xerox Education Company

TABLE OF CONTENTS

	Page
CHAPTER 1. INTRODUCTION AND HISTORICAL BACKGROUND	1
CHAPTER 2. OPTIMAL CONTROL PROBLEMS WITH TERMINAL STATE CONSTRAINTS	6
Problem Development	6
Methods of Application	13
CHAPTER 3. OPTIMAL CONTROL PROBLEMS WITH UNSPECIFIED INITIAL CONDITIONS	23
Problem Development	23
Numerical Examples	30
Forced rod problem	31
Variable launch site rocket launch problem	38
CHAPTER 4. OPTIMAL BRANCHED TRAJECTORY PROBLEMS	42
Problem Development	42
Command Service Module Abort Problem	56
CHAPTER 5. OPTIMAL CONTROL PROBLEMS WITH INEQUALITY CONSTRAINTS	68
Problem Development	68
Numerical Examples	89
Rectangular wing problem	93
Panel flutter problem	107
CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY	126
REFERENCES	142
ACKNOWLEDGMENTS	146
APPENDIX: FORTRAN IV LISTING OF QUADRATIC INTERPOLATING ONE-DIMENSIONAL SEARCH	147

CHAPTER 1. INTRODUCTION AND HISTORICAL BACKGROUND

One of the primary objectives of the engineer is to obtain the best possible results from the hardware and equipment available to him. If the desired results can be translated into a well-defined cost functional, that is, a mathematical quantity that is to be minimized, and if the hardware to be used can be modeled as a dynamical system governed by a set of control functions, this general objective can be achieved by solving an optimal control problem. The stringent requirements placed on aerospace systems has greatly increased the complexity of the mathematical model by the addition of such things as inequality constraints on the control functions or even inequality constraints on state variables of the system. However, it is largely this design environment which has caused much research to be done and subsequently much knowledge to be gained in the area of optimal control.

Much has been written on the theoretical aspects of optimal control.¹⁻³ The necessary and sufficiency conditions for solutions to optimal control problems are now fairly well understood and readily available. In general, however, analytical solutions to most optimal control problems are very hard to find, and the enormous amount of theory which has been derived over the past years gives little satisfaction to the practicing engineer who diligently has modeled the physical situation into a mathematical model and now has less understanding of the mathematical model than the original physical situation. The answer to this predicament has been the use of numerical techniques to solve the above mentioned optimal control problem, and for that reason these numerical techniques deserve further study. The

advent of the high speed computer has brought numerical techniques into prominence the past few years. The advances in both computer technology and numerical algorithms will surely safeguard this place of prominence for the years to come.

Numerical techniques for solving optimal control problems have classically been broken down into two groups, the indirect techniques and the direct techniques. Indirect techniques are characterized by attempting to solve the two point boundary value problem arising from the application of the necessary conditions to the optimal control problem. No further mention will be made of indirect techniques except to note that this area is generally divided into two topics, variation of extremals and quasi-linearization, and to indicate that a large amount of literature has been published on these two topics.⁴ The subject of this investigation, the direct techniques, is normally considered to be more successful than the indirect techniques for solving optimal control problems. The direct techniques are generally characterized by an altering of the control time history such that at each iteration the cost functional is decreased.

The development of the direct optimization techniques has proceeded along two relatively distinct paths. One path was pioneered by Kelley,⁵ while Bryson and Denham⁶ were opening the frontier in a different direction. Kelley advanced the concept of using a penalty function to satisfy terminal state constraints. This method necessitates the addition of a term to the cost functional which measures the dissatisfaction of the terminal state constraints. This technique then, transforms the constrained optimization problem into an unconstrained optimization problem. Perhaps

the main drawback to this very straightforward approach is that the generally large penalty constant needed to provide the necessary constraint satisfaction also makes the dynamics very sensitive to changes in the control. To overcome this difficulty Fiacco and McCormick⁷ proposed solving a sequence of unconstrained problems for which the penalty constant is successively increased. Although their work dealt with constraints in conjunction with parameter optimization problems, it is easily extended to the case of terminal state constraints on optimal control problems. This suggestion has stimulated many ideas for picking the initial penalty constant and for updating the value of the penalty constant from one unconstrained subproblem to the next. One additional help in overcoming this sensitivity has been the use of gradient algorithms more sophisticated than steepest descent such as conjugate gradients⁸ and the Davidon technique.⁹

The ideas originally proposed by Bryson and Denham for handling terminal state constraints in optimal control problems have become known as projection operator techniques. Though not nearly as straightforward an approach as the penalty function technique, its sophistication allows it to be much more versatile than the use of penalty functions. Later, Denham and Bryson¹⁰ extended their original work to handle inequality constraints on state or control variables. Sinnott and Luenberger¹¹ were the first to use conjugate gradients in conjunction with the projection operator to handle terminal state constraints. Although the analytical development they used was somewhat different than that used by Bryson and Denham,⁶ the equivalence of the resulting projection operators has been

shown by Willoughby and Pierson.¹² Shortly following the efforts of Sinnott and Luenberger,¹¹ Mehra and Bryson¹³ used a projection technique in conjunction with the conjugate gradients method for V/STOL flight path optimization. Horwitz and Sarachik,¹⁴ using a more rigorous development of the projection operator, proved the convergence of Davidon's method for the linear-quadratic problem with linear terminal state constraints.

Projection operator techniques now enjoy widespread popularity in many engineering fields. Historically, a projection operator was first developed for a finite-dimensional Euclidean space and used in connection with nonlinear programming problems.^{15,16} More recently, Junkins¹⁷ has shown the equivalence of gradient projection optimization techniques and minimum norm optimization techniques for both parameter optimization problems and optimal control problems with terminal state constraints. Rozendaal¹⁸ and Gera¹⁹ have both applied gradient projection technology to optimal branched trajectory problems. McCart, Haug, and Streeter²⁰ have used projection techniques in the field of structural optimization problems with inequality constraints on a function of the control.

One purpose of this investigation is to extend the use of gradient projection techniques to still a new class of optimal control problems, the class with unspecified values for some of state variables at the initial time. Both branched trajectory problems and problems involving inequality constraints on the control are solved using a new technique for applying projection operators utilizing a one-dimensional minimization along the constraint surface. This new technique, an automated gradient projection algorithm has been proposed and documented by Rajtora and

Pierson.²¹ Finally, additional insight is given in the development and application of the projection operators for the above-mentioned optimal control problems.

The use of the projection operator to handle terminal constraints is dealt with in Chapter 2, and some philosophies for applying the projection operator to a specific problem are reviewed. In Chapter 3 the projection operator is developed for the case of unknown initial state values and applied to two rather straightforward example problems. Branched trajectories and problems involving inequality constraints are discussed in Chapters 4 and 5, respectively, and appropriate numerical examples are given. Finally, Chapter 6 contains a summary of this investigation, appropriate conclusions based on numerical experience, and suggestions for further study.

CHAPTER 2. OPTIMAL CONTROL PROBLEMS WITH TERMINAL STATE CONSTRAINTS

Problem Development

The terminal state constraint, which specifies values of the terminal states or functional relationships between the terminal states, was the first type of constraint to be treated using the projection technique. It is perhaps the commonest of all constraints placed on optimal control problems. Indeed, it is hard to imagine a trajectory optimization problem for which at least one of the state variables is not specified at the final time.

Consider the problem of determining the r -vector control function $u(t)$, $t_0 \leq t \leq t_f$, which minimizes the cost functional

$$J = \phi[x(t_f)] + \int_{t_0}^{t_f} L(x, u, t) dt, \quad (2.1)$$

subject to the n^{th} -order linear dynamical system

$$\dot{x} = G(t)x(t) + B(t)u(t), \quad x(t_0) = x_0, \quad (2.2)$$

with x_0 , t_0 , and t_f given, and subject to the p ($p \leq n$) terminal state constraints

$$Ax(t_f) = C, \quad (2.3)$$

where $G(t)$ and $B(t)$ are $n \times n$ and $n \times r$ time-varying matrices, respectively, A is a constant $p \times n$ matrix of rank p , and C is a p -vector of constants. As was mentioned in the Introduction, the projection operator for the above problem has been developed essentially along three different paths.

Because of the generality of their work, the development here of the projection operator will follow that of Horwitz and Sarachik.¹⁴ This will provide a foundation for the other classes of problems to be treated in the following three chapters.

Let $u_1(t)$ and $u_2(t)$ be controls which produce trajectories $x_1(t)$ and $x_2(t)$, respectively. Furthermore, let the variations of the control and state vectors be defined by

$$\delta u(t) = u_2(t) - u_1(t)$$

and

$$\delta x(t) = x_2(t) - x_1(t).$$

If both controls $u_1(t)$ and $u_2(t)$ satisfy Eq. (2.3), then

$$A\delta x(t_f) = A[x_2(t_f) - x_1(t_f)] = 0. \quad (2.4)$$

Since the system dynamics (2.2) are linear, the variations δu and δx are related by the equation

$$\dot{\delta x}(t) = G(t)\delta x(t) + B(t)\delta u(t), \quad \delta x(t_0) = 0, \quad (2.5)$$

for which

$$\delta x(t) = \int_{t_0}^t \tilde{\Phi}(t, \tau) B(\tau) \delta u(\tau) d\tau, \quad (2.6)$$

where $\tilde{\Phi}(t, \tau)$ is the state transition matrix of the linear dynamical system (2.2) or (2.5). For convenience, let the $p \times r$ time-varying matrix $\tilde{F}(\tau)$ be defined by

$$\xi(\tau) = A\tilde{Q}(t_f, \tau)B(\tau). \quad (2.7)$$

so that Eq. (2.4) becomes

$$\int_{t_0}^{t_f} \xi(\tau) \delta u(\tau) d\tau = 0. \quad (2.8)$$

Let U be a set of r -vector control functions $u(t)$, $t_0 \leq t \leq t_f$.^{**} Define \tilde{U} as that subset of U such that all elements in \tilde{U} satisfy Eq. (2.8). \tilde{U} is called the set of admissible control variations for this optimal control problem. If the rows of $\xi(t)$, which are denoted by the row vectors $\xi_1, \xi_2, \dots, \xi_p$, are linearly independent, they span a p -dimensional subspace, call it U^* , of U . Thus, from Eq. (2.8), the set of admissible control variations \tilde{U} , can be defined by the relation

$$\tilde{U} = \{ \delta u \in U : \int_{t_0}^{t_f} \xi_i \delta u dt = 0, i = 1, 2, \dots, p \} \quad (2.9)$$

or

$$\tilde{U} = \{ \delta u \in U : \langle \xi_i', \delta u \rangle = 0, i = 1, 2, \dots, p \}$$

where $\langle v, w \rangle$ is defined as $\int_{t_0}^{t_f} v' w dt$ with integration limits t_0 to t_f and is the inner product on U , and the symbol $'$ means transpose. \tilde{U} , being the orthogonal complement²² of U^* , is also a subspace of U . In addition, if U is a Hilbert space, U is the direct sum of \tilde{U} and U^* , $U = \tilde{U} \oplus U^*$; that is

^{**}A control variation δu , being the difference of two elements in U , is also a member of U .

for every $\delta u^* \in U^*$ such that

$$\delta u = \tilde{\delta u} + \delta u^*. \quad (2.10)$$

Define the projection operator P as $P[\delta u] = \tilde{\delta u}$, where according to the previous notation $\tilde{\delta u}$ belongs to \tilde{U} . The subspace U^* can be defined as

$$U^* = \{ \delta u \in U : \delta u = \sum_{i=1}^p \eta_i \xi'_i \} \quad (2.11)$$

$$= \{ \delta u \in U : \delta u = \xi' \eta \},$$

where η is a constant p -vector with elements $\eta_1, \eta_2, \dots, \eta_p$.

Making use of the relationships found in Eq. (2.9-2.11), the following set of equations is generated.

$$\langle \xi'_1, \xi' \eta \rangle = \langle \xi'_1, \delta u \rangle$$

$$\langle \xi'_2, \xi' \eta \rangle = \langle \xi'_2, \delta u \rangle$$

$$\begin{array}{c} \cdot \\ \vdots \\ \cdot \end{array} \quad \begin{array}{c} \cdot \\ \vdots \\ \cdot \end{array}$$

$$\langle \xi'_p, \xi' \eta \rangle = \langle \xi'_p, \delta u \rangle.$$

The above set of equations can be written more concisely as

$$\left[\int_{t_0}^{t_f} \xi(t) \xi'(t) dt \right] \eta = \int_{t_0}^{t_f} \xi(t) \delta u(t) dt.$$

Solving for η ,

$$\eta = \left[\int_{t_0}^{t_f} \xi(t) \xi'(t) dt \right]^{-1} \int_{t_0}^{t_f} \xi(t) \delta u(t) dt.$$

From Eq. (2.11), δu^* can be written as

$$\delta u^* = \xi' \left[\int_{t_0}^{t_f} \xi \xi' dt \right]^{-1} \int_{t_0}^{t_f} \xi \delta u dt. \quad (2.12)$$

Since $\tilde{\delta u}$ is equal to δu minus δu^* from Eq. (2.10),

$$\tilde{\delta u} = \delta u - \xi' \left[\int_{t_0}^{t_f} \xi \xi' dt \right]^{-1} \int_{t_0}^{t_f} \xi \delta u dt.$$

The actual projection operator defined by the above relation is

$$P[.] = I[.] - \xi' \left[\int_{t_0}^{t_f} \xi \xi' dt \right]^{-1} \int_{t_0}^{t_f} \xi [.] dt. \quad (2.13)$$

It should be remembered that the project operator given by Eq. (2.13) was derived under the assumptions of linear dynamics and linear terminal state constraints of the form $Ax(t_f)=C$.

Now, consider the problem of determining the r -vector control function $u(t)$, $t_0 \leq t \leq t_f$, which minimizes the cost functional

$$J = \phi[x(t_f)] + \int_{t_0}^{t_f} L(x, u, t) dt, \quad (2.14)$$

subject to the n^{th} -order nonlinear dynamical system

$$\dot{x}(t) = f(x, u, t), \quad x(t_0) = x_0, \quad (2.15)$$

with x_0 , t_0 , and t_f given, and subject to the p ($p \leq n$) terminal state constraints

$$\psi[x(t_f)] = 0. \quad (2.16)$$

Linearization of the system dynamics (2.15) about a nominal $u(t)$ and $x(t)$ gives

$$\dot{\delta x} = f_x(x, u, t)\delta x(t) + f_u(x, u, t)\delta u(t), \quad \delta x(t_0) = 0, \quad (2.17)$$

for which

$$\delta x(t) = \int_{t_0}^t \Phi(t, \tau) f_u(\tau) \delta u(\tau) d\tau, \quad (2.18)$$

where $\Phi(t, \tau)$ is the state transition matrix of linear system (2.17), f_x and f_u are matrices with elements $\partial f_i / \partial x_j$, and $\partial f_i / \partial u_j$, respectively, evaluated on the nominal path. In addition, if each element of the vector $\psi[x]$ is differentiable and if each iteration in the process satisfies Eq. (2.16), then near the nominal $x(t_f)$,

$$\psi_x[x(t_f)]\delta x(t_f) \approx \psi[x(t_f) + \delta x(t_f)] - \psi[x(t_f)] = 0. \quad (2.19)$$

Combining Eq. (2.18) with Eq. (2.19) yields

$$\int_{t_0}^{t_f} \psi_x \Phi(t_f, \tau) f_u(\tau) \delta u(\tau) d\tau = 0. \quad (2.20)$$

For convenience, let the $p \times r$ time-varying matrix $\zeta(\tau)$ be defined by

$$\zeta(\tau) = \psi_x \Phi(t_f, \tau) f_u(\tau), \quad (2.21)$$

so that Eq. (2.20) becomes

$$\int_{t_0}^{t_f} \zeta(\tau) \delta u(\tau) d\tau = 0. \quad (2.22)$$

Eq. (2.22) is now analogous to Eq. (2.8). Therefore, the projection operator (2.13) developed earlier can be used to determine admissible

control variations for the present optimal control problem involving nonlinear dynamics and nonlinear terminal state constraints with the results being valid wherever the linearization of nonlinear dynamics and nonlinear terminal state constraints is valid. As a consequence of this

$$\delta \tilde{u} = \delta u - \zeta \left[\int_{t_0}^{t_f} \zeta \zeta' dt \right]^{-1} \int_{t_0}^{t_f} \zeta \delta u dt.$$

The projection operator defined by the above relation is the same as Eq. (2.13),

$$P[.] = I[.] - \zeta' \left[\int_{t_0}^{t_f} \zeta \zeta' dt \right]^{-1} \int_{t_0}^{t_f} \zeta [.] dt. \quad (2.23)$$

However, the application of it is much more difficult due to the linearity requirements just mentioned. Methods for applying this projection operator are discussed in detail in a later section of this chapter.

Now if the dynamical system (2.15) is adjoined to the cost functional with the n-vector multiplier function $\lambda(t)$, the variation in J due to a variation in the control vector for fixed t_0 , t_f , and x_0 is²³

$$\delta J = [(\phi_x - \lambda') \delta x]_{t=t_f} + \int_{t_0}^{t_f} [(H_x + \dot{\lambda}') \delta x + H_u \delta u] dt, \quad (2.24)$$

where the function H (the Hamiltonian) is defined as

$$H[x(t), u(t), \lambda(t), t] = L[x(t), u(t), t] + \lambda'(t) f[x(t), u(t), t].$$

Choosing

$$\dot{\lambda}'(t) = -H_x = -L_x - \lambda' f_x \text{ and } \lambda'(t_f) = \phi_x,$$

simplifies Eq. (2.24) to

$$\delta J = \int_{t_0}^{t_f} H_u' \delta u \, dt = \langle H_u', \delta u \rangle.$$

Since H_u' , can be written as $H_u^{*'} + \tilde{H}_u'$, where $H_u^{*'}$ belongs to U^* , and \tilde{H}_u' belongs to \tilde{U} , the expression for δJ becomes

$$\delta J = \langle H_u^{*'} + \tilde{H}_u', \delta u \rangle = \langle \tilde{H}_u', \delta \tilde{u} \rangle,$$

since δu must belong to \tilde{U} to be an admissible control variation.

For an extremum, δJ must be zero for arbitrary $\delta \tilde{u}(t)$; this can happen only if $\tilde{H}_u' = 0$, for $t_0 \leq t \leq t_f$. If θ_i and $p_i(t)$ represent the stepsize and direction of search, respectively, then $u_{i+1} = u_i + \theta_i p_i$, and it becomes quite natural to let $p_i = -\tilde{H}_{u_i}'$ to insure that δJ is negative for each iteration. This is, of course, the steepest descent technique of picking the direction of search.

Methods of Application

Bryson and Denham⁶ were among the first to apply gradient projection technology to optimal control problems. A complete description of their use of the projection operator is not feasible here. It is important to note, however, that the iteration procedure starts with a nominal control which does not satisfy the terminal boundary conditions, at each step in the minimization procedure the control variation is constrained, and the change in the terminal constraint satisfaction is specified. The control variation is constrained by specifying the parameter dP defined by the equation

$$(dP)^2 = \int_{t_0}^{t_f} \delta u^T(t) W(t) \delta u(t) dt,$$

to insure that the control variation $\delta u(t)$ is small enough for the linearization of the dynamical equations to be valid. $W(t)$ is an arbitrary positive definite $r \times r$ matrix of weighting functions chosen to improve convergence of the steepest descent procedure. The change in the terminal state constraint satisfaction can not be specified completely independent of dP since a requirement on valid linearization of the system dynamics is also going to require a small change in the constraint satisfaction. It would appear that the terminal state constraints could also be approached so slowly that the cost functional could be decreased below the actual minimum and later have to be increased in order to satisfy the terminal state constraints. The obvious drawbacks of this method are having to choose the parameter dP and all the changes of the constraint satisfaction as well as specifying $W(t)$ if the 'optimizer' wishes to use anything but simplest steepest descent algorithm. Although these drawbacks are not insurmountable and a number of problems⁶ have been solved by this method, it could certainly not be classified as a fully automated procedure.

Sinnott and Luenberger¹¹ apply the projection operator (2.23) in conjunction with a conjugate gradient algorithm. Their method of actually implementing the projection operator is, in brief, to:

- (a) choose θ_i such that $J[u_i(t) + \theta_i p_i(t)]$ as a function of θ_i ,
is a minimum;

$$(b) \quad \text{let } \hat{u}_{i+1}(t) = u_i(t) + m_i \theta_i p_i(t); \quad (2.25)$$

$$(c) \quad \text{let } \psi_i = \psi [\hat{x}(t_f)] , \text{ where } \hat{x}(t_f) \text{ corresponds to } \hat{u}_{i+1}(t);$$

$$(d) \quad \text{let } \Delta u(t) = -\zeta(t) \left[\int_{t_0}^{t_f} \zeta(t) \zeta(t) dt \right]^{-1} \psi_i; \text{ and}$$

$$(e) \quad \text{let } u_{i+1}(t) = \hat{u}_{i+1}(t) + n_i \Delta u(t). \quad (2.26)$$

Steps a and b constitute a one-dimensional minimization along the projected direction of search p_i . For linear dynamics and linear terminal state constraints, the stepsize parameter m_i is set equal to one since all controls along $u_i + \theta_i p_i$ satisfy the terminal state constraints. This is, of course, not true for the case of nonlinear terminal state constraints or a nonlinear dynamical system. If the system dynamics are nonlinear or the terminal state constraints are nonlinear, the value of m_i is greater than zero due to the minimization process, but typically less than one due to linearity restrictions. This is explained in detail later in this chapter. Steps a and b shall be referred to as 'proceeding parallel to the constraint surface'. Steps c and d determine the control variation $\Delta u(t)$ necessary to change the value of the function $\psi[x(t_f)]$ by the amount $-n_i \psi_i$.

For this reason, these two steps shall be referred to as 'proceeding orthogonally to the constraint surface'.

Assuming it is desired to change the constraint function value by the amount $\Delta \psi$, the question of finding the correct control variation $\delta u(t)$ to give this terminal constraint change still remains. Note that for the

case of linear dynamics and linear terminal state constraints

Eqs. (2.4), (2.6), (2.7) and Eq. (2.12), $\delta u^*(t)$ can be written as

$$\delta u^* = \xi' \left[\int_{t_0}^{t_f} \xi \xi' dt \right]^{-1} A \delta x(t_f) .$$

Here it is easily seen that δu^* is uniquely determined by the change in the function $Ax(t_f)$ produced by δu . Since the change in the value of $Ax(t_f)$ produced by δu is exactly equal to the change of $Ax(t_f)$ produced by δu^* , the δu in question takes on the form

$$\delta u = \delta u^* = \xi' \left[\int_{t_0}^{t_f} \xi \xi' dt \right]^{-1} \Delta \psi . \quad (2.27)$$

If it is desired to change the value of $Ax(t_f)$ from ψ_i to zero, then $\Delta \psi$ is set equal to $-\psi_i$, and the relation expressed in step d is obtained as a result. Ordinarily, either the system dynamics or the terminal state constraints or both are nonlinear so that only a partial correction back to the constraints is desirable. In this case, $\Delta \psi$ is set equal to $-n_i \psi_i$, where n_i is a scalar stepsize parameter greater than zero and less than one. The end result is Eq. (2.26).

Methods for selecting the stepsize parameters m_i and n_i for problems involving either nonlinear system dynamics or nonlinear terminal state constraints have very little theoretical basis. Sinnott and Luenberger¹¹ suggest that if the constraint (2.16) is not satisfied, parameters m_i and n_i should be chosen so that (a) $\|\psi\|$ decreases at each step, where $\|\cdot\|$ is some suitable norm, and (b) the performance index J decreases at each step.

If $\|\psi\| < \epsilon$ for some specified $\epsilon > 0$, the constraint is considered satisfied and motion takes place along the constraint surface. Although the intent of these steps is clear, the implementation of them is not straightforward.

Mehra and Bryson¹³ had success at applying a slightly different conjugate gradient algorithm to V/STOL flight path optimization problems. They suggest that the parameter m_i in Eq. (2.25) in proceeding 'parallel to the constraints', can be determined by insuring that the constraint violation at the beginning and end of the step are not significantly different, i.e., the linearization has not been violated. Similarly, a linearization check is made on the orthogonal correction step (2.26) by comparing the actual constraint variation $\psi[x + \delta x] - \psi[x]$ to the quantity $\psi_x \delta \bar{x}(t_f)$, where $\delta \bar{x}(t_f)$ is the result of integrating the linear system (2.17). The parameter n_i is reduced from unity only if there is a significant difference between these quantities. This method of stepsize selection, while reasonable and well defined, does not provide full utilization of each step in the iteration process.

Willoughby²⁴ has extended the work of Sinnott and Luenberger¹¹, in which only linear terminal state constraints were considered, to nonlinear terminal state constraints. He advises that m_i be reduced from unity only if (a) the correction in Eq. (2.26) leads to greater rather than smaller constraint violation, or (b) the value of the cost functional after correcting the control in Eq. (2.26) increases. He found that this criterion does at least for the test case solved, lead to faster convergence due to the larger stepsizes taken.

More recently, Rajtóra and Pierson²¹ have proposed a stepsize selection technique which incorporates a one-dimensional minimization to automatically give the exact stepsize to obtain the greatest decrease of the cost functional at each step in the iterative numerical process while satisfying the constraints at all times. Since this technique will be used in the next three chapters, it is described in detail here. For notational convenience let u_ψ denote a control obtained by repeatedly applying Eq. (2.26) to u until the terminal constraints are satisfied. The proposed stepsize selection technique is that a one-dimensional minimization of the cost functional be performed along the projected direction of search p_i . That is, the stepsize parameter θ_i is chosen such that $J\{[u_i + \theta_i p_i]_\psi\}$ as a function of θ_i is a minimum. This necessitates the use of a one-dimensional minimization routine requiring only function value information. Rajtóra and Pierson²¹ used this technique only in conjunction with steepest descent. However, they noted that it has sound theoretical justification since proofs of conjugate direction gradient methods for problems with linear dynamics and quadratic performance index require a one-dimensional minimization at each step, and more general optimal control problems can often be approximated by such a linear-quadratic problem near the minimum.

The automated gradient projection algorithm can be stated as:

- (a) selection of an estimate $u_0(t)$ of the optimal control;
- (b) orthogonal correction of $u_0(t)$ back to the constraint by repeated use of Eq. (2.26) until $\|\psi\| < \epsilon$, for a specified $\epsilon > 0$;
- (c) letting $p_i = -\tilde{H}'_{u_i}$ (steepest descent);

(d) choosing θ_i such that $J\{[u_i(t) + \theta_i p_i(t)]_\psi\}$, as a function of θ_i , is a minimum;

(e) setting $u_{i+1}(t) = [u_i(t) + \theta_i p_i(t)]_\psi$;

and (f) repeating items c through e until convergence is obtained. A schematic representation of the projection of a control variation δu and correction of a control back to the constraint is presented in Fig. 1. It should be noted that in this case two orthogonal corrections are required to return to the constraint surface. Step b of the above procedure requires choosing parameters n_i (or possibly a number of them depending on the number of orthogonal corrections to the constraint which are required) and ϵ . The technique used by Mehra and Bryson¹³ for choosing n_i which was discussed earlier in this section has proven most satisfactory. The reduction of n_i by a factor of one-half, should the linearity requirements of Mehra and Bryson not be satisfied, has also proven most satisfactory. Choosing too large a reduction factor unnecessarily increases the required computational effort. However, a relatively wide range of values may be chosen for this reduction factor since it is not critical to the overall result of step b. In any event, it is seldom used except in correcting the original guessed nominal control back to the constraint surface.

Although a value for the constraint satisfaction parameter ϵ must be specified, its choice is also not critical and should not be considered as a drawback to the general procedure. It must be small enough to generate a smooth curve for $J[(u_i + \theta_i p_i)_\psi]$ versus θ_i over which the one-dimensional minimization can operate and yet must not be so small as to require additional unnecessary corrections back to the constraint surface. Preliminary

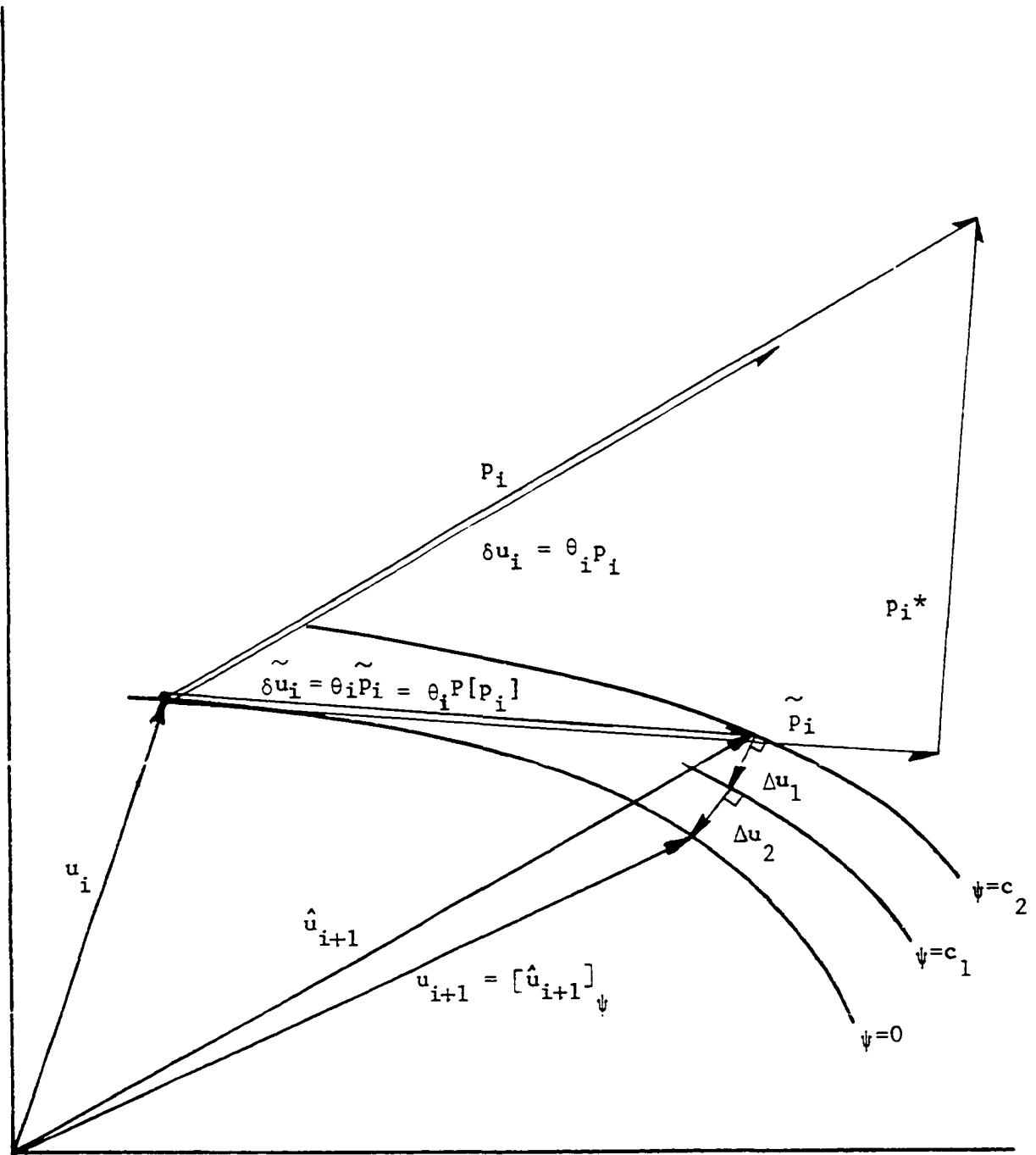


Fig. 1. Schematic representation of the i^{th} iteration for the projection technique.

numerical experience indicates that the value of ϵ can be made an order of magnitude larger or smaller without affecting the accuracy of the one-dimensional minimization or affecting the computational effort required in correcting back to the constraint. Ordinarily, a physical understanding of the magnitudes of the involved state variables provides insight into specifying values of ϵ .

A flowchart for performing one step of the above algorithm is presented in Fig. 2. The numerical integration required in blocks 1 and 2 is carried out using a standard 4th-order Runge-Kutta fixed stepsize program. The integrals in block 3 and 4 are evaluated using Simpson's rule. The matrix inversion in block 3 is carried out using a double pivoting Gauss elimination process.

A listing of the one-dimensional minimization subroutine represented by block 5 in Fig. 2 is given in the Appendix. The subroutine is separated into two main parts. The first part of the program bounds the minimum, while the second part of the program uses a quadratic interpolating scheme to actually find the minimum. Referring to Fig. 1 the control u is corrected back to the constraint by repeated usage of Eq. (2.26), which, in turn, requires execution of blocks 1 through 3 of the flowchart on Fig. 2.

The example problem to which this technique was applied was the rocket launch problem treated by Lasdon, Mitter, and Waren.⁸ Excellent results were obtained after only five steps and a total execution time on an IBM 360/65 computer of 19.2 seconds.²¹ Due to the success of this step-size selection procedure on this example problem, it will be adapted to more sophisticated classes of optimal control problems in the following chapters.

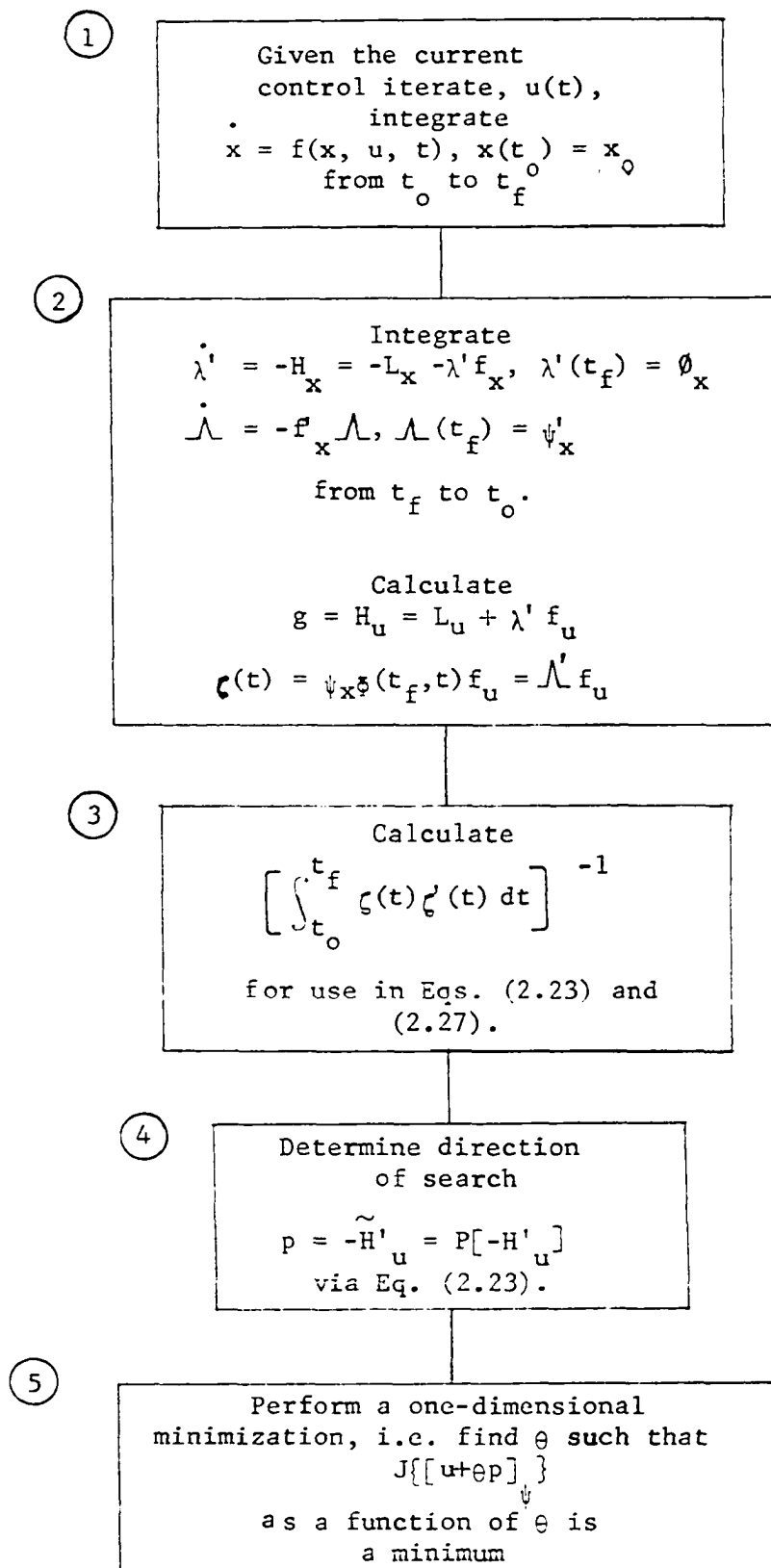


Fig. 2. Flowchart for the automated gradient projection algorithm.

CHAPTER 3. OPTIMAL CONTROL PROBLEMS WITH UNSPECIFIED INITIAL CONDITIONS

Problem Development

The class of optimal control problems with unspecified initial conditions on some of the state variables has just recently enjoyed some popularity. The V/STOL flight path optimization study conducted by Mehra and Bryson¹³ involved an unspecified flight path angle at the initial time. Tripathi and Narendra,²⁵ in an attempt to duplicate the results of the previous authors, found surprisingly different results. Obviously, this is one area where much investigation is yet to be done. The field of structural optimization is one field other than trajectory optimization where results from this investigation are of great importance. One example of a structural optimization problem is treated later in this chapter, and still others involving unknown initial conditions are presented as numerical examples in Chapter 5.

Consider the problem of determining the r -vector control function $u(t)$, $t_0 \leq t \leq t_f$, and the k -vector ($k < n$) set of initial conditions σ which minimize the cost functional

$$J = \phi[x(t_f)] + \int_{t_0}^{t_f} L(x, u, t) dt \quad (3.1)$$

subject to the n^{th} -order nonlinear dynamical system

$$\begin{aligned} \dot{x}(t) &= f(x, u, t), \quad x(t_0) = x_0, \\ \{x_{0_i} &= \sigma_i, \quad i = 1, 2, \dots, k, \text{ unspecified}\} \\ \{x_{0_i} &, \quad i = k + 1, k + 2, \dots, n, \text{ given}\} \end{aligned} \quad (3.2)$$

with t_0 and t_f given, and subject to the p ($p \leq n$) terminal state constraints

$$\psi[x(t_f)] = 0. \quad (3.3)$$

Linearization of the dynamics (3.2) about a nominal $u(t)$ and $x(t)$ gives

$$\delta \dot{x} = f_x(x, u, t) \delta x(t) + f_u(x, u, t) \delta u(t) \quad (3.4)$$

with t_0 and t_f given, for which

$$\delta x(t) = [\bar{\Phi}_1, \bar{\Phi}_2, \dots, \bar{\Phi}_k] \delta \sigma + \int_{t_0}^t \bar{\Phi}(t, \tau) f_u(\tau) \delta u(\tau) d\tau, \quad (3.5)$$

where $\bar{\Phi}_i$ is the i^{th} column of the state transition matrix $\bar{\Phi}(t, t_0)$, and all other notation is precisely the same as in Chapter 2. Again, if each element of the vector $\psi[x]$ is differentiable and if each step in the iteration process satisfies Eq. (3.3), then near a nominal $x(t_f)$,

$$\psi_x \delta x \approx \psi[x + \delta x] - \psi[x] = 0. \quad (3.6)$$

Combining Eq. (3.5) with Eq. (3.6) yields

$$\psi_x [\bar{\Phi}_1, \bar{\Phi}_2, \dots, \bar{\Phi}_k]_{t=t_f} \delta \sigma + \int_{t_0}^{t_f} \psi_x \bar{\Phi}(t_f, \tau) f_u(\tau) \delta u(\tau) d\tau = 0. \quad (3.7)$$

For convenience, let the $p \times r$ time-varying matrix $\xi(\tau)$ be defined by $\xi(\tau) = \psi_x \bar{\Phi}(t_f, \tau) f_u(\tau)$ and the $p \times k$ constant matrix γ be defined by $\gamma = \psi_x [\bar{\Phi}_1, \bar{\Phi}_2, \dots, \bar{\Phi}_k]$, so that Eq. (3.7) becomes

$$\gamma \delta \sigma + \int_{t_0}^{t_f} \xi(\tau) \delta u(\tau) d\tau = 0. \quad (3.8)$$

It should be noted here to avoid possible confusion that the following development of the projection operator should be for a linear dynamical system with linear terminal state constraints of the form $Ax(t_f) = C$, where A is a $p \times n$ constant matrix of rank p and C is a vector of constants, as was done in Chapter 2 and then should be extended to the present case of nonlinear dynamics and nonlinear terminal state constraints. Instead, the projection operator is developed directly from the linearized dynamics (3.4) and linearized terminal state constraints (3.6) so that the end result will be valid only in the region where the linearization is valid.

Let Ω be the Cartesian product of Σ and U , i.e. $\Omega = \Sigma \times U$, where Σ is a set of k -vector initial conditions σ , and U is a set of r -vector control functions $u(t)$, $t_0 \leq t \leq t_f$. Therefore, Ω contains vectors w of the form

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \\ w_{k+1}(t) \\ w_{k+2}(t) \\ \vdots \\ w_{k+r}(t) \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_k \\ u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix} = \begin{bmatrix} \sigma \\ u(t) \end{bmatrix}$$

The variation of a vector w belonging to Ω , defined in the natural way as

$$\delta w = \begin{bmatrix} \delta w_1 \\ \delta w_2 \\ \vdots \\ \delta w_k \\ \delta w_{k+1}(t) \\ \delta w_{k+2}(t) \\ \vdots \\ \delta w_{k+r}(t) \end{bmatrix} = \begin{bmatrix} \delta \sigma_1 \\ \delta \sigma_2 \\ \vdots \\ \delta \sigma_k \\ \delta u_1(t) \\ \delta u_2(t) \\ \vdots \\ \delta u_r(t) \end{bmatrix} = \begin{bmatrix} \delta \sigma \\ \delta \sigma(t) \end{bmatrix}$$

is also an element of Ω . Define $\tilde{\Omega}$ as that subset of Ω such that all control variations $\delta \tilde{w}$ belonging to $\tilde{\Omega}$ Eq. (3.6). $\tilde{\Omega}$ is the set of admissible control variations for this optimal control problem.

Now define the $p \times (k + r)$ partitioned time-varying matrix $\zeta(t)$ by $\zeta(t) = [\gamma | \xi(t)]$. Also define an inner product on Ω by

$$\langle v, w \rangle = \sum_{i=1}^k v_i w_i + \int_{t_0}^{t_f} \left[\sum_{i=k+1}^{k+r} v_i(t) w_i(t) \right] dt. \quad (3.9)$$

If the rows of $\zeta(t)$, which are denoted by the row vectors $\zeta_1, \zeta_2, \dots, \zeta_p$, are linearly independent, they span a p -dimensional subspace, call it Ω^* , of Ω . From Eq. (3.8), the set of admissible control variations $\tilde{\Omega}$ can be defined by the relation

$$\tilde{\Omega} = \{ \delta w \in \Omega : \langle \zeta_i, \delta w \rangle = 0, i = 1, 2, \dots, p \}. \quad (3.10)$$

Analogous to the earlier development, if Ω is a Hilbert space, then Ω is

the direct sum of $\tilde{\Omega}$ and Ω^* , i.e. $\Omega = \tilde{\Omega} \oplus \Omega^*$. Any $\delta\omega$ belonging to Ω is then the sum of a unique $\delta\tilde{\omega}$ belonging to $\tilde{\Omega}$ and a unique $\delta\omega^*$ belonging to Ω^* .

Define the projection operator P as $P[\delta\omega] = \delta\tilde{\omega}$, where according to the previous notation $\delta\tilde{\omega}$ belongs to $\tilde{\Omega}$ and represents the unique projection of $\delta\omega$ onto $\tilde{\Omega}$. The subspace Ω^* can be defined as

$$\begin{aligned}\Omega^* &= \{ \delta\omega \in \Omega : \delta\omega = \sum_{i=1}^p \eta_i \zeta'_i \}, \\ &= \{ \delta\omega \in \Omega : \delta\omega = \zeta' \eta \},\end{aligned}\tag{3.11}$$

where η is the p -vector with constant elements $\eta_1, \eta_2, \dots, \eta_p$.

Using the fact that $\Omega = \tilde{\Omega} \oplus \Omega^*$ and Eqs. (3.10) and (3.11), the following set of equations is generated.

$$\begin{aligned}\langle \zeta'_1, \zeta' \eta \rangle &= \langle \zeta'_1, \delta\omega \rangle \\ \langle \zeta'_2, \zeta' \eta \rangle &= \langle \zeta'_2, \delta\omega \rangle \\ &\vdots \\ \langle \zeta'_p, \zeta' \eta \rangle &= \langle \zeta'_p, \delta\omega \rangle.\end{aligned}$$

The above set of equations can be written more concisely as

$$\left[\gamma\gamma' + \int_{t_0}^{t_f} \xi\xi' dt \right] \eta = \gamma \delta\sigma + \int_{t_0}^{t_f} \xi \delta u dt.$$

Solving for η , one obtains

$$\eta = \left[\gamma\gamma' + \int_{t_0}^{t_f} \xi\xi' dt \right]^{-1} \left[\gamma \delta\sigma + \int_{t_0}^{t_f} \xi \delta u dt \right].$$

Since $\tilde{\delta\omega}$ is equal to $\delta\omega$ minus $\delta\omega^*$, combining the above equation with Eq. (3.11) gives

$$\tilde{\delta\omega} = \delta\omega - \xi' \left[\gamma\gamma' + \int_{t_0}^{t_f} \xi\xi' dt \right]^{-1} \left[\gamma \delta\sigma + \int_{t_0}^{t_f} \xi\delta u dt \right]. \quad (3.12)$$

As a notational convenience, define the product of the two partitioned matrices $[a|b(t)]$ and $[c|d(t)]'$ as

$$[a|b(t)] \otimes \begin{bmatrix} c \\ - \\ d(t) \end{bmatrix} = ac + \int_{t_0}^{t_f} b(t)d(t) dt, \quad (3.13)$$

where to be conformable for this multiplication the number of columns of a must equal the number of rows of c , and the number of columns of $b(t)$ must equal the number of rows of $d(t)$. The matrix product defined by Eq. (3.13) is consistent with letting the i^{th} element in the j^{th} row of the product equal the inner product (3.9) of the i^{th} row of $[a|b(t)]$ and the j^{th} column of $[c|d(t)]'$.

Using the above notation, Eq. (3.12) can be written as

$$\tilde{\delta\omega} = \delta\omega - \xi' \{ [\gamma|\xi(t)] \otimes [\gamma|\xi(t)]' \}^{-1} \{ [\gamma|\xi(t)] \otimes \delta\omega \}.$$

The projection operator defined by the above relation is

$$P[.] = I[.] - \xi' \{ [\gamma|\xi(t)] \otimes [\gamma|\xi(t)]' \}^{-1} \{ [\gamma|\xi(t)] \otimes [.] \}. \quad (3.14)$$

Thus, defining the matrix product (3.13) makes it possible to explicitly define an operator which can be applied to a member of the vector space Ω . Although this is conceptually very pleasing, no real advantage is gained

since Eq. (3.12) provides all the necessary information needed to obtain $\tilde{\delta\omega}$, given $\delta\omega$, and is much more straightforward to apply.

Now if the dynamical system (3.2) is adjoined to the cost functional with the multiplier functions $\lambda(t)$, the variation J due to a variation of the control vector for fixed t_0 and t_f is²³

$$\delta J = [(\phi_x - \lambda')\delta x]_{t=t_f} + [\lambda'\delta x]_{t=t_0} + \int_{t_0}^{t_f} [(H_x + \dot{\lambda}')\delta x + H_u\delta u] dt, \quad (3.15)$$

where the function H (the Hamiltonian) is defined as

$$H[x(t), u(t), \lambda(t), t] = L[x(t), u(t), t] + \lambda'(t)f[x(t), u(t), t].$$

Choosing

$$\dot{\lambda}'(t) = -H_x = -L_x - \lambda'f_x \text{ and } \lambda'(t_f) = \phi_x,$$

and noting that $\delta x_i(t_0) = \delta x_{0_i} = 0$, $i > k$, Eq. (3.15) becomes

$$\delta J = \sum_{i=1}^k (\lambda_i \delta x_i)_{t=t_0} + \int_{t_0}^{t_f} H_u \delta u dt. \quad (3.16)$$

Defining the gradient g as

$$g = \begin{bmatrix} \lambda_1(t_0) \\ \lambda_2(t_0) \\ \vdots \\ \lambda_k(t_0) \\ H_u' \end{bmatrix},$$

Eq. (3.16) can be written as

$$\delta J = \langle g^*, \delta \tilde{\omega} \rangle + \langle \tilde{g}, \delta \tilde{\omega} \rangle = \langle \tilde{g}, \delta \tilde{\omega} \rangle \quad (3.17)$$

since $\delta \omega$ was originally assumed to be an admissible control variation.

For an extremum, δJ must be zero for arbitrary $\delta \tilde{\omega}$; this can happen only if $\tilde{g} = 0$. If θ_i and p_i represent the stepsize and direction of search, respectively, then $\omega_{i+1} = \omega_i + \theta_i p_i$, and it is natural to let $p_i = -\tilde{g}_i$ to insure that δJ is negative for each iteration. This is similar to what was done in Chapter 2 and shall again be referred to as the steepest descent technique of picking the direction of search.

As was stated in Chapter 2, the numerical examples which follow utilize the one-dimensional minimization technique proposed by Rajtóra and Pierson²¹ as an automated means of choosing the stepsize parameters m_i and n_i defined earlier. This algorithm, already documented in Chapter 2, requires only that ω be substituted for u to be applicable to problems treated in this chapter.

In order to simplify the notation, the first k values of the initial conditions of the state variables were assumed to be unspecified. This is not the case with the numerical examples in the next section, but this should not cause any concern to the reader who is aware of the apparent discrepancy.

Numerical Examples

All computations on the two numerical examples which follow were made on an IBM 360/65 computer using double precision arithmetic. All integrations were performed using a standard 4th-order Runge-Kutta program

with the time interval divided into 50 uniform segments. A quadratic polynomial interpolation scheme was used for each one-dimensional minimization.

Forced rod problem

The following problem formulation taken from Icerman²⁶ is a minimum weight structural optimization problem which is nicely adapted to the procedures discussed in this chapter. The structural member of continuously varying cross section whose weight is to be minimized is shown in Fig. 3. It is fixed at one end ($x=0$) and subjected to the axial load $P \cos \omega t$ at the other end ($x=l$). Both the amplitude P and the frequency ω are given, and ω is to be smaller than the fundamental natural frequency of the rod. The axial displacement $W(x,t)$, the downward displacement at time t of the section whose initial coordinate is x , is written as $w(x) \cos \omega t$. The dynamic response to the given load, defined to be $Pw(l)$, is constrained to a specified value.

The minimization problem can be formally stated as:

$$\text{Minimize} \quad \int_0^l A(x) \, dx$$

subject to the second-order differential equation.

$$\frac{d}{dx} \left[EA(x) \frac{d}{dx} w(x) \right] + \omega^2 \rho A(x) w(x) = 0,$$

with the boundary conditions $w(0) = 0$, $EA(l) dw(l)/dx = P$, and $w(l) = w_l =$ constant,

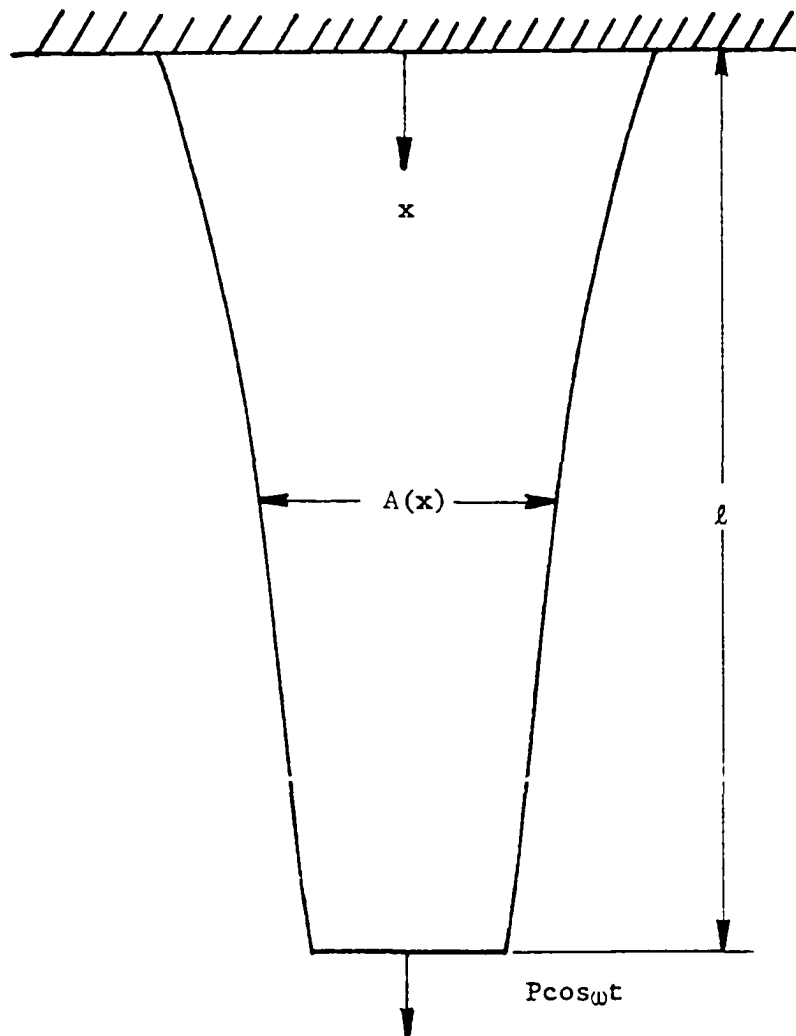


Fig. 3. Rod of continuously varying cross section subjected to harmonic axial loading.

where

E = Youngs modulus,

ρ = constant density of rod,

and $A(x)$ = cross section area.

The third boundary condition is the 'dynamic response' constraint where w_ℓ must be specified.

Analytical results show that the cross section area that gives the least value to the cost functional $\int_0^\ell A(x)dx$ is given by

$$A(x) = \frac{P \cosh(k\ell) \sinh(kx)}{Ek w_\ell \cosh^2(kx)}, \quad k^2 = \omega^2 \rho / E. \quad (3.18)$$

Finally, the minimum volume is

$$\int_0^\ell A(x) dx = \frac{P}{Ek^2 w_\ell} \sinh^2(k\ell). \quad (3.19)$$

It is interesting to note here that the optimal cross section area and minimum volume for a rod in free axial vibration with a point mass fixed to the bottom tip ($x=\ell$) are of the same for as those above.²⁷⁻²⁹ The only difference in the solutions is that the constants in the front of the solutions are a function of w_ℓ and the tip mass rather than E , k , ρ , P , and w_ℓ .

In transforming the problem of the harmonically forced rod to optimal control notation, let

$$t = x,$$

$$u(t) = A(x),$$

$$x_1(t) = w(x),$$

$$\text{and } x_2(t) = A(x) \frac{dw}{dx}.$$

Now, the optimal control problem can be stated as:

$$\text{Minimize } J = \int_0^{\ell} u(t) dt$$

subject to the 2nd-order dynamical system

$$\dot{x}_1 = x_2/u, \quad x_1(0) = x_{1_0} = 0,$$

$$\dot{x}_2 = -k^2 x_1 u, \quad x_2(0) = x_{2_0} \text{ unspecified,}$$

and subject to the terminal state constraints

$$x_1(\ell) = w_{\ell},$$

$$\text{and } x_2(\ell) = P/E.$$

The Hamiltonian is then

$$H = u + \lambda_1 x_2/u - \lambda_2 k^2 x_1 u$$

where the multiplier functions λ_i are determined by

$$\dot{\lambda}_1 = -\partial H/\partial x_1 = k^2 u \lambda_2, \quad \lambda_1(\ell) = 0,$$

$$\text{and } \dot{\lambda}_2 = -\partial H/\partial x_2 = -\lambda_1/u, \quad \lambda_2(\ell) = 0.$$

The simple result obtained for H_u , due to λ_1 and λ_2 equaling zero on the entire time interval $0 \leq t \leq \ell$, is

$$H_u = 1, \quad 0 \leq t \leq \ell.$$

In addition, $\lambda_2(0)$, which is the other element of the gradient, is equal to zero for the same reason.

The problem being discussed here was solved using three different initial controls and the following values for the problem parameters for a standard aluminum bar:

$$\ell = 100 \text{ inches,}$$

$$w_\ell = 1 \text{ inch,}$$

$$P = 200 \text{ pounds,}$$

$$E = 10.3 \times 10^3 \text{ pounds/inch}^2,$$

$$\omega = 10 \text{ radians/second,}$$

$$\text{and } \rho = 0.1/32.2 \text{ slugs/inch}^3.$$

The minimum volume determined from Eq. (3.19) using these parameters is 214.4918820 cubic inches. The optimal control or cross section area is plotted in Fig. 4 for the previously specified parameters.

The three initial controls used the initial guesses for $x_2(0)$ of 0.019418, 0.038835, and 0.009709 inches per inch and uniform cross section areas of 1.9418, 3.8835, and 0.9709 square inches, respectively. The first initial control was based on the uniform cross section rod which would give a one inch deflection for static loading of 200 pounds. The second and third initial controls are simply multiples of two times the first initial control and one-half times the first initial control. The value of ϵ used in determining constraint satisfaction was 10^{-5} .

The cost function for the initial controls just given obtained a value below 214.492 cubic inches in 2, 4 and 3 steps, respectively, with total execution times of 5.09, 13.67, and 6.37 seconds. These will be known as solution one, solution two and solution three, respectively. The convergence history obtained by using the proposed steepest descent algorithm with the second initial guess of the control is given in Table 1. The

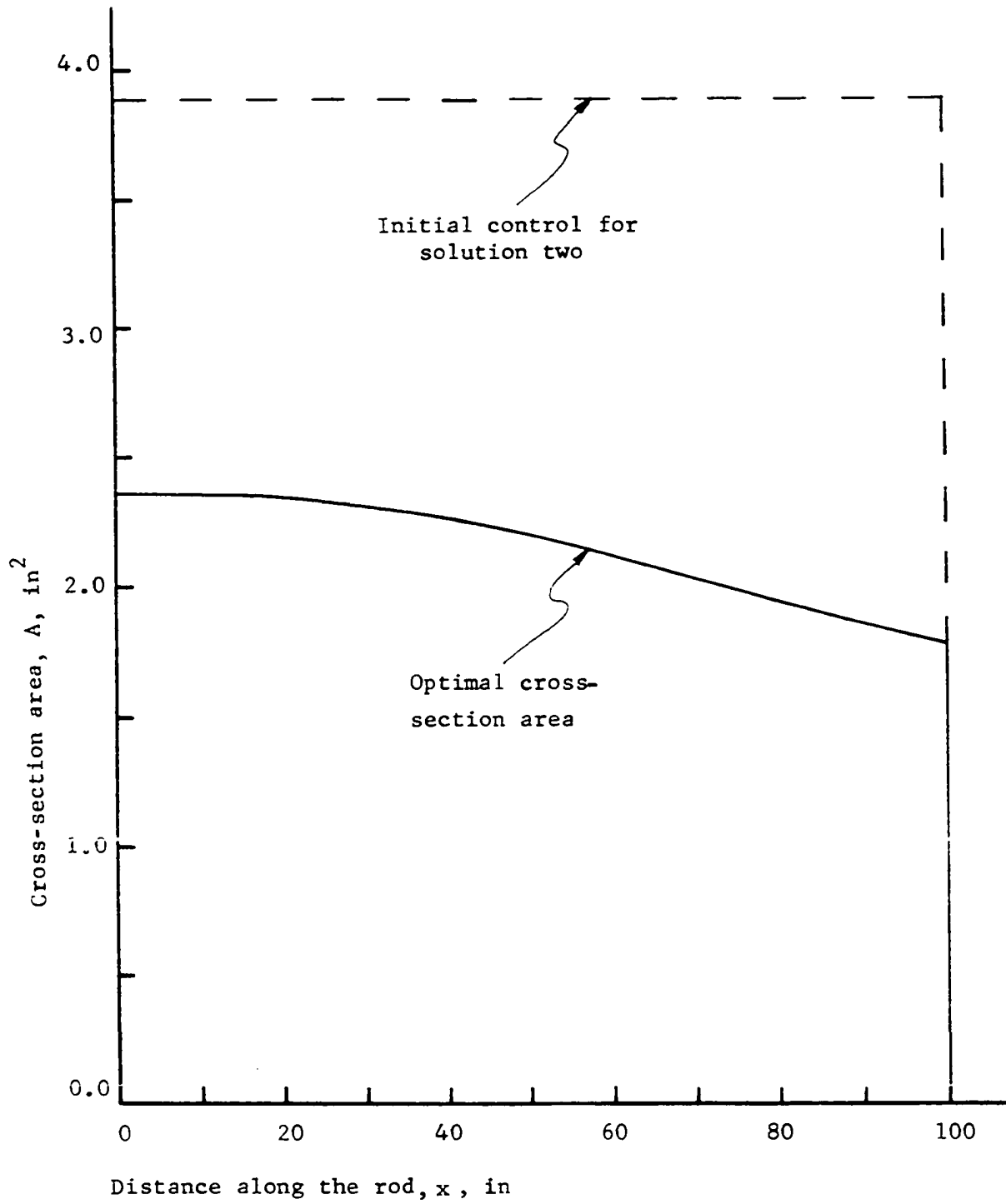


Fig. 4. Optimum cross-section area.

Table 1. Convergence history for solution two of the forced rod problem.

Step number	J_3 in.	$\langle \tilde{g}, \tilde{g} \rangle$	$x_2(0)$ in./in.	$u(0)_2$ in.	$u(40)$ in. ²	$u(60)$ in. ²	$u(100)$ in. ²
0	388.350	3.9744	.038835	3.8835	3.8835	3.8835	3.8835
0_{\downarrow}	234.412	26.1992	.023798	1.7928	2.1117	2.4394	3.2180
1	215.580	1.8739	.022233	2.6818	2.1996	1.9746	1.7898
2	214.517	0.0516	.022417	2.3979	2.2151	2.1268	1.7309
3	214.493	0.0025	.022419	2.3652	2.2428	2.1179	1.7839
4	214.492	0.0001	.022418	2.3593	2.2455	2.1181	1.7646
Optimal ³⁰	214.492	0	.022419	2.3563	2.2462	2.1179	1.7676

step number 0_ψ refers to the control $[u_0(t)]_ψ$ where the notation $u_ψ$ has previously been defined in Chapter 2. If a function evaluation is defined as one forward and one backward integration, analogous to blocks 1 and 2 of Fig. 2, this quantity can be used as a measure to determine the computational effort required by an algorithm. Table 2 gives the number of function evaluations per step (NFEVAL) and the average number of function evaluations at each step needed to correct back to the constraint (ANFECC) for each solution.

Variable launch site rocket launch problem

The rocket launch problem considered here is very similar to that treated by Lasdon, Mitter, and Waren⁸ and again mentioned in connection with the one-dimensional minimization algorithm proposed by Rajtora and Pierson.²¹ The objective of that problem was to maximize the horizontal velocity of a rocket under the assumptions of a constant gravitation acceleration of 32 ft./sec.², two-dimensional vacuum flight, and a constant thrust acceleration of twice that of the gravitational acceleration. The control variable was the thrust direction with respect to the horizontal. The thrusting time was specified at 100 seconds. The terminal state constraints required a zero vertical speed at an altitude of 100,000 feet. In addition to these requirements, the present problem requires that the down range flight distance be fixed, and the initial launch site is no longer specified. Application of the necessary conditions to these two different problems shows that the optimal thrust direction time histories for them are identical. Thus, this is a problem contrived for numerical testing purposes only.

Table 2. Computational effort required in solving the forced rod problem.

Step Number	Solution 1		Solution 2		Solution 3	
	NFEVAL	ANFECC	NFEVAL	ANFECC	NFEVAL	ANFECC
0	1		1		1	
0_{Ψ}	3	3	13	13	3	3
1	9	2	28	4.6	7	1 1/3
2	6	1	12	2	6	1
3			6	1	6	1
4			6	1		

In nondimensionalized form the optimal control problem is stated as:

$$\text{Minimize } J = -x_3(1)$$

subject to the 4th-order dynamical system

$$\dot{x}_1 = x_2, \quad x_1(0) = 0,$$

$$\dot{x}_2 = 6.4 \sin u - 3.2, \quad x_2(0) = 0,$$

$$\dot{x}_3 = 6.4 \cos u, \quad x_3(0) = 0,$$

$$\dot{x}_4 = x_3, \quad x_4(0) \text{ unspecified,}$$

and subject to the terminal state constraints

$$x_1(1) = 1,$$

$$x_2(1) = 0,$$

$$x_4(1) = 1.$$

The initial thrust angle was chosen to be $u_0(t) = (\pi/2)(1-t)$ and the initial guess for $x_4(0)$, the launch site position, was chosen to be zero. The execution time for eight steps of the proposed algorithm was 18.0 seconds. The convergence history of these eight steps is given in Table 3. The value of ϵ used in determining constraint satisfaction was 10^{-5} .

Table 3. Convergence history of the rocket launch problem

Step number	J	$\langle \tilde{g}, \tilde{g} \rangle$	$x_4(0)$	$u(0)$	$u(.4)$	$u(.6)$	$u(1)$	NFEVAL	ANFECC
0	-4.07436	0.75520	0	$\pi/2$.3 π	.2 π	0.0	1	
0 _{ψ}	-3.28830	1.19097	-.018714	1.5946	1.1996	0.6299	-.9158	4	4
1	-3.46257	1.25791	-.311148	1.2692	1.2437	0.8874	-.7114	17	3.25
2	-3.50337	0.12421	-.307244	1.4287	1.1475	0.8384	-.9132	25	2.43
3	-3.50672	0.05403	-.314304	1.3477	1.1635	0.8188	-.9357	7	1.33
4	-3.50749	0.01121	-.317166	1.3909	1.1653	0.8134	-.9539	6	1
5	-3.50780	0.00529	-.320064	1.3581	1.1652	0.8102	-.9618	7	1.33
6	-3.50795	0.00253	-.321737	1.3798	1.1647	0.8095	-.9672	6	1
7	-3.50802	0.00132	-.323207	1.3623	1.1643	0.8091	-.9698	7	1.33
8	-3.50805	0.00068	-.324109	1.3742	1.1640	0.8093	-.9719	6	1
30 optimal	-3.50809	0	-.32684	1.3670	1.1627	0.8097	-.9756		

CHAPTER 4. OPTIMAL BRANCHED TRAJECTORY PROBLEMS

Problem Development

The trajectory of several vehicles traveling together for a period of time and then separating in order to proceed individually to separate terminal states is shown schematically in Fig. 5. The simplified trajectory represented in Fig. 5 having q branches is typical of a much larger class known as branched trajectories. They differ from conventional trajectories in that the state and control dimensions vary with time. This investigation includes only those branched trajectories which have a single branch point (point B in Fig. 5), although generalization to additional branch points is straightforward and represents one of the strengths of the projection operator development presented here. State variables are generally continuous at the branch point, but the control variables are generally not. The ψ^j , $j = 1, 2, \dots, q$, in Fig. 5 represent the terminal state constraint functions of the q branches.

Mason³¹ converts the branched trajectory optimization problem to a conventional Bolza problem using several linear time transformations. This allows him to apply the well-established necessary conditions of optimal control theory directly to the branched trajectory problem. Mason discusses the applicability of the single branch point branched trajectory to multiple payload launch trajectories, launch trajectories based on abort capabilities, maneuvers of lunar lander/orbiter vehicles, and cooperative multiple aircraft maneuvers. He has obtained numerical solutions via indirect optimization techniques for the first two of these categories.

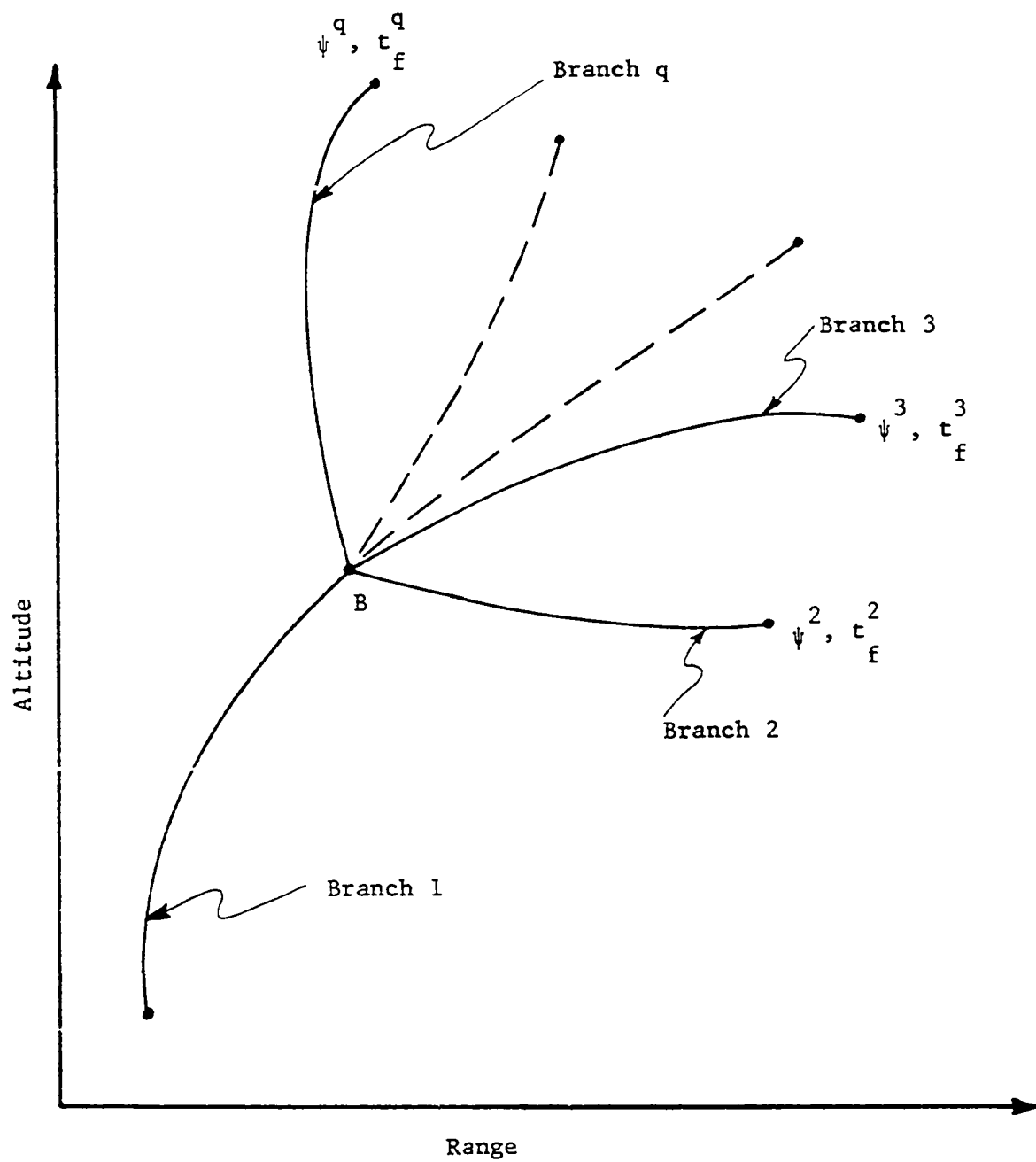


Fig. 5. Schematic representation of a branched trajectory with a single branch point,

Additional numerical experience on abort capability problems is given by Mason, Smith, and Dickerson.³²

Rozendaal¹⁸ and Gera¹⁹ have treated optimal branched trajectory problems using steepest descent in conjunction with a projection operator. Both men derive and apply the projection operator similarly to the derivation and application of the projection operator due to Bryson and Denham.⁶ Gera¹⁹ gives numerical results for a rather simple multiple (two) payload launch assuming rocket flight in a vacuum and specified branch and terminal times. The system dynamics for each branch are then similar to those of the second example problem of Chapter 3. The objective is to maximize the sum of the altitudes for the upper two stages while keeping the kinetic energy per unit mass of the upper two stages fixed at the final time. Gera obtains good numerical results when compared with the analytical solution.

Rozendaal¹⁸ treats the more complicated problem of designing optimal space shuttle vehicle ascent trajectories. He converts the branched trajectory problem into an unbranched trajectory problem whose total duration in time is equal to the sum of the individual branch times. This formulation will here be denoted as the 'series formulation' of the optimal branched trajectory problem. Rozendaal states that this technique avoids the increases in computational complexity and computer storage requirements associated with methods which map all branches into a common interval of the independent variable as Gera¹⁹ and Mason³¹ suggest. The latter formulation will be denoted here as the 'parallel formulation' of the optimal branched trajectory problem. Rozendaal reports a 10 percent payload gain over an ascent trajectory designed by more conventional means.

The optimal branched trajectory problem defined below is more general than that treated by Gera,¹⁹ although it is not nearly as general as that described by Rozendaal¹⁸ or Mason.³¹ It is adequate, however, to treat the numerical example presented later in this chapter and permits a relatively compact derivation of the necessary conditions here. It is assumed that there are no inequality constraints present, that there are only three branches, that all state variables are continuous across the single branch point, and that there is no terminal state constraint on branch one. Actually, in the formulation used by Mason,³¹ the specification that the state variables be continuous at the branch point is a terminal state constraint. The algorithm presented later, however, connects the branches in a series form as done by Rozendaal¹⁸ rather the parallel approach used by Gera¹⁹ so that this constraint is satisfied automatically by the numerical integration.

Consider the problem of determining the three r-vector control functions $u^j(t)$, $t_o^j \leq t \leq t_f^j$, $j = 1, 2, 3$, which minimize the cost functional

$$J = \phi[x^1(t_f^1), x^2(t_f^2), x^3(t_f^3)] + \sum_{j=1}^3 \int_{t_o^j}^{t_f^j} L^j(x^j, u^j, t) dt, \quad (4.1)$$

subject to the three n^{th} -order nonlinear dynamical systems

$$\dot{x}^j = f^j(x^j, u^j, t), \quad x^j(t_o^j) = x_o^j, \quad j = 1, 2, 3, ** \quad (4.2)$$

with x_o^1 and t_o^1 given $t_f^1 = t_o^2 = t_o^3$, t_f^1 , t_f^2 , and t_f^3 unspecified, and continuity of all state variables at the branch point specified. In addition, there exist p^j ($p^j \leq n$) terminal state constraints

**Superscripts on the time variable and on subscripts are left off wherever the usage makes their value clearly understood.

$$\psi^j[x^j(t_f)] = 0, \quad j = 2, 3. \quad (4.3)$$

Mason³¹ and Gera¹⁹ recommend the linear time transformation

$$\left. \begin{aligned} t^j &= t_o^j + (t_f^j - t_o^j)\tau^j, \\ &= t_o^j + \alpha^j \tau^j, \end{aligned} \right\} \begin{aligned} 0 \leq \tau \leq 1, \quad t_o^j \leq t^j \leq t_f^j, \\ j = 1, 2, 3. \end{aligned} \quad (4.4)$$

An alternate time transformation and the one used in this investigation is

$$\left. \begin{aligned} t^j &= t_o^j + [(t_f^j - t_o^j)/\tau_f^j]\tau^j, \\ &= t_o^j + \alpha^j \tau^j, \end{aligned} \right\} \begin{aligned} 0 \leq \tau \leq \tau_f^j, \quad t_o^j \leq t^j \leq t_f^j, \\ j = 1, 2, 3. \end{aligned} \quad (4.5)$$

where τ_f^j is a fixed value approximately equal to $t_f^j - t_o^j$ so that each α^j is approximately equal to one. The linear time transformation (4.4) enables the user to do two main things: a) compress or expand the time domain of each branch into a common interval of the independent variable time, and b) handle varying time intervals for the individual branches. The linear time transformation (4.5) also provides a means of treating varying time intervals but does not arbitrarily compress or expand the time interval which might detract from the physical reality of the problem. This investigation treats the branches in a series manner as done by Rozendaal¹⁸ so that no compression or elongation of the time intervals is necessary. Thus, time transformation (4.5) will be used. The parameter α is set equal to one for a branch with a fixed time interval. One additional change from the formulation used by Mason³¹ or Gera¹⁹ is that each

α^j is treated as a constant control function, i.e. a control parameter, as opposed to treating it as a constant state variable. The later approach requires the incorporation of additional differential equations.

Due to the linear time transformation (4.5), consider now the problem of determining the three r-vector control functions $u^j(\tau)$, $0 \leq \tau \leq \tau_f$, $j = 1, 2, 3$, and the three scalar control parameters α^j , $j = 1, 2, 3$, which minimize the cost functional

$$J = \phi[x^1(\tau_f), x^2(\tau_f), x^3(\tau_f)] + \sum_{j=1}^3 \int_0^{\tau_f} \ell^j(x^j, u^j, \tau) d\tau \quad (4.6)$$

subject to the three n^{th} -order nonlinear dynamical systems

$$\dot{x}^j(\tau) = F^j(x^j, u^j, \alpha^j, \tau), \quad x^j(0) = x_o^j, \quad j = 1, 2, 3, \quad (4.7)$$

with x_o^1 given, τ_f^1 , τ_f^2 , and τ_f^3 unspecified, $F^j(x^j, u^j, \alpha^j, \tau)$ equaling $\alpha^j F^j(x^j, u^j, \tau_o^j + \alpha^j \tau)$, $\ell^j(x^j, u^j, \alpha^j, \tau)$ equaling $\alpha^j \ell^j(x^j, u^j, \tau_o^j + \alpha^j \tau)$, and continuity of all state variables specified at the branch point. In addition, there exist p^j ($p^j \leq n$) terminal state constraints

$$\psi^j[x^j(\tau_f)] = 0, \quad j = 2, 3. \quad (4.8)$$

For a completely rigorous development the projection operator which handles the terminal state constraints (4.8) should be derived using a linear dynamical system and linear terminal state constraints. It could then be extended to a nonlinear dynamical system and nonlinear terminal state constraints as was done in Chapter 2. The projection operator developed here, however, as in Chapter 3, is derived directly for the nonlinear case. This should cause no confusion provided everyone is aware that the

projection operator so developed is only valid in the region for which the linearization of the nonlinear system dynamics and nonlinear terminal state constraints is valid.

Linearization of the system dynamics (4.7) about a nominal $u^j(\tau)$, $x^j(\tau)$, and α^j , where α^j is treated as a constant control function, yields

$$\begin{aligned} \delta \dot{x}^j = & F_x^j(x^j, u^j, \alpha^j, \tau) \delta x^j(\tau) + F_u^j(x^j, u^j, \alpha^j, \tau) \delta u^j(\tau) \\ & + F_\alpha^j(x^j, u^j, \alpha^j, \tau) \delta \alpha^j, \quad j = 1, 2, 3. \end{aligned} \quad (4.9)$$

Since the initial state variables for branch one are fixed, the solution to Eq. (4.9) for branch one is

$$\delta x^1(\tau) = \int_0^\tau \Phi^1(\tau, t) F_u^1(t) \delta u^1(t) dt + \int_0^\tau \Phi^1(\tau, t) F_\alpha^1(t) dt \delta \alpha^1 \quad (4.10)$$

with all the notation having previously been defined in Chapter 2 except for the superscripts which refer to the branch number. An equation similar to Eq. (4.10) can be written for either branch two or branch three once the variation due to the change in initial conditions is provided in the equation.

$$\begin{aligned} \delta x^j(\tau) = & \Phi^j(\tau, 0) \delta x^j(0) + \int_0^\tau \Phi^j(\tau, t) F_u^j(t) \delta u^j(t) dt \\ & + \int_0^\tau \Phi^j(\tau, t) F_\alpha^j(t) dt \delta \alpha^j, \quad j = 2, 3. \end{aligned} \quad (4.11)$$

Since the state variables are continuous at the branch point, substitution of Eq. (4.10) into Eq. (4.11) gives

$$\begin{aligned} \delta x^j(\tau) = & \bar{\Phi}^j(\tau, 0) \int_0^{\tau_f} \bar{\Phi}^1(\tau_f, t) F_u^1(t) \delta u^1(t) dt + \bar{\Phi}^j(\tau, 0) \int_0^{\tau_f} \bar{\Phi}^1(\tau_f, t) F_\alpha^1(t) dt \delta \alpha^1 \\ & + \int_0^{\tau} \bar{\Phi}^j(\tau, t) F_u^j(t) \delta u^j(t) dt + \int_0^{\tau} \bar{\Phi}^j(\tau, t) F_\alpha^j(t) dt \delta \alpha^j, j = 2, 3. \end{aligned} \quad (4.12)$$

Now, if each element of the vector $\psi^j[x^j]$, $j = 2, 3$, is differentiable and if each iteration in the process satisfies Eq. (4.8), then near a nominal $x^j(\tau_f)$,

$$\psi_x^j[x^j(\tau_f)] \delta x^j(\tau_f) \approx \psi^j[x^j(\tau_f) + \delta x^j(\tau_f)] - \psi^j[x^j(\tau_f)] = 0, j=2,3. \quad (4.13)$$

Combining Eq. (4.12) with Eq. (4.13) yields

$$\begin{aligned} & \int_0^{\tau_f} \psi_x^j \bar{\Phi}^j(\tau_f, 0) \bar{\Phi}^1(\tau_f, t) F_u^1(t) \delta u^1(t) dt + \int_0^{\tau_f} \psi_x^j \bar{\Phi}^j(\tau_f, 0) \bar{\Phi}^1(\tau_f, t) F_\alpha^1(t) dt \delta \alpha^1 \\ & + \int_0^{\tau_f} \psi_x^j \bar{\Phi}^j(\tau_f, t) F_u^j(t) \delta u^j(t) dt + \int_0^{\tau_f} \psi_x^j \bar{\Phi}^j(\tau_f, t) F_\alpha^j(t) dt \delta \alpha^j = 0, j=2,3. \end{aligned} \quad (4.14)$$

For convenience, define the $(p^2 + p^3) \times r$ time-varying matrices $\bar{\xi}^1(t)$, $\bar{\xi}^2(t)$, and $\bar{\xi}^3(t)$, and the $(p^2 + p^3) \times 3$ constant matrix γ as

$$\bar{\xi}^1(t) = \begin{bmatrix} \psi_x^2 \bar{\Phi}^2(\tau_f, 0) \bar{\Phi}^1(\tau_f, t) F_u^1(t) \\ - - - - - \\ \psi_x^3 \bar{\Phi}^3(\tau_f, 0) \bar{\Phi}^1(\tau_f, t) F_u^1(t) \end{bmatrix},$$

$$\bar{\xi}^2(t) = \begin{bmatrix} \psi_x^2 \bar{\Phi}^2(\tau_f, t) F_u^2(t) \\ - - - - - \\ 0 \end{bmatrix},$$

$$\bar{\xi}^3(t) = \begin{bmatrix} 0 \\ - - - - - \\ \psi_x^3 \bar{\Phi}^3(\tau_f, t) F_u^3(t) \end{bmatrix},$$

$$\text{and } \gamma = \begin{bmatrix} \int_0^{\tau_f} \psi_x^2 \Phi^2(\tau_f, 0) \Phi^1(\tau_f, t) F_\alpha^1(t) dt & \cdot & \int_0^{\tau_f} \psi_x^2 \Phi^2(\tau_f, t) F_\alpha^2(t) dt & \cdot & 0 \\ \hline \int_0^{\tau_f} \psi_x^3 \Phi^3(\tau_f, 0) \Phi^1(\tau_f, t) F_\alpha^1(t) dt & \cdot & 0 & \cdot & \int_0^{\tau_f} \psi_x^3 \Phi^3(\tau_f, t) F_\alpha^3(t) dt \end{bmatrix},$$

if all the time intervals are unspecified as in the original problem statement. Should any of the time intervals be specified, say time interval j for branch j , then α^j is permanently set equal to 1, and the matrix γ is simplified by the deletion of column j . In addition, let the partitioned $(p^2 + p^3) \times (3r + 3)$ time-varying matrix $\zeta(t)$ be defined by

$$\zeta(t) = [\bar{\xi}^1(t) | \bar{\xi}^2(t) | \bar{\xi}^3(t) | \gamma].$$

For a more general development involving q branches the dimensionality of $\zeta(t)$ would be $(\sum_{j=2}^q p^j) \times (qr + q)$. Using the first four definitions above, Eq. (4.14) becomes

$$\sum_{j=1}^3 \int_0^{\tau_f} \bar{\xi}^j(t) \delta u^j(t) dt + \gamma \delta \alpha = 0 \quad (4.15)$$

where α is a q -vector of elements α^j , $j=1, 2, \dots, q$, q being equal to three in this case.

Let Ω be a set of $(3r + 3)$ -vector controls defined as the Cartesian product of U^1 , U^2 , U^3 , and Λ , i.e. $U = U^1 \times U^2 \times U^3 \times \Lambda$. U^1 , U^2 , and U^3 are the r -vector control functions of branch one, branch two, and branch three, respectively. The three elements of Λ , α^1 , α^2 , and α^3 , are the time interval multipliers defined by Eq. (4.5) of branch one, branch two, and branch three, respectively. A vector ω of Ω can be written as

$$\begin{aligned}
 \omega &= \begin{bmatrix} \omega_1 \\ \omega_2 \\ \cdot \\ \cdot \\ \cdot \\ \omega_{3r+3} \end{bmatrix} = \begin{bmatrix} u_1^1 \\ \cdot \\ \cdot \\ \cdot \\ u_r^1 \\ u_1^2 \\ \cdot \\ \cdot \\ \cdot \\ u_r^2 \\ u_1^3 \\ \cdot \\ \cdot \\ \cdot \\ u_r^3 \\ \cdot \\ \cdot \\ \cdot \\ \alpha^1 \\ \alpha^2 \\ \alpha^3 \end{bmatrix} = \begin{bmatrix} u^1 \\ u^2 \\ u^3 \\ \alpha \end{bmatrix}
 \end{aligned}$$

The variation of a vector ω belonging to Ω , defined in the natural way as

$$\delta\omega = \begin{bmatrix} \delta\omega_1 \\ \delta\omega_2 \\ \vdots \\ \delta\omega_{3r+3} \end{bmatrix} = \begin{bmatrix} \delta u_1^1 \\ \vdots \\ \delta u_r^1 \\ \delta u_1^2 \\ \vdots \\ \delta u_r^2 \\ \delta u_1^3 \\ \vdots \\ \delta u_r^3 \\ \delta\alpha^1 \\ \delta\alpha^2 \\ \delta\alpha^3 \end{bmatrix} = \begin{bmatrix} \delta u^1 \\ \delta u^2 \\ \delta u^3 \\ \delta\alpha \end{bmatrix},$$

is also an element of Ω .

Now, if the rows of $\zeta(t)$, which are denoted by the row vectors, $\zeta_1, \zeta_2, \dots, \zeta_p$ where $p = p^2 + p^3$, are linearly independent, they span a p -dimensional subspace, call it Ω^* , of Ω . If an inner product is defined on

$$\Omega \text{ as } \langle v, w \rangle = \int_0^T \sum_{i=1}^r v_i w_i dt + \int_0^T \sum_{i=r+1}^{2r} v_i w_i dt + \int_0^T \sum_{i=3r+1}^{3r} v_i w_i dt + \sum_{i=3r+1}^{3r+3} v_i w_i, \quad (4.16)$$

then the set of admissible control variations $\tilde{\Omega}$ can be defined from Eq.

(4.15) by the relation

$$\tilde{\Omega} = \{ \delta\omega \in \Omega : \langle \zeta_i, \delta\omega \rangle = 0, i = 1, 2, \dots, p \}. \quad (4.17)$$

Once again, if Ω is restricted to being a Hilbert space, Ω is the direct sum of $\tilde{\Omega}$ and Ω^* , or $\Omega = \tilde{\Omega} \oplus \Omega^*$.

Define the projection operator P as $P[\delta\omega] = \tilde{\delta\omega}$. The subspace Ω^* can be defined as

$$\begin{aligned}\Omega^* &= \{\delta\omega \in \Omega: \delta\omega = \sum_{i=1}^p \eta_i \xi_i\} \\ &= \{\delta\omega \in \Omega: \delta\omega = \xi \eta\}\end{aligned}$$

where η is a p -vector with constant elements $\eta_1, \eta_2, \dots, \eta_p$. Since $\Omega = \tilde{\Omega} \oplus \Omega^*$, $\tilde{\delta\omega}$ may now be written as

$$\tilde{\delta\omega} = \delta\omega - \xi \eta, \quad (4.18)$$

so when utilizing Eq. (4.17), the following set of equations is generated.

$$\begin{aligned}\langle \xi_1, \xi \eta \rangle &= \langle \xi_1, \delta\omega \rangle \\ \langle \xi_2, \xi \eta \rangle &= \langle \xi_2, \delta\omega \rangle \\ &\vdots \\ \langle \xi_p, \xi \eta \rangle &= \langle \xi_p, \delta\omega \rangle\end{aligned}$$

These equations can be rewritten as

$$\begin{aligned}\left[\int_0^T \xi^1(t) \xi^{1'}(t) dt + \int_0^T \xi^2(t) \xi^{2'}(t) dt + \int_0^T \xi^3(t) \xi^{3'}(t) dt + \gamma' \right] \eta = \\ \int_0^T \xi^1(t) \delta u^1(t) dt + \int_0^T \xi^2(t) \delta u^2(t) dt + \int_0^T \xi^3(t) \delta u^3(t) dt + \gamma \delta \alpha. \quad (4.19)\end{aligned}$$

It is now possible to solve for $\bar{\eta}$ from Eq. (4.19) which when combined with Eq. (4.18) gives the desired result. From the appearance of Eqs. (4.18) and (4.19) it is not real easy to interpret the form of the projection operator P . This situation could be handled as in Chapter 3 by defining a type of matrix multiplication on Ω , but this would prove to be of little practical advantage. In the actual application of the projection technique, Eq. (4.18) is used in conjunction with Eq. (4.19) without the actual projection operator being explicitly given.

Extending Eq. (3.15) to optimal branched trajectory problems, the variation in J due to a variation of the three r -vector control functions $u^1(t)$, $u^2(t)$, $u^3(t)$, and the three control parameters α^1 , α^2 , and α^3 , that is, a variation of ω , is:

$$\begin{aligned} \delta J = & [(\delta x^1 - \lambda^1) \delta x^1]_{\tau=\tau_f} + [\lambda^1 \delta x^1]_{\tau=0} + \int_0^{\tau_f} [(H_x^1 + \lambda^1) \delta x^1 + H_u^1 \delta u^1 + H_\alpha^1 \delta \alpha^1] dt \\ & + [(\delta x^2 - \lambda^2) \delta x^2]_{\tau=\tau_f} + [\lambda^2 \delta x^2]_{\tau=0} + \int_0^{\tau_f} [(H_x^2 + \lambda^2) \delta x^2 + H_u^2 \delta u^2 + H_\alpha^2 \delta \alpha^2] dt \\ & + [(\delta x^3 - \lambda^3) \delta x^3]_{\tau=\tau_f} + [\lambda^3 \delta x^3]_{\tau=0} + \int_0^{\tau_f} [(H_x^3 + \lambda^3) \delta x^3 + H_u^3 \delta u^3 + H_\alpha^3 \delta \alpha^3] dt, \end{aligned}$$

where the function H^j (the Hamiltonian) is defined as before along each of the three branches as

$$H^j[x^j, u^j, \alpha^j, \lambda^j, \tau] = \ell^j[x^j, u^j, \alpha^j, \tau] + \lambda^j F^j[x^j, u^j, \alpha^j, \tau], \quad j = 1, 2, 3.$$

Since the initial state vector for the first branch is given and the state variables are continuous at the branch point, the expression for the variation of the cost functional becomes

$$\begin{aligned}
\delta J = & [\phi_{x1} - \lambda^1(\tau_f) + \lambda^2(0) + \lambda^3(0)] \delta x^1(\tau_f) + [(\phi_{x2} - \lambda^2) \delta x^2]_{\tau=\tau_f} \\
& + [(\phi_{x3} - \lambda^3) \delta x^3]_{\tau=\tau_f} + \int_0^{\tau_f} [(H_x^1 + \dot{\lambda}^1) \delta x^1 + H_u^1 \delta u^1 + H_\alpha^1 \delta \alpha^1] dt \\
& \int_0^{\tau_f} [(H_x^2 + \dot{\lambda}^2) \delta x^2 + H_u^2 \delta u^2 + H_\alpha^2 \delta \alpha^2] dt + \int_0^{\tau_f} [(H_x^3 + \dot{\lambda}^3) \delta x^3 + H_u^3 \delta u^3 + H_\alpha^3 \delta \alpha^3] dt.
\end{aligned} \tag{4.20}$$

Choosing

$$\dot{\lambda}^j(\tau) = -H_x^j = -\ell_x^j - \lambda^j F_x^j, \quad j = 1, 2, 3,$$

$$\lambda^j(\tau_f) = \phi_{xj}, \quad j = 2, 3,$$

and $\lambda^1(\tau_f) = \phi_{x1} + \lambda^2(0) + \lambda^3(0),$

simplifies Eq. (4.20) to

$$\delta J = \sum_{j=1}^3 \left(\int_0^{\tau_f} H_u^j \delta u^j + H_\alpha^j \delta \alpha^j dt \right),$$

or

$$\delta J = \sum_{j=1}^3 \left(\int_0^{\tau_f} H_u^j \delta u^j dt \right) + \sum_{j=1}^3 \left(\int_0^{\tau_f} H_\alpha^j dt \delta \alpha^j \right). \tag{4.21}$$

Now, if the gradient is defined to be

$$g = \begin{bmatrix} H_u^1, \\ H_u^2, \\ H_u^3, \\ \int H_\alpha^1 dt \\ \int H_\alpha^2 dt \\ \int H_\alpha^3 dt \end{bmatrix}$$

with the obvious limits on the integrals, Eq. (4.21) can be written in the very simple form of

$$\delta J = \langle g, \delta \omega \rangle .$$

Since g can be written as $\tilde{g} + g^*$, where \tilde{g} belongs to $\tilde{\Omega}$, and g^* belongs to Ω^* , the expression of δJ becomes

$$\delta J = \langle \tilde{g} + g^*, \delta \omega \rangle = \langle \tilde{g}, \delta \tilde{\omega} \rangle , \quad (4.22)$$

since $\delta \omega$ must belong to $\tilde{\Omega}$ to be an admissible control variation.

For an extremum, δJ must be zero for arbitrary $\delta \tilde{\omega}$, which can only happen if $\tilde{g} = 0$. Again, the easiest way to insure that δJ is negative for each iteration is to let the direction of search p_1 be the negative projected gradient, $-\tilde{g}$. This is the steepest descent technique of picking the direction of search.

Command Service Module Abort Problem

The example problem treated here is the same problem treated by Mason³¹ and Mason, Smith, and Dickerson.³² They stress the importance of branched trajectory optimization to secondary mission or abort capability trajectory design. The basic fundamentals of solving optimal branched trajectory problems are provided in Ref. 31, while the concept of complete mission planning including a secondary or abort mission in the event the primary mission cannot be completed is emphasized in Ref. 32. When a launch vehicle has performance capability in excess of that required for

the primary mission, the excess propellant may be used to shape the primary mission trajectory so the performance of a secondary mission objective is improved.

The objective of the example problem is to maximize the abort capability of the Command Service Module (CSM) on an Apollo mission should an abort be required at the staging of the Saturn S-II and Saturn S-IVB. This point was picked as the most critical point of the trajectory due to problems which might occur during separation or during the ignition sequence of the S-IVB. Referring to Fig. 5, branch 1 is the trajectory of the of the trajectory of the S-II vehicle carrying the S-IVB, the Lunar Excursion Module (LEM), and the CSM. Branch 2 is the trajectory of the S-IVB carrying the LEM and the CSM. Branch 3 is the abort trajectory of the CSM. The loaded S-IVB and the LEM are dropped prior to the abort trajectory, branch 3. Mason, Smith, and Dickerson³² consider two distinct measures of performance for branch 3. First, a specific orbit is chosen for the secondary mission, and the final mass in this orbit is maximized. This effectively maximizes the propellant remaining when the vehicle achieves the orbit which can then be used for further maneuvering. Alternately, the final mass and the circular orbit conditions are specified for the secondary mission and the altitude of this orbit is maximized. Only the first case is considered in the present investigation.

The minimization problem can be stated as:

$$\text{Minimize} \quad J = -m(t_f^3)$$

subject to the 3rd-order dynamical system on each branch

$$\dot{h} = v \sin \gamma$$

$$\dot{v} = \frac{T}{m} \cos \theta - \frac{\mu}{(r_e + h)^2} \sin \gamma$$

$$\dot{\gamma} = \frac{T}{mv} \sin \theta - \left(\frac{\mu}{(r_e + h)^2 v} - \frac{v}{(r_e + h)} \right) \cos \gamma$$

and subject to the terminal state constraints

$$h^j(\tau_f) = h_{\tau_f}^j, \quad j = 2, 3,$$

$$v^j(\tau_f) = v_{\tau_f}^j, \quad j = 2, 3,$$

$$\gamma^j(\tau_f) = \gamma_{\tau_f}^j, \quad j = 2, 3,$$

with values for $h_{\tau_f}^j$, $v_{\tau_f}^j$, and $\gamma_{\tau_f}^j$ specified in Table 4 and where the

vehicle mass is assumed to be linearly time-varying:

$$m = m_0 - \beta(t - t_0).$$

The state variables v , γ , and h are, respectively the vehicle speed, the flight path angle referenced to the local horizontal, and the height above the Earth's surface (altitude). The control variable θ is the angle between the velocity and thrust vectors. The remaining parameters T , β , r_e , m_0 , and μ are the thrust, constant mass flow rate, radius of the Earth, initial mass, and gravitational parameter. Necessary values for these parameters and values for state variables at the initial time are given in Table 4.

All computations performed on the numerical example just described were made on an IBM 360/65 computer using double precision arithmetic.

Table 4. Command Service Module abort problem

Initial states				
	Velocity, v <u>km/sec</u>	Gamma, γ <u>deg</u>	Altitude, h <u>km</u>	
	2.848127	14.91	87.771	
Stage Data				
Branch number	Thrust, T <u>nt</u>	Beta, β <u>kg/sec</u>	Initial mass, m_0 <u>kg</u>	Burn time, τ_f <u>sec</u>
1	4448222	1068.2	611582.1	377.65
2	889644.3	213.1	164846.5	167.42
3	85636.8	30.5	19820.0	
Final states				
Branch number	Velocity, v^{**} <u>km/sec</u>	Gamma, γ <u>deg</u>	Altitude, h <u>km</u>	
2	7.796727	0	179.0	
3	7.780724	0	206.0	
Physical constants				
	Earth's radius, r_e <u>km</u>	Gravitational constant, μ <u>km³/sec²</u>		
	6,378.165	398,603.2		

**Circular orbit velocity.

All integrations were performed using a standard 4th-order Runge-Kutta program with the time interval for branch one, branch two and branch three divided into 60, 40, and 40 uniform segments, respectively. The same quadratic polynomial interpolation scheme mentioned earlier was used for each one-dimensional minimization.

The nominal control used as the initial guess of the optimal control was

$$u^1(\tau) = 0.4 \text{ rad}, \quad 0 \leq \tau \leq \tau_f^1,$$

$$u^2(\tau) = 0.1 \text{ rad}, \quad 0 \leq \tau \leq \tau_f^2,$$

$$u^3(\tau) = 0.2 \text{ rad}, \quad 0 \leq \tau \leq \tau_f^3,$$

$$\alpha^3 = 1,$$

where τ_f^3 was assumed to be 250 sec. The value of ϵ used in determining constraint satisfaction was 10^{-1} . Although this value is fairly large, the two terminal altitudes are normally satisfied to less than .01 km, the two terminal velocities are normally satisfied to less than 0.0001 km/sec, and the two terminal flight path angles to less than $(.1)10^{-6}$ radian. The norm used to measure $\|\psi\|$ was the ordinary Euclidean norm with no weighting factor on any of the elements.

Three steps of the procedure outlined previously were performed with a total execution time of 46.09 sec. Some numerical results of the proposed algorithm are given in Table 5. The parameter \tilde{g}_α listed in Table 5 represents the projected gradient of α^3 . In addition, altitude versus velocity plots for branch 2 and branch 3 are given in Fig. 6, and the control obtained by step 3 is given in Fig. 7.

Table 5. Convergence history of branched trajectory problem.

Step number	J kg	$\langle \tilde{g}, \tilde{g} \rangle$	\tilde{g}_α	$h^1(t_f)$ km	$v^1(t_f)$ km/sec	$\gamma^1(t_f)$ rad	NFEVAL	ANFECC
0	-12,195.0	303	0.0496	180.12	6.7325	0.04939	1	
0 _{ψ}	-13,466.1	1700	0.2790	177.80	6.7917	0.02337	6	6
1	-13,476.5	209	0.0343	178.51	6.7918	0.02258	12	1.4
2	-13,477.8	107	0.0176	178.71	6.7917	0.02320	6	1
3	-13,478.2	27	0.0045	178.79	6.7917	0.02221	10	1

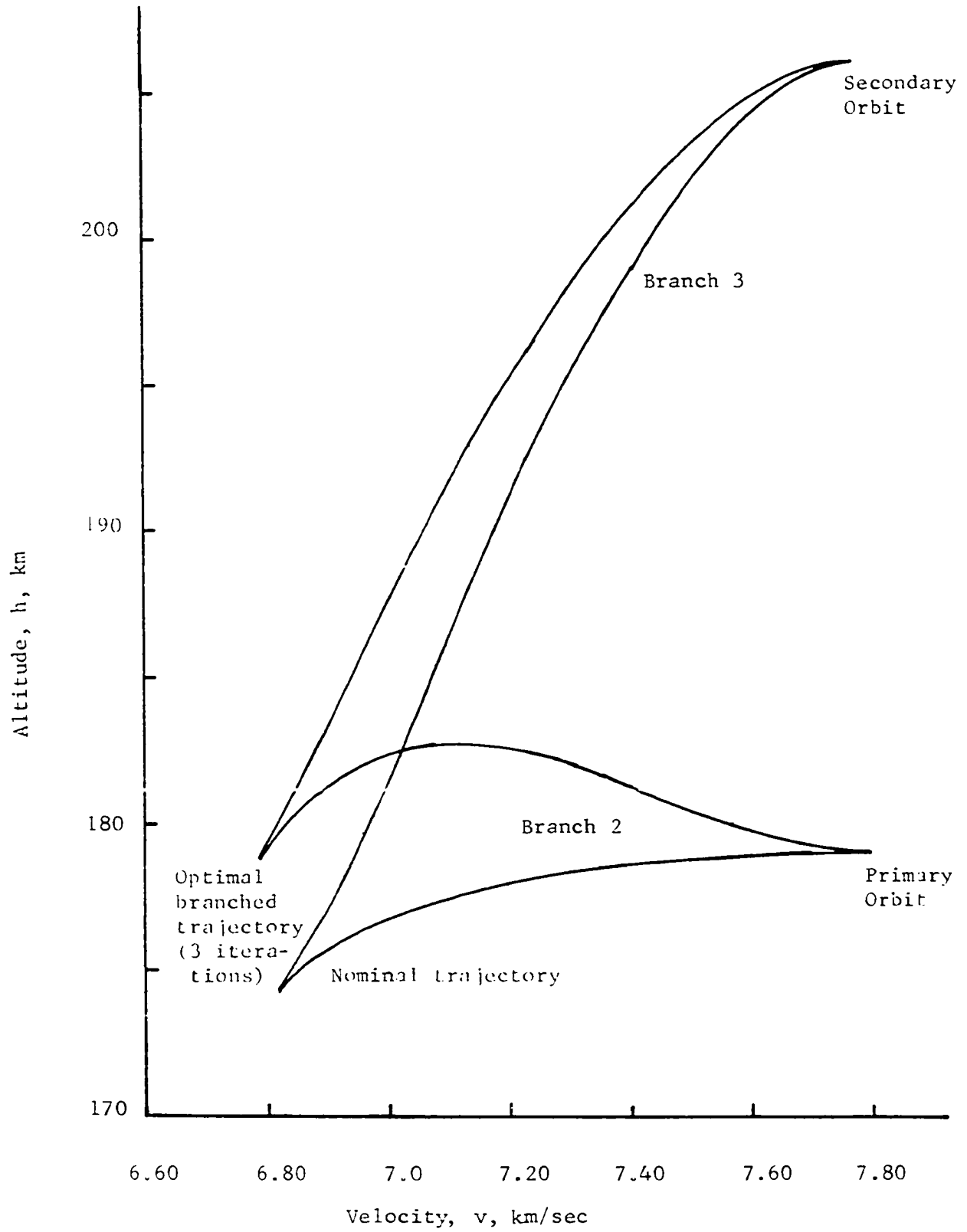


Fig. 6. Nominal and optimal trajectories for branches 2 and 3.

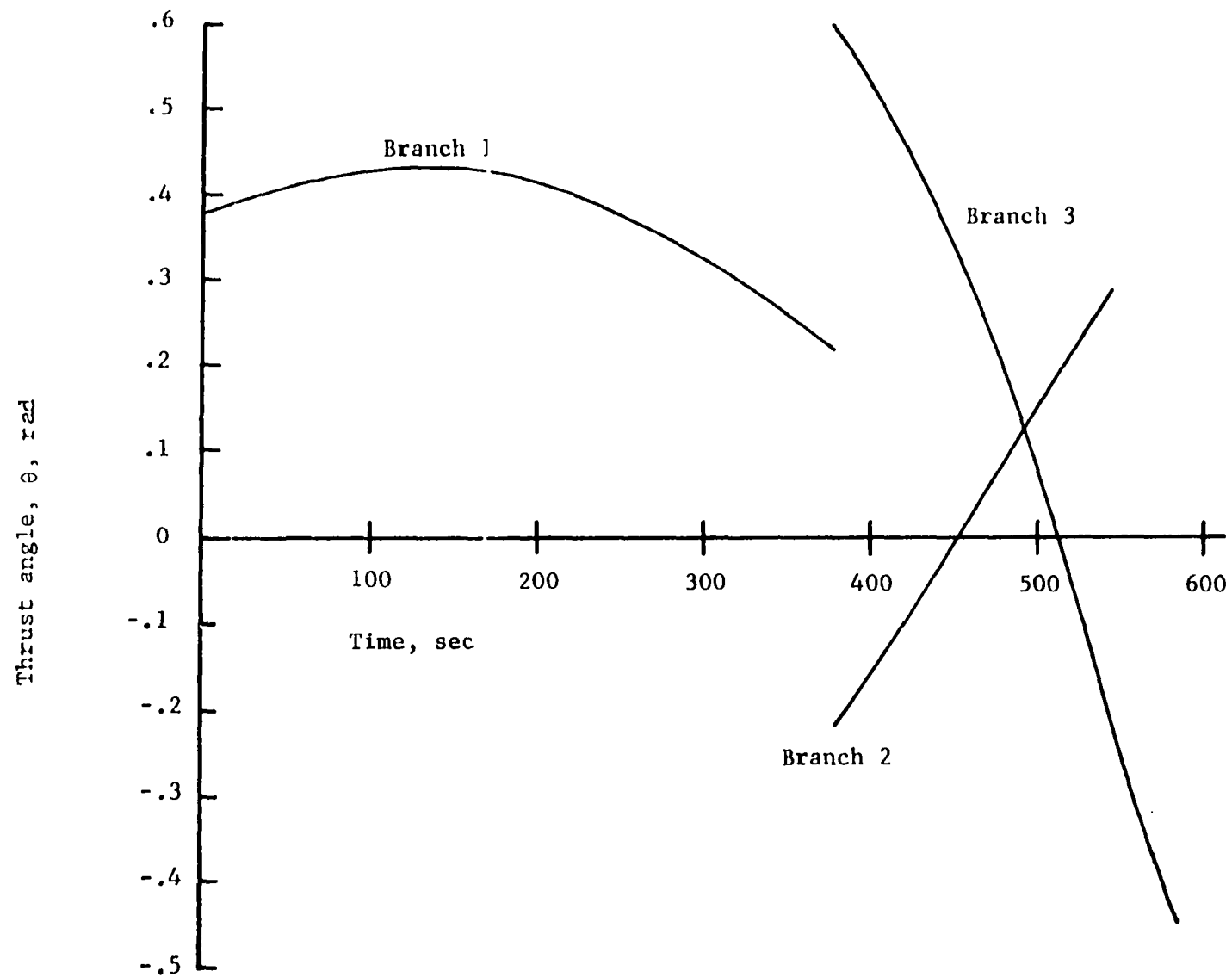


Fig. 7. Optimal control for Command Service Module abort problem.

Figs. 6 and 7 correspond very well with similar figures presented by Mason.³¹ However, Mason claims a branch 3 payload of 14,000 kg, whereas only a 13,478 kg payload is obtained here. This represents a difference in branch three burn time of approximately 17 sec. Due to this apparent discrepancy, additional investigation was deemed desirable.

Earlier in this section it was stated that the launch vehicle for the primary mission must have performance capability in excess of that required for the primary mission. In order to determine what performance capability is required to just fulfill the primary mission with no excess capability a conventional minimum time study can be performed on branches 1 and 2 alone allowing the time interval of branch 2 to vary. In addition, this minimum time solution also provides Mason³¹ an initial guess on the value of the unknown multiplier functions λ^1 and λ^2 which he must have in order to start the first iteration of his indirect optimization technique for solving the optimal branched trajectory problem. This minimum time solution maximizes the branch 2 payload, and the resultant solution will be referred to as nominal solution one or nominal trajectory one for the reason just stated. Mason reports a branch 2 payload of 129,805 kg which is equivalent to a branch 2 burn time of 163.4 sec. Using a simplified form of the equations derived earlier in this chapter, a branch 2 payload of 129,930 kg (or a branch 2 burn time of 163.9 sec) was obtained using the projected gradient method. Mason provides additional performance capability for the primary mission by specifying for the optimal branched trajectory a payload of 129,168 kg, thus allowing 637 kg more fuel to be

burned on the branch 2 trajectory. The value of 129,168 kg is actually 762 kg more than necessary according to this investigation so that the optimal branched solution found here should be better in terms of the branch 3 payload than that obtained by Mason.

Additional insight can be gained by performing a minimum time analysis of branch 3, referred to as nominal solution two or nominal trajectory two, assuming the staging point of nominal trajectory one provides the fixed initial conditions. This was originally done by Mason³¹ to determine the maximum burn time, i.e. the minimum payload, that could be expected for the branched trajectory problem. Secondly, the minimum time study was performed to obtain an estimate of the value of the unknown multiplier function λ^3 at the branch point which Mason needed in order to start the first iteration of his indirect minimization technique for solving the optimal branched trajectory problem. This is the reason for it being referred to as nominal solution two. He reports a payload for this mission of 13,720 kg while only 13,148 kg payload was obtained here indicating a difference in branch 3 burn time of 18.8 sec. Mason's payload of 13,720 kg for nominal trajectory two is even larger than the optimal branched trajectory branch 3 payload of 13,478 kg obtained here. Rather obviously, branch 3 is the cause of the rather large discrepancy encountered earlier. Branch 2 of nominal trajectory one and nominal trajectory two obtained by this investigation are plotted on Fig. 6 and labeled as 'nominal trajectories'.

There is one striking similarity between Mason's solution³¹ and the solution reported here. Mason received an increase in branch 3 payload of

280 kg at the expenditure of an additional 637 kg of propellant in branch 2. This investigation received an increase in branch 3 payload of 330 kg at the expenditure of an additional 762 kg of propellant in branch 2.

These figures represent approximately the same branch 3 payload increase per additional kg of branch 2 propellant expended. More specifically, they represent 0.440 fractional increase in the first case, and 0.433 fractional increase in the latter.

The reason for the increase in branch 3 payload for the branched trajectory can be partially established from Fig. 6. The branched solution starts at approximately the same velocity as the nominal but at a much higher altitude. This is more favorable to the secondary or branch three mission. In addition, the flight path angle of the optimal branched trajectory at the branch point is also slightly larger than that of the nominal trajectory which would also be more favorable to the secondary mission.

The rather large discrepancy of the two optimal branched solutions has not been determined; however, there is no reason to suspect the validity of any of the figures obtained by this investigation. One possible source of error is the thrust magnitude of the CSM reported by Mason³¹ to be 85,636.8 nt. Another reference³³ gives the thrust magnitude as 97,333 nt. This is approximately 14 percent higher which is definitely a significant amount. Solving the optimal branched trajectory problem once again changing only to this higher level of thrust for branch three, a branch 3 payload of 14,084 kg was obtained. This is 84 kg more than was reported by

Mason or a difference in burn time of approximately 2.8 seconds. Therefore, if Mason did cite an incorrect value of the CSM thrust magnitude a large amount of the original discrepancy is no longer a mystery.

CHAPTER 5. OPTIMAL CONTROL PROBLEMS WITH INEQUALITY CONSTRAINTS

Problem Development

One of the earliest attempts to treat inequality constraints on a function of the control and/or state variables was via a penalty function method. Kelley⁵ introduces an auxiliary state variable which is the integral to the present time of a quadratic measure of the violation of the inequality constraint. The terminal value of this quantity is treated as an additional terminal state constraint and brought as close to zero as is necessary to provide a satisfactorily small violation of the inequality constraint. The penalty function approach has the advantage that no a priori information is needed concerning either the number or location of the constrained arcs. This is not the case with projection operator techniques. Ordinarily, however, a good physical understanding of the physical variables involved in the problem can overcome this disadvantage of the projection operator technique.

As stated earlier in the Introduction, Denham and Bryson¹⁰ have extended their original work with projection operators to handle inequality constraints on functions of the control and/or state variables. They give numerical solutions to two atmospheric entry problems using both the projection operator and the previously mentioned penalty function approach. One of these problems places a constraint on the aerodynamic deceleration to be less than five times the acceleration of gravity at sea level. The other problem requires the vehicle to remain below a

specified altitude after its first dip into the atmosphere. More recently, McCart, Haug, and Streeter²⁰ have refined the results of Denham and Bryson¹⁰ for application to structural optimization problems. They present solutions to some fairly simple minimum weight structural problems for which the natural frequency of the structure is specified and the second moment of the cross-sectional area is held above a specified level.

The following problem statement contains only a single control variable as does the problem statement of Denham and Bryson,¹⁰ but it is sufficient to solve the preceding example problems. Consider the problem of determining the scalar control function $u(t)$, $t_0 \leq t \leq t_f$, which minimizes the cost functional

$$J = \phi[x(t_f)] + \int_{t_0}^{t_f} L(x, u, t) dt \quad (5.1)$$

subject to the n^{th} -order nonlinear dynamics

$$\dot{x}(t) = f(x, u, t), \quad x(t_0) = x_0, \quad (5.2)$$

with x_0 and t_0 given, subject to the p ($p \leq n$) terminal state constraints

$$\psi[x(t_f)] = 0, \quad (5.3)$$

and subject to the scalar control variable inequality constraint

$$C(x, u, t) \geq 0, \quad (5.4)$$

or the scalar state variable inequality constraint

$$S(x, t) \geq 0. \quad (5.5)$$

Before proceeding with the development of the projection operator it is necessary to obtain a complete understanding of the state variable inequality constraint $S \geq 0$. Since the constraint function S must be zero for the entire time interval that the solution lies on the constraint boundary, higher order derivatives also must vanish over the constrained time interval, that is

$$\frac{d^k S}{dt^k} = S^k = 0, \quad k = 1, 2, \dots$$

where S^k indicates the k^{th} derivative of S with respect to time. Since

$$\begin{aligned} \frac{dS}{dt} &= \frac{\partial S}{\partial t} + \frac{\partial S}{\partial x} \dot{x} \\ &= \frac{\partial S}{\partial t} + \frac{\partial S}{\partial x} f(x, u, t) \end{aligned}$$

it is possible that dS/dt may be an explicit function of the control variable. If not, the procedure may be repeated for the second, third, etc. derivative until the last time derivative does explicitly involve the control variable. By definition, a state variable inequality constraint is of order q if the q^{th} derivative is the lowest order derivative to explicitly involve the control variable.

The development of the projection operator here is for only one constrained arc, i.e. there is only one time interval along the trajectory where the inequality constraint is active. However, the extension to additional constrained arcs is straightforward. During the time period

that the trajectory is on the constrained arc for a control variable inequality constraint, the control variable u is determined by Eq. (5.4). Neighboring solutions for this time period must satisfy

$$\delta C = C_x \delta x + C_u \delta u = 0. \quad (5.6)$$

During the time period that the trajectory is on the constrained arc for a state variable inequality constraint, the control variable u is determined by

$$S^q(x, u, t) = 0, \quad (5.7)$$

so that S^q plays the same role as C along the constrained arc. In addition, neighboring solutions on the constrained arc must satisfy

$$\delta S^q = S_x^q \delta x + S_u^q \delta u = 0. \quad (5.8)$$

A schematic representation S , S^q , and C time histories is given in Fig. 8 where t_1 and t_2 represent the entering and exit times, respectively. Perhaps the largest difference in the application of the projection operator here as opposed to the treatment given by Denham and Bryson¹⁰ is shown in Fig. 8. The algorithm presented in Ref. 10 ensures continuity of S^q at the exit corner as well as ensuring continuity of C at both corners. That is not the case for this investigation. Instead, the algorithm presented here specifically provides for a discontinuity of S or C at both the constraint entrance and exit points and attempts to drive this discontinuity to zero. Obviously, both treatments of the projection operator must be restricted to problems for which the solution

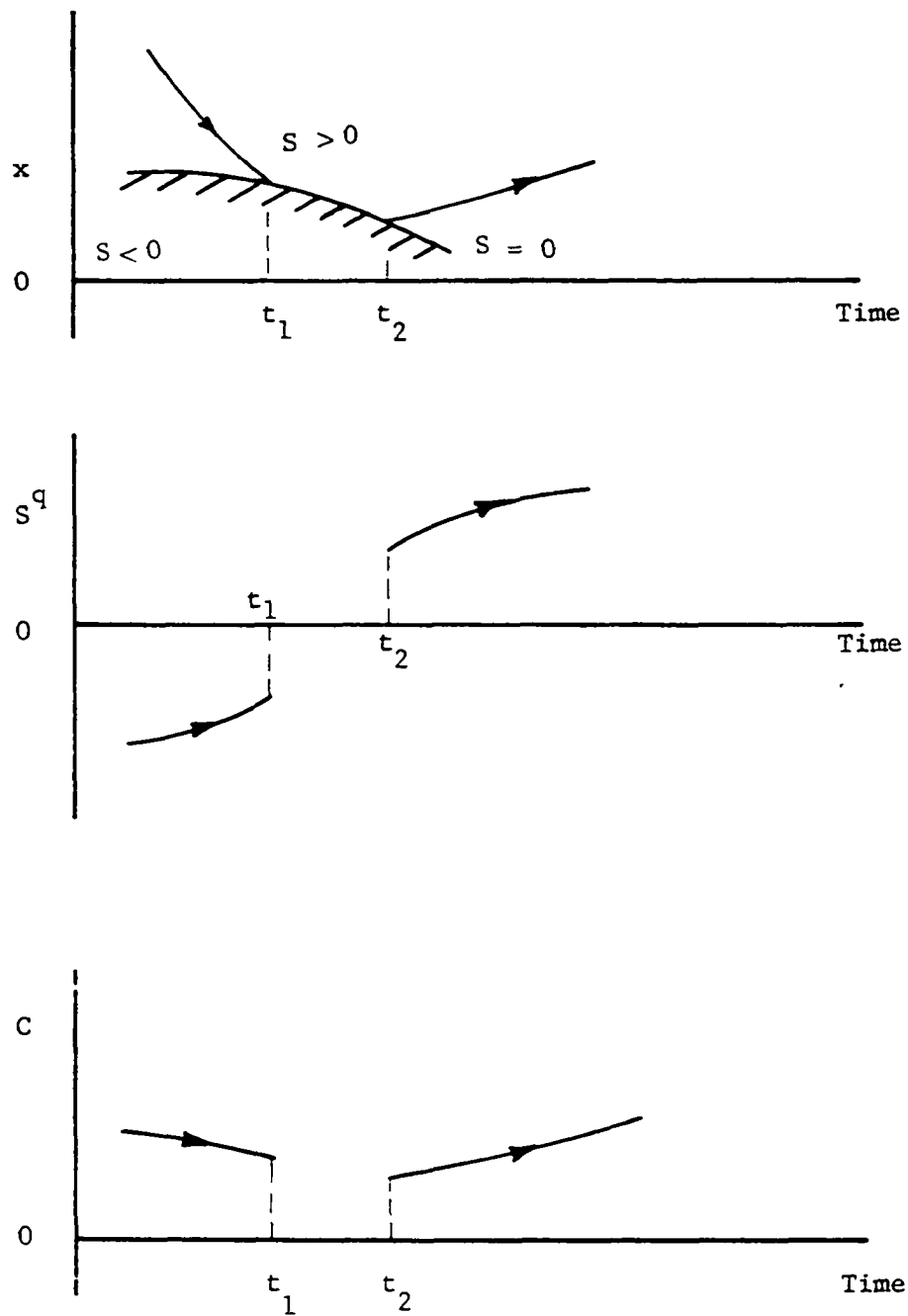


Fig. 8. Schematic representation of S , S^q , and C time histories

is, indeed, continuous at both corners. Denham and Bryson¹⁰ treat problems with discontinuous or 'bang-bang' controls as a special case. This report does not consider them at all.

Fig. 8 indicates S^q is negative at t_1^- and positive at t_2^+ . The reasoning for S^q at t_1^- being negative is that with $S^k = 0$, $k = 1, 2, \dots, (q-1)$, if S^q is not negative the trajectory will move away from the constraint boundary. At t_2^+ the trajectory is supposed to move away from the constraint boundary, and therefore S^q should be positive at that point.

In the following development, the trajectory can be broken into three branches by either the time transformation (4.4) or (4.5). The three branches are distinguished by superscripts j , $j = 1, 2, 3$, with the superscripts left off the independent variable time and all subscripts wherever the usage makes the value of the superscript clear. The projection operator development which follows actually uses time transformation (4.5) since it is the more general of the two time transformations. In addition, the development is for a state variable inequality constraint since the control variable inequality constraint is essentially a trivial special case.

Adopting the time transformation (4.5) and assuming that branch 2 is the one constrained arc, the minimization problem can be stated as: Find the three scalar control functions u^j , $j = 1, 2, 3$ and the three scalar control parameters α^j , $j = 1, 2, 3$ which minimize

$$J = \phi[x^3(\tau_f)] + \sum_{j=1}^3 \int_0^{\tau} f_{\ell}^j(x^j, u^j, \alpha^j, t) dt, \quad (5.9)$$

subject to the n^{th} -order nonlinear dynamics on each branch

$$\dot{x}^j(\tau) = F^j(x^j, u^j, \alpha^j, \tau), \quad x^j(0) = x_o^j, \quad j = 1, 2, 3, \quad (5.10)$$

with x_o^1 given, $x^{j+1}(0) = x^j(t_f)$ for $j = 1, 2$, subject to the p ($p \leq n$) terminal state constraints

$$\psi[x^3(\tau_f)] = 0, \quad (5.11)$$

subject to the control variable equality constraint

$$s^q(x^2, u^2, \alpha^2, \tau) = 0 \quad (5.12)$$

along the constrained arc, and subject to the mid-point constraints

$$N[x^1(\tau_f), \alpha^1] = \begin{bmatrix} s^1[x^1(\tau_f), \alpha^1] \\ s^1[x^1(\tau_f), \alpha^1] \\ \vdots \\ s^{q-1}[x^1(\tau_f), \alpha^1] \end{bmatrix} = 0, \quad (5.13)$$

where it is assumed on each branch that $F(x, u, \alpha, \tau)$ equals $\alpha f(x, u, t_o + \alpha\tau)$, $l(x, u, \alpha, \tau)$ equals $\alpha L(x, u, t_o + \alpha\tau)$, $s^k(x, \alpha)$ equals $s^k(x, t_o + \alpha\tau_f)$ $k = 1, 2, \dots, (q-1)$, and $s(x, u, \alpha, \tau)$ equals $S^q(x, u, t_o + \alpha\tau)$.

Linearization of the system dynamics (5.10) about nominal $x^j(\tau)$, $u^j(\tau)$, and α^j gives

$$\delta x^j(\tau) = F_{x^j}^j(x^j, u^j, \alpha^j, \tau) \delta x(\tau) + F_{u^j}^j(x^j, u^j, \alpha^j, \tau) \delta u^j(\tau) + F_{\alpha^j}^j(x^j, u^j, \alpha^j, \tau) \delta \alpha^j \quad j=1,2,3. \quad (5.14)$$

for which

$$\delta \mathbf{x}^j(\tau) = \bar{\Phi}^j(\tau, 0) \delta \mathbf{x}^j(0) + \int_0^\tau \bar{\Phi}^j(\tau, t) F_u^j(t) \delta u^j(t) dt + \int_0^\tau \bar{\Phi}^j(\tau, t) F_\alpha^j(t) dt \delta \alpha^j \quad j=1,3. \quad (5.15)$$

If each element of $N[\mathbf{x}^1, \alpha^1]$ is differentiable with respect to \mathbf{x}^1 and α^1 and if each iteration in the process satisfies Eq. (5.13), then near a nominal $\mathbf{x}^1(\tau_f)$ and α^1

$$\begin{aligned} N_{\mathbf{x}}[\mathbf{x}^1(\tau_f), \alpha^1] \delta \mathbf{x}^1(\tau_f) + N_\alpha[\mathbf{x}^1(\tau_f), \alpha^1] \delta \alpha^1 &\approx \\ N[\mathbf{x}^1 + \delta \mathbf{x}^1, \alpha^1 + \delta \alpha^1]_{\tau=\tau_f} - N[\mathbf{x}^1(\tau_f), \alpha^1] &= 0. \end{aligned} \quad (5.16)$$

Combining Eq. (5.15) with Eq. (5.16), after noting $\delta \mathbf{x}^1(0) = 0$, yields

$$\int_0^{\tau_f} N_{\mathbf{x}} \bar{\Phi}^1(\tau_f, t) F_u^1(t) \delta u^1(t) dt + \int_0^{\tau_f} N_{\mathbf{x}} \bar{\Phi}^1(\tau_f, t) F_\alpha^1(t) dt \delta \alpha^1 + N_\alpha \delta \alpha^1 = 0$$

which can be simplified to

$$\int_0^{\tau_f} N_{\mathbf{x}} \bar{\Phi}^1(\tau_f, t) F_u^1(t) \delta u^1(t) dt + \left[\int_0^{\tau_f} N_{\mathbf{x}} \bar{\Phi}^1(\tau_f, t) F_\alpha^1(t) dt + N_\alpha \right] \delta \alpha^1 = 0. \quad (5.17)$$

Defining the $q \times 1$ time varying matrix $\bar{\xi}^1(t)$ as

$$\bar{\xi}^1(t) = N_{\mathbf{x}} \bar{\Phi}^1(\tau_f, t) F_u^1(t) \quad (5.18)$$

and the $q \times 1$ constant matrix γ^1 as

$$\gamma^1 = \int_0^{\tau_f} N_{\mathbf{x}} \bar{\Phi}^1(\tau_f, t) F_\alpha^1(t) dt + N_\alpha \quad (5.19)$$

Eq. (5.17) is simplified to

$$\int_0^{\tau_f} \bar{\xi}^1(t) \delta u^1(t) dt + \gamma^1 \delta \alpha^1 = 0. \quad (5.20)$$

From Eq. (5.15) $\delta \mathbf{x}^3(\tau)$ and $\delta \mathbf{x}^1(\tau)$ are determined by

$$\begin{aligned} \delta \mathbf{x}^3(\tau) = & \bar{\Phi}^3(\tau, 0) \delta \mathbf{x}^3(0) + \int_0^\tau \bar{\Phi}(\tau, t) F_u^3(t) \delta u^3(t) dt \\ & + \int_0^\tau \bar{\Phi}(\tau, t) F_\alpha^3(t) dt \delta \alpha^3 \end{aligned} \quad (5.21)$$

and

$$\begin{aligned} \delta \mathbf{x}^1(\tau) = & \int_0^\tau \bar{\Phi}^1(\tau, t) F_u^1(t) \delta u^1(t) dt \\ & + \int_0^\tau \bar{\Phi}^1(\tau, t) F_\alpha^1(t) dt \delta \alpha^1. \end{aligned} \quad (5.22)$$

The specification that s^q equal zero on branch two requires

$$\delta s^q(x^2, u^2, \alpha^2, \tau) = s_x^q \delta x^2 + s_u^q \delta u^2 + s_\alpha^q \delta \alpha^2 = 0.$$

Solving for δu^2 from the above equation

$$\delta u^2(\tau) = -s_x^q / s_u^q \delta x^2(\tau) - s_\alpha^q / s_u^q \delta \alpha^2$$

which can then be combined with Eq. (5.14) to obtain

$$\delta \dot{\mathbf{x}}^2(\tau) = [F_x^2 - F_u^2 s_x^q / s_u^q] \delta \mathbf{x}^2(\tau) + [F_\alpha^2 - F_u^2 s_\alpha^q / s_u^q] \delta \alpha^2 \quad (5.23)$$

The variation of $\mathbf{x}^2(\tau)$ can then be written as

$$\delta \mathbf{x}^2(\tau) = \bar{\Phi}^2(\tau, 0) \delta \mathbf{x}^2(0) + \int_0^\tau \bar{\Phi}^2(\tau, t) [F_\alpha^2 - F_u^2 s_\alpha^q / s_u^q] dt \delta \alpha^2 \quad (5.24)$$

where now $\bar{\Phi}^2(\tau, t)$ is the state transition matrix of the linearized system (5.23) as opposed to the linear system (5.14).

Since $\delta \mathbf{x}^3(0)$ equals $\delta \mathbf{x}^2(\tau_f)$ and $\delta \mathbf{x}^2(0)$ equals $\delta \mathbf{x}^1(\tau_f)$, Eqs. (5.21), (5.22) and (5.24) can be combined to yield

$$\begin{aligned}
\delta \mathbf{x}^3(\tau_f) = & \int_0^{\tau_f} \bar{\Phi}^3(\tau_f, 0) \bar{\Phi}^2(\tau_f, 0) \bar{\Phi}^1(\tau_f, t) F_{\alpha}^1(t) dt \delta \alpha^1 \\
& + \int_0^{\tau_f} \bar{\Phi}^3(\tau_f, 0) \bar{\Phi}^2(\tau_f, t) [F_{\alpha}^2 - F_u^2 s_{\alpha}^q / s_u^q] dt \delta \alpha^2 \\
& + \int_0^{\tau_f} \bar{\Phi}^3(\tau_f, t) F_{\alpha}^3(t) dt \delta \alpha^3 \\
& + \int_0^{\tau_f} \bar{\Phi}^3(\tau_f, 0) \bar{\Phi}^2(\tau_f, 0) \bar{\Phi}^1(\tau_f, t) F_u^1(t) \delta u^1(t) dt \\
& + \int_0^{\tau_f} \bar{\Phi}^3(\tau_f, t) F_u^3(t) \delta u^3(t) dt.
\end{aligned} \tag{5.25}$$

Now, if each element of $\psi[\mathbf{x}^3]$ is differentiable with respect to \mathbf{x}^3 and if each iteration in the process satisfies Eq. (5.11), then near a nominal $\mathbf{x}^3(\tau_f)$

$$\psi_{\mathbf{x}}[\mathbf{x}^3(\tau_f)] \delta \mathbf{x}^3(\tau_f) \approx \psi[\mathbf{x}^3(\tau_f) + \delta \mathbf{x}^3(\tau_f)] - \psi[\mathbf{x}^3(\tau_f)] = 0. \tag{5.26}$$

For convenience, define the $p \times 1$ time-varying matrices $\bar{\xi}^2(t)$ and $\bar{\xi}^3(t)$ and the $p \times 1$ constant matrices γ^2 , γ^3 , and γ^4 as

$$\begin{aligned}
\bar{\xi}^2(t) &= \psi_{\mathbf{x}} \bar{\Phi}^3(\tau_f, 0) \bar{\Phi}^2(\tau_f, 0) \bar{\Phi}^1(\tau_f, t) F_u^1(t), \\
\bar{\xi}^3(t) &= \psi_{\mathbf{x}} \bar{\Phi}^3(\tau_f, t) F_u^3(t) \\
\gamma^2 &= \int_0^{\tau_f} \psi_{\mathbf{x}} \bar{\Phi}^3(\tau_f, 0) \bar{\Phi}^2(\tau_f, 0) \bar{\Phi}^1(\tau_f, t) F_{\alpha}^1(t) dt, \\
\gamma^3 &= \int_0^{\tau_f} \psi_{\mathbf{x}} \bar{\Phi}^3(\tau_f, 0) \bar{\Phi}^2(\tau_f, t) [F_{\alpha}^2(t) - F_u^2 s_{\alpha}^q / s_u^q] dt, \\
\text{and} \quad \gamma^4 &= \int_0^{\tau_f} \psi_{\mathbf{x}} \bar{\Phi}^3(\tau_f, t) F_{\alpha}^3(t) dt.
\end{aligned}$$

Due to the five preceding definitions and Eq. (5.25), Eq. (5.26) becomes

$$\int_0^{\tau_f} \xi^2(t) \delta u^1(t) dt + \int_0^{\tau_f} \xi^3(t) \delta u^3(t) dt + \gamma^2 \delta \alpha^1 + \gamma^3 \delta \alpha^2 + \gamma^4 \delta \alpha^3 = 0. \quad (5.27)$$

Now let Ω be the Cartesian product of U^1 , U^3 , and A , i.e.
 $\Omega = U^1 \times U^3 \times A$, where U^1 is a set of scalar control functions $u^1(\tau)$,
 U^3 is a set of scalar control functions $u^3(\tau)$, and A is a set of 3-vector
control parameters α , with elements α^1 , α^2 , and α^3 . Therefore, Ω contains
vectors w of the form

$$w = \begin{bmatrix} w_1(\tau) \\ w_2(\tau) \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} = \begin{bmatrix} u^1(\tau) \\ u^3(\tau) \\ \alpha^1 \\ \alpha^2 \\ \alpha^3 \end{bmatrix} = \begin{bmatrix} u^1(\tau) \\ u^3(\tau) \\ \alpha \end{bmatrix}.$$

The control $u^2(\tau)$ for branch 2 is not an independent variable since it is
determined by Eq. (5.12) and therefore is not an element of w . The
variation of a vector w belonging to Ω defined in the natural way as

$$\delta w = \begin{bmatrix} \delta w_1(\tau) \\ \delta w_2(\tau) \\ \delta w_3 \\ \delta w_4 \\ \delta w_5 \end{bmatrix} = \begin{bmatrix} \delta u^1(\tau) \\ \delta u^3(\tau) \\ \delta \alpha^1 \\ \delta \alpha^2 \\ \delta \alpha^3 \end{bmatrix} = \begin{bmatrix} \delta u^1(\tau) \\ \delta u^3(\tau) \\ \delta \alpha \end{bmatrix}$$

is also an element of Ω . Define $\tilde{\Omega}$ as the subset of Ω such that all control variations $\delta\tilde{w}$ belonging to $\tilde{\Omega}$ satisfy Eq. (5.16) and (5.26). $\tilde{\Omega}$ is then the set of admissible control variations for this optimal control problem.

Now define the $(q+p) \times 5$ time-varying matrix $\xi(t)$, the $(q+p) \times 1$ time matrices $\beta(t)$ and $\rho(t)$ and the $(q+p) \times 3$ constant matrix γ as

$$\xi(t) = \begin{bmatrix} \xi^1(t) & | & 0 & | & \gamma^1 & | & 0 & | & 0 \\ \text{---} & & & & & & & & \\ \xi^2(t) & | & \xi^3(t) & | & \gamma^2 & | & \gamma^3 & | & \gamma^4 \end{bmatrix}, \quad (5.28)$$

$$\beta(t) = \begin{bmatrix} \xi^1(t) \\ \text{---} \\ \xi^2(t) \end{bmatrix},$$

$$\rho(t) = \begin{bmatrix} 0 \\ \text{---} \\ \xi^3(t) \end{bmatrix},$$

and $\gamma = \begin{bmatrix} \gamma^1 & | & 0 & | & 0 \\ \text{---} & & & & \\ \gamma^2 & | & \gamma^3 & | & \gamma^4 \end{bmatrix}.$

Also define an inner product on Ω by

$$\langle v, w \rangle = \int_0^{Tf} v_1(\tau) w_1(\tau) d\tau + \int_0^{Tf} v_2(\tau) w_2(\tau) d\tau + \sum_{i=3}^5 v_i w_i.$$

If the rows of $\zeta(\tau)$, which are denoted by the row vectors $\zeta_1, \zeta_2, \dots, \zeta_Q$ ($Q = q+p$), are linearly independent, they span a Q -dimensional subspace, call it Ω^* , of Ω . From Eqs. (5.20) and (5.27), the set of admissible control variations $\tilde{\Omega}$ can be defined by the relation

$$\tilde{\Omega} = \{\delta u \in \Omega: \langle \zeta'_i, \delta u \rangle = 0, i = 1, 2, \dots, Q\}. \quad (5.29)$$

Once again requiring Ω to be a Hilbert space, Ω is the direct sum of $\tilde{\Omega}$ and Ω^* . Next, define the projection operator P as $P[\delta u] = \delta \tilde{u}$, where $\delta \tilde{u}$ is the unique projection of δu onto $\tilde{\Omega}$. The subspace Ω^* , defined earlier, is given by

$$\begin{aligned} \Omega^* &= \{\delta u \in \Omega: \delta u = \sum_{i=1}^Q \eta_i \zeta'_i\}, \\ &= \{\delta u \in \Omega: \delta u = \zeta' \eta\}, \end{aligned} \quad (5.30)$$

where η is a Q -vector with constant elements $\eta_1, \eta_2, \dots, \eta_Q$.

Using the fact that $\Omega = \tilde{\Omega} \oplus \Omega^*$ and Eqs. (5.29) and (5.30) the following set of equations is generated.

$$\begin{aligned} \langle \zeta'_1, \zeta' \eta \rangle &= \langle \zeta'_1, \delta u \rangle \\ \langle \zeta'_2, \zeta' \eta \rangle &= \langle \zeta'_2, \delta u \rangle \\ &\vdots \\ \langle \zeta'_Q, \zeta' \eta \rangle &= \langle \zeta'_Q, \delta u \rangle. \end{aligned}$$

The above set of equations is more concisely written using the definitions of $\beta(t)$, $\rho(t)$ and γ as

$$\left[\int_0^{\tau_f} \beta \beta' dt + \int_0^{\tau_f} \rho \rho' dt + \gamma \gamma' \right] \eta = \int_0^{\tau_f} \beta \delta u^1 dt + \int_0^{\tau_f} \rho \delta u^3 dt + \gamma \delta \alpha. \quad (5.31)$$

Since $\delta \tilde{w}$ is equal to δw minus δw^* , solving the above equation for η and combining this with Eq. (5.30) gives

$$\delta \tilde{w} = \delta w - \zeta' \left[\int_0^{\tau_f} \beta \beta' dt + \int_0^{\tau_f} \rho \rho' dt + \gamma \gamma' \right]^{-1} \left[\int_0^{\tau_f} \beta \delta u^1 dt + \int_0^{\tau_f} \rho \delta u^3 dt + \gamma \delta \alpha \right]. \quad (5.32)$$

The above equation does not give the exact form of the projection operator but does supply all the needed information to apply the projection operator technique to the optimization problem.

Bryson, Denham, and Dreyfus³⁴ have derived the necessary minimizing conditions for the Mayer formulation of the preceding optimal control problem. After adjoining the dynamical system to the cost functional with the n -vector multiplier function $\lambda(\tau)$ and including the additional terms accounting for the problem of Bolza formulation and the linear time transformation, the first variation of the cost functional J due to a variation of the control vector w is

$$\begin{aligned} \delta J = & \left[(\phi_x - \lambda'^3) \delta x^3 \right]_{\tau=\tau_f} + [\lambda'^3(0) - \lambda^2(\tau_f)] \delta x^2(\tau_f) \\ & + [\lambda'^2(0) - \lambda'^1(\tau_f)] \delta x^1(\tau_f) + \int_0^{\tau_f} [(H_x^1 + \dot{\lambda}'^1) \delta x^1 + H_u^1 \delta u^1 + H_\alpha^1 \delta \alpha^1] dt \\ & + \int_0^{\tau_f} \{ [H_x^2 + \dot{\lambda}'^2 - H_u^2 s_x^q / s_u^q] \delta x^2 + [H_\alpha^2 - H_u^2 s_\alpha^q / s_u^q] \delta \alpha^2 \} dt \\ & + \int_0^{\tau_f} [(H_x^3 + \dot{\lambda}'^3) \delta x^3 + H_u^3 \delta u^3 + H_\alpha^3 \delta \alpha^3] dt. \end{aligned} \quad (5.33)$$

The function H (the Hamiltonian) is defined as

$$H[x, u, \alpha, \lambda, \tau] = \ell[x, u, \alpha, \tau] + \lambda' F[x, u, \alpha, \tau]$$

on each branch and therefore superscripts are left off. Choosing

$$\dot{\lambda}'^j(\tau) = -H_x^j = -\ell_x^j - \lambda'^j F_x^j, \quad j = 1, 3$$

$$\dot{\lambda}'^2(\tau) = -H_x^2 + H_u^2 s_x^q / s_u^q,$$

with boundary conditions

$$\lambda'^3(\tau_f) = 0_x, \lambda'^2(\tau_f) = \lambda'^3(0), \text{ and } \lambda'^1(\tau_f) = \lambda'^2(0),$$

Eq. (5.33) simplifies to

$$\begin{aligned} \delta J = & \int_0^{\tau_f} H_u^1 \delta u^1 dt + \int_0^{\tau_f} H_u^3 \delta u^3 dt + \int_0^{\tau_f} H_\alpha^1 dt \delta \alpha^1 \\ & + \int_0^{\tau_f} [H_\alpha^2 - H_u^2 s_\alpha^q / s_u^q] dt \delta \alpha^2 + \int_0^{\tau_f} H_\alpha^3 dt \delta \alpha^3. \end{aligned} \quad (5.34)$$

If the gradient is defined as

$$g = \begin{bmatrix} H_u^1 \\ H_u^2 \\ \int H_\alpha^1 dt \\ \int [H_\alpha^2 - H_u^2 s_\alpha^q / s_u^q] dt \\ \int H_\alpha^3 dt \end{bmatrix} \quad (5.35)$$

with the obvious limits on the integrals, Eq. (5.34) can be written as

$$\delta J = \langle g, \delta w \rangle. \quad (5.36)$$

Since the gradient (5.35) can be written as \tilde{g} plus g^* , Eq. (5.36) becomes

$$\delta J = \langle g^*, \delta w \rangle + \langle \tilde{g}, \delta w \rangle = \langle \tilde{g}, \delta w \rangle.$$

Since the control variation in Eq. (5.36) must be an admissible control variation. For an extremum, δJ must be zero for arbitrary δw ; this can happen only if $\tilde{g} = 0$.

If θ_i and p_i represent the stepsize and direction of search, respectively, then $w_{i+1} = w_i + \theta_i p_i$, and letting $p_i = -\tilde{g}_i$ assures that δJ is negative at each iteration. This is again the steepest descent method of picking the direction of search. The one-dimensional minimization technique outlined in Chapter 2 is applicable to problems treated in this chapter provided that w be substituted for u . At some steps the projection operator technique developed in this chapter may require the use of a constrained one-dimensional minimization. The constrained one-dimensional minimization technique and its application is discussed later.

One technique for treating unspecified final times and any unspecified mid-point times is that of a linear time transformation such as presented by Eq. (4.5), and the results of this have just been shown. There are basically two ways for treating unspecified final or mid-point times. The other technique is that of using the unspecified times themselves as control parameters as opposed to using the time intervals as control parameters. The derivation of the necessary minimizing conditions and the projection operator for this other formulation is actually quite straightforward. However, the application of it to a problem requires changing the integration limits, and this greatly complicates the programming logic.

For completeness, a brief derivation of the projection operator for this latter technique is given below.

Returning to the original statement of the optimal control problem (5.1-5.5) the mid-point constraints which must be satisfied are

$$N[x(t_1), t_1] = \begin{bmatrix} S[x(t_1), t_1] \\ S^1[x(t_1), t_1] \\ \vdots \\ S^{q-1}[x(t_1), t_1] \end{bmatrix} = 0,$$

which requires that

$$N_x dx + N_t dt \stackrel{\sim}{=} N[x + dx, t + dt] - N[x, t] = 0. \quad (5.38)$$

Linearization of the system dynamics (5.2) about a nominal $u(t)$ and $x(t)$ gives

$$\dot{\delta x}(t) = f_x(x, u, t)\delta x(t) + f_u(x, u, t)\delta u(t), \quad \delta x_0 = 0. \quad (5.39)$$

If $x_1(t)$ represents a trajectory which enters the constraint at t_1 and $x_2(t)$ is a neighboring trajectory which enters the constraint at $t_1 + dt$, then

$$dx(t_1) = x_2(t_1 + dt_1) - x_1(t_1) = \delta x(t_1) + \dot{x}(t_1^-) dt_1$$

where

$$\delta x(t_1) = \int_{t_0}^{t_1} \bar{\Phi}(t_1, \tau) f_u(\tau) \delta u(\tau) d\tau$$

and $\bar{\Phi}(t, \tau)$ is the state transition matrix of the linearized system

(5.39). Eq. (5.38) then becomes

$$\int_{t_0}^{t_1} N_x \bar{\varphi}(t_1, \tau) f_u(\tau) \delta u(\tau) d\tau + \{N_x \dot{x}(t_1^-) + N_t\} dt_1 = 0.$$

Defining the $q \times 1$ time-varying matrix $\xi^1(\tau)$ and the $q \times 1$ constant matrix γ^1 as

$$\xi^1(\tau) = N_x \bar{\varphi}(t_1, \tau) f_u(\tau)$$

and

$$\gamma^1 = N_x \dot{x}(t_1^-) + N_t,$$

the preceding equation is simplified to

$$\int_{t_0}^{t_1} \xi^1(\tau) \delta u(\tau) d\tau + \gamma^1 dt_1 = 0. \quad (5.40)$$

In addition, the terminal state constraints (5.3) must be satisfied at each iteration so that

$$\psi_x[x(t_f)] dx(t_f) \approx \psi[x(t_f) + dx(t_f)] - \psi[x(t_f)] = 0. \quad (5.41)$$

The equation to calculate $dx(t_f)$ is

$$dx(t_f) = \bar{\varphi}(t_f, t_2) \delta x(t_2) + \int_{t_2}^{t_f} \bar{\varphi}(t_f, \tau) f_u(\tau) \delta u(\tau) d\tau + \dot{x}(t_f) dt_f. \quad (5.42)$$

Along the constrained arc Eq. (5.8) must be satisfied. Solving for δu from Eq. (5.8) and combining with Eq. (5.39) gives the linear system the form

$$\dot{\delta x}(t) = [f_x + f_u s_x^q / s_u^q] \delta x \quad (5.43)$$

and consequently

$$\delta x(t_2) = \varphi(t_2, t_1) \delta x(t_1) + [\dot{x}(t_2^-) - \dot{x}(t_2^+)] dt_2 \quad (5.44)$$

where $\varphi(t, \tau)$ is now the state transition matrix of the linear system (5.43) as opposed to the linear system (5.39). Obtaining $\delta x(t_1)$ in an analogous manner, the value of $\delta x(t_1)$ is determined by

$$\delta x(t_1) = \int_{t_0}^{t_1} \tilde{\Phi}(t_1, \tau) f_u(\tau) \delta u(\tau) d\tau + [\dot{x}(t_1^-) - \dot{x}(t_1^+)] dt.$$

Combining this expression for $\delta x(t_1)$ with Eqs. (5.42) and (5.44), the expression for $dx(t_f)$ becomes

$$\begin{aligned} dx(t_f) = & \int_{t_0}^{t_1} \tilde{\Phi}(t_f, t_2) \varphi(t_2, t_1) \tilde{\Phi}(t_1, \tau) f_u(\tau) \delta u(\tau) d\tau \\ & + \int_{t_2}^{t_f} \tilde{\Phi}(t_f, \tau) f_u(\tau) \delta u(\tau) d\tau + \tilde{\Phi}(t_f, t_2) \varphi(t_2, t_1) [\dot{x}(t_1^-) - \dot{x}(t_1^+)] dt_1 \\ & + \tilde{\Phi}(t_f, t_2) [\dot{x}(t_2^-) - \dot{x}(t_2^+)] dt_2 + \dot{x}(t_f) dt_f. \end{aligned} \quad (5.45)$$

As a notational convenience, define the $p \times 1$ time-varying matrices $\xi^2(\tau)$ and $\xi^3(\tau)$ and the $p \times 1$ constant matrices γ^2 , γ^3 , and γ^4 as

$$\xi^2(\tau) = \psi_x \tilde{\Phi}(t_f, t_2) \varphi(t_2, t_1) \tilde{\Phi}(t_1, \tau) f_u(\tau) \delta u(\tau) d\tau,$$

$$\xi^3(\tau) = \psi_x \tilde{\Phi}(t_f, \tau) f_u(\tau) \delta u(\tau) d\tau,$$

$$\gamma^2 = \psi_x \tilde{\Phi}(t_f, t_2) \varphi(t_2, t_1) [\dot{x}(t_1^-) - \dot{x}(t_1^+)],$$

$$\gamma^3 = \psi_x \tilde{\Phi}(t_f, t_2) [\dot{x}(t_2^-) - \dot{x}(t_2^+)],$$

and $\gamma^4 = \psi_x \dot{x}(t_f).$

Due to the five preceding definitions and Eq. (5.45), Eq. (5.41) becomes

$$\int_{t_0}^{t_1} \xi^2(\tau) \delta u(\tau) d\tau + \int_{t_2}^{t_3} \xi^3(\tau) \delta u(\tau) d\tau + \gamma^2 dt_1 + \gamma^3 dt_2 + \gamma^4 dt_f. \quad (5.46)$$

The control space Ω is now the Cartesian product of U^1 , U^3 , and T where U^1 is a set of control functions on the time interval t_0 to t_1 , U^3 is a set of control functions on the time interval t_2 to t_f , and T is a set of 3-element control parameter vectors t . The three elements of the vector t are the unspecified times t_1 , t_2 , and t_f . Define an inner product on Ω as

$$\langle v, w \rangle = \int_{t_0}^{t_1} v(\tau) w(\tau) d\tau + \int_{t_2}^{t_f} v(\tau) w(\tau) d\tau + \sum_{i=3}^5 v_i w_i.$$

Defining the matrices $\xi(\tau)$, $\beta(\tau)$, $\rho(\tau)$, and γ in the same manner as earlier in this chapter, the development of the two projection operators is identical. The analogous equation to Eq. (5.32) is

$$\delta w = \delta w - \xi \left[\int_{t_0}^{t_1} \beta \beta' dt + \int_{t_2}^t \tilde{\beta} \rho \rho' dt + \gamma \gamma' \right]^{-1} \left[\int_{t_0}^{t_1} \tilde{\beta} \delta u dt + \int_{t_2}^t \rho \delta u dt + \gamma \delta t \right]. \quad (5.47)$$

The necessary minimizing conditions for the present optimization problem are again taken from Bryson, Denham, and Dreyfus³⁴. After adjoining the dynamical system to the cost functional with the r-vector multiplier function $\lambda(t)$ and including the additional terms accounting for the Bolza formulation of the problem instead of the Mayer formulation, the first variation of the cost functional J due to a variation of the control vector w is

$$\begin{aligned}
\delta J = & [(\lambda' \dot{x} - L)_{t=t_f} - (\lambda' \dot{x} + L)_{t=t_f}] dt_f + [\lambda'(t_2^+) - \lambda'(t_2^-)] dx(t_2) \\
& + [-(\lambda' \dot{x} + L)_{t=t_2^+} + (\lambda' \dot{x} + L)_{t=t_2^-}] dt_2 + [\lambda'(t_1^+) - \lambda'(t_1^-)] dx(t_1) \\
& + [-(\lambda' \dot{x} + L)_{t=t_1^+} + (\lambda' \dot{x} + L)_{t=t_1^-}] dt_1 + \int_{t_0}^{t_1} [(H_x + \dot{\lambda}') \delta x + H_u \delta u] dt \\
& + \int_{t_1}^{t_2} [\dot{\lambda}' + H_x - H_u S_x^q / S_u^q] \delta x dt + \int_{t_2}^{t_f} [(H_x + \dot{\lambda}') \delta x + H_u \delta u] dt. \quad (5.48)
\end{aligned}$$

The function H (the Hamiltonian) is defined as

$$H[x, u, \lambda, t] = L(x, u, t) + \lambda' f(x, u, t).$$

Choosing

$$\dot{\lambda}'(t) = -H_x = -L_x - \lambda' f_x$$

along the trajectory where the constraint is not active and

$$\dot{\lambda}'(t) = -H_x + H_u S_x^q / S_u^q$$

between time points t_1 and t_2 with boundary conditions

$$\lambda'(t_f) = \lambda'_x, \quad \lambda'(t_2^-) = \lambda'(t_2^+), \quad \text{and} \quad \lambda'(t_1^-) = \lambda'(t_1^+),$$

Eq. (5.48) simplifies to

$$\begin{aligned}
\delta J = & [(\lambda' \dot{x} + L)_{t=t_f} - (\lambda' \dot{x} + L)_{t=t_f}] dt_f + [-(\lambda' \dot{x} + L)_{t=t_2^+} + (\lambda' \dot{x} + L)_{t=t_2^-}] dt_2 \\
& + [-(\lambda' \dot{x} + L)_{t=t_1^+} + (\lambda' \dot{x} + L)_{t=t_1^-}] dt_1 + \int_{t_0}^{t_1} H_u \delta u(t) dt + \int_{t_2}^{t_f} H_u \delta u(t) dt. \quad (5.49)
\end{aligned}$$

Defining the gradient as

$$g = \begin{bmatrix} H_u(t), & t_0 \leq t \leq t_1 \\ H_u(t), & t_2 \leq t \leq t_f \\ [-(\lambda' \dot{x} + L)_{t=t_1^+} + (\lambda' \dot{x} + L)_{t=t_1^-}] \\ [-(\lambda' \dot{x} + L)_{t=t_2^+} + (\lambda' \dot{x} + L)_{t=t_2^-}] \\ [(\lambda' \dot{x} + L)_{t=t_f}] \end{bmatrix}, \quad (5.50)$$

Eq. (5.44) is further simplified to

$$\delta J = \langle g, \delta w \rangle = \langle g^* + \tilde{g}, \tilde{\delta w} \rangle = \langle \tilde{g}, \tilde{\delta w} \rangle.$$

Again, for an extremum, δJ must be zero for arbitrary $\tilde{\delta w}$; this can only happen if $\tilde{g} = 0$. The method of steepest descent can be applied to this problem by picking a direction of search in the negative gradient direction.

Numerical Examples

All computations performed on the two numerical examples presented in this section were made on an IBM 360/65 computer using double precision arithmetic. All integrations were performed using a standard 4th-order Runge-Kutta program with the time interval divided into uniform segments along each constrained or unconstrained arc. The entire time interval was divided into 50 segments, and the uniform segment size of the various arcs was kept approximately equal. The same quadratic polynomial interpolation scheme mentioned earlier (with one exception which will be mentioned later) was used for each one-dimensional minimization.

Neither example presented has one constrained arc for which the projection operators here were developed. In the one case this necessitates a simplification of the previous development, and in the other case it requires a straightforward extension of the previous development. Both example problems have fixed terminal times. This requires for a total of k constrained and unconstrained arcs that the additional equation

$$\alpha^1 \tau_f^1 + \alpha^2 \tau_f^2 + \dots + \alpha^k \tau_f^k = t_f - t_o \quad (5.51)$$

be satisfied at each iteration when the linear time translation (4.5) is utilized. The additional equation

$$\alpha^1 + \alpha^2 + \dots + \alpha^k = t_f - t_o \quad (5.52)$$

must be satisfied when the linear time translation (4.4) has been used.

The variation of Eq. (5.51) is

$$\delta \alpha^1 \tau_f^1 + \delta \alpha^2 \tau_f^2 + \dots + \delta \alpha^k \tau_f^k = 0 \quad (5.53)$$

and the variation of Eq. (5.52) is

$$\delta \alpha^1 + \delta \alpha^2 + \dots + \delta \alpha^k = 0. \quad (5.54)$$

There are many ways of ensuring the satisfaction of the previously mentioned equations. One obvious way is to treat Eq. (5.53) or Eq. (5.54) as an additional row of $\zeta(t)$ in Eq. (5.28), thus making it a $(q + p + 1) \times 5$ time-varying matrix and making η a $(q + p + 1)$ -vector, at least for the one constrained arc case treated earlier. In this investigation α^k and $\delta \alpha^k$ were solved for in terms of α^i , $i = 1, 2, \dots, (k-1)$ and $\delta \alpha^i$,

$i = 1, 2, \dots, (k-1)$ from either Eqs. (5.51) and (5.53) or Eqs. (5.52) and (5.54) and substituted into Eqs. (5.20), (5.27), (5.9) and (5.10). This procedure has the effect of decreasing the dimensionality of the control vector and changing $\zeta(t)$ to a $(q + p) \times 4$ time-varying matrix while retaining $\tilde{\eta}$ as a $(q + p)$ -vector. Alternately, any other of the α^i 's and $\delta\alpha^i$'s could be solved for in terms of the other $k-1$ α^i 's and $\delta\alpha^i$'s and a similar procedure carried out.

If neither of the linear time transformations are used but rather the time points themselves used as the control parameters, a much easier technique of specifying a fixed time point is used. In particular, if the i^{th} time point is specified, the i^{th} column of γ is deleted. This in turn requires also deleting a column from $\zeta(t)$.

It was mentioned earlier that the use of the one-dimensional minimization in this chapter differs somewhat from its use in previous chapters. Referring to Fig. 9, θ_1 is the result of determining θ such that $J[u+\theta p]$ as a function of θ is a minimum where p still represents the projected direction of search. The parameter θ_2 is the result of determining θ such that $J[(u+\theta p)_{\downarrow}]$ as a function of θ is a minimum. This parameter θ_2 is the stepsize used in the preceding chapters of this report and from now on is referred to as the unconstrained stepsize. The parameter θ_3 shall be referred to as the constrained stepsize parameter. The exact reason for using this constrained stepsize parameter at any given step in the minimization process will be explained in the discussion of the individual example problem. However, in general, the stepsize is constrained so that a certain element or elements of the control do not vary

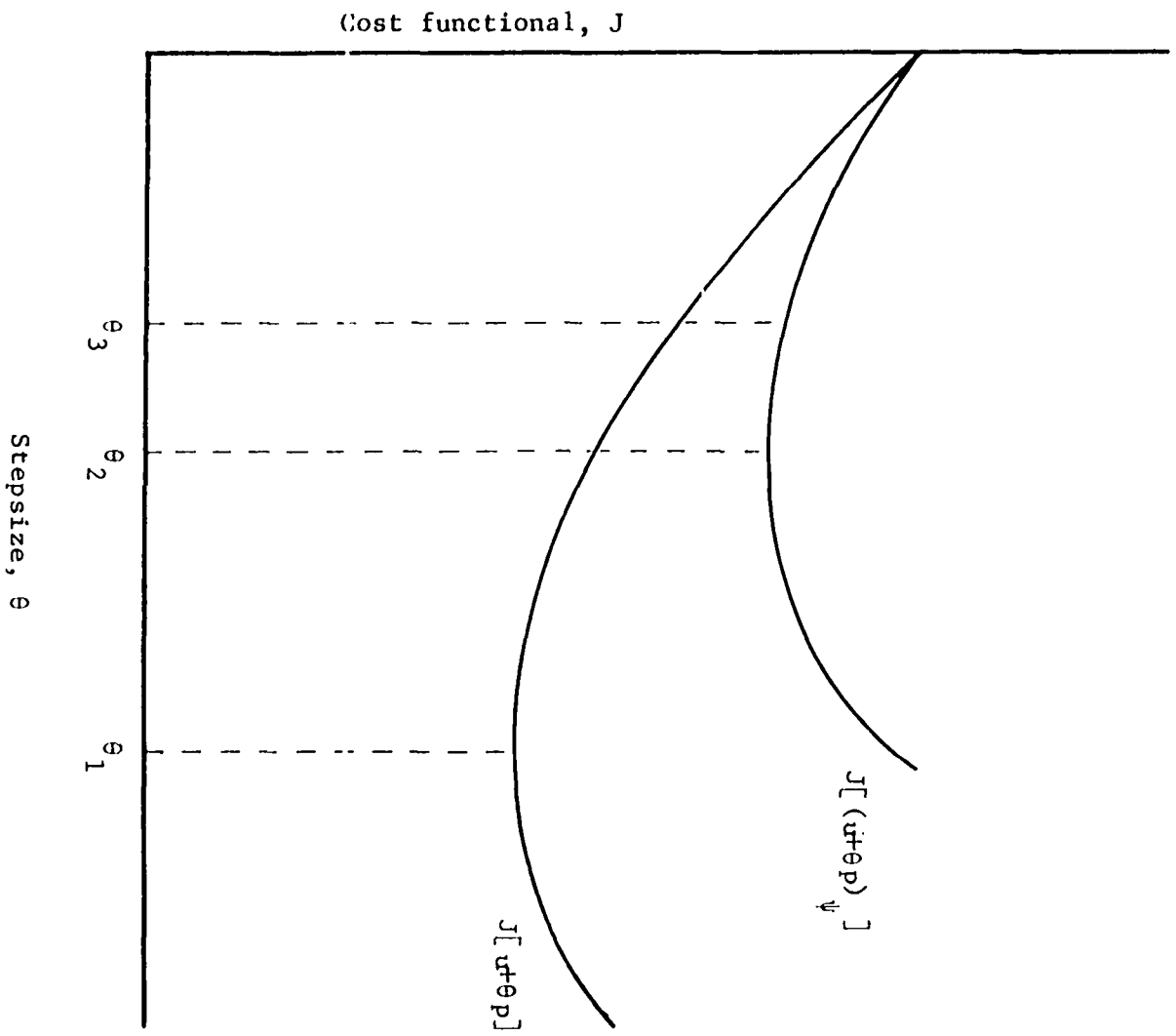


Fig. 9. One-dimensional function value profile.

by more than a given amount. Two important things to remember about the constrained one-dimensional minimization are: a) that the minimization still occurs along the constraint surface, and b) that if θ_3 is larger than θ_2 the one-dimensional minimization occurring in this chapter is identical with that of previous chapters.

One additional numerical scheme used in this chapter and not in the previous ones is the use of a conjugate direction algorithm in addition to the use of steepest descent. The conjugate gradient algorithm used here¹³ can be described by picking the direction of search to be

$$\begin{aligned} p_i &= -\tilde{g}_i + \beta p_{i-1}, \\ \beta &= \langle \tilde{g}_i, \tilde{g}_i \rangle / \langle \tilde{g}_{i-1}, \tilde{g}_{i-1} \rangle, \quad i > 0, \\ \beta &= 0, \quad i = 0. \end{aligned} \tag{5.55}$$

Historically, the above conjugate gradient algorithm was first used by engineers for unconstrained parameter optimization problems. The obvious intent was to obtain the solution to the parameter optimization problem more quickly than possible by steepest descent by taking advantage of the algorithm's quadratic convergence capabilities. The algorithm was first used in connection with optimal control problems by Lasdon, Mitter, and Waren.⁸

Rectangular wing problem

The problem of minimizing the skin mass of the cantilever straight wing shown in Fig. 10 subject to a minimum skin thickness constraint is

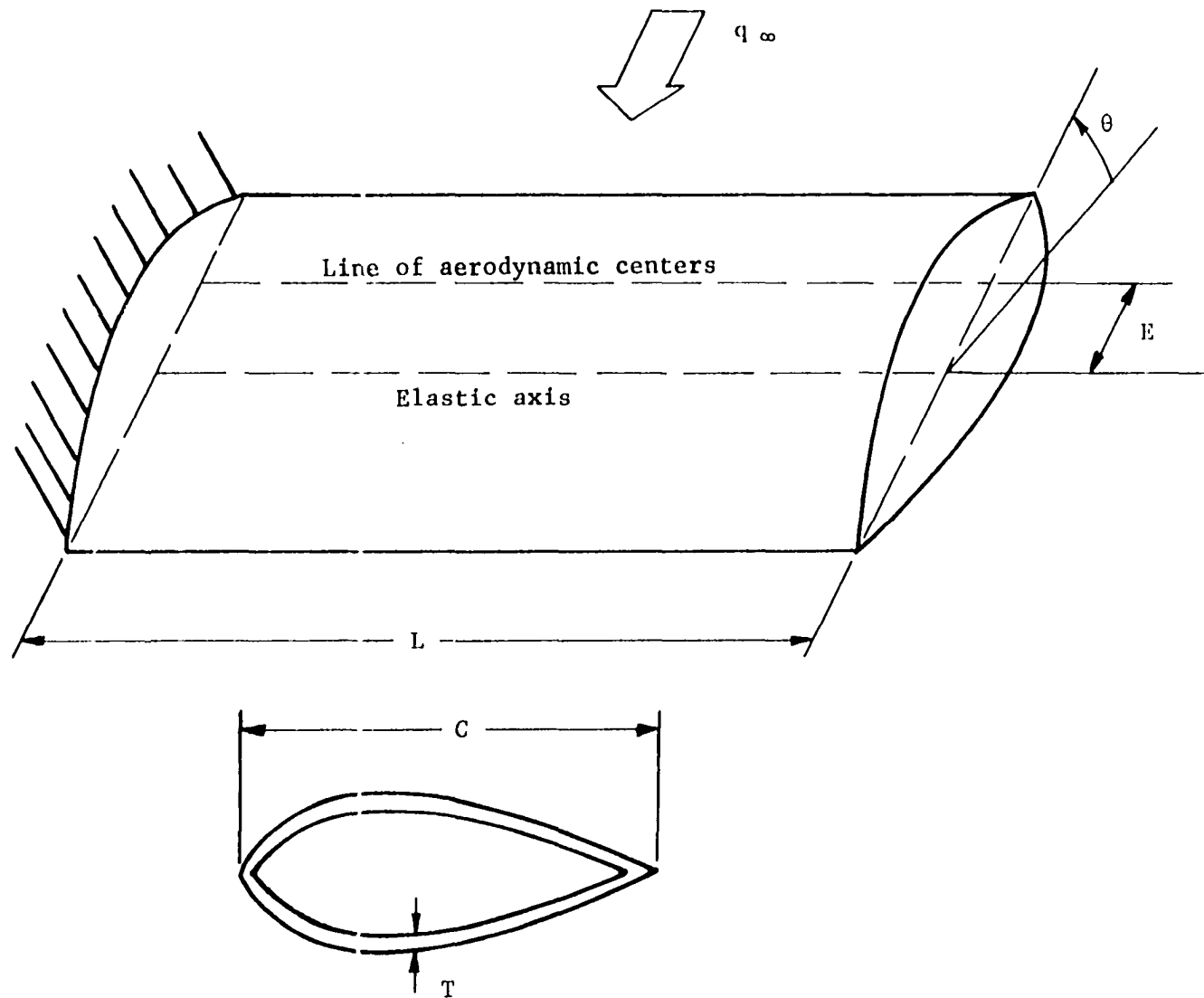


Fig. 10. Cantilever wing with constant chord.

treated both theoretically and numerically by indirect optimization techniques by Armand and Vitte.³⁵ The wing cross-sectional profile, constant along the span, has a lift-coefficient slope α_0 , and the wing's elastic axis is perpendicular to the free stream. The system is governed by the 2nd-order differential equation

$$\frac{d}{dX} \left(GJ \frac{d\theta}{dX} \right) + qCEa_0\theta = 0 \quad (5.56)$$

where q , G , and J are respectively the dynamic pressure, the modulus of rigidity, and the polar moment of inertia, and C and E are defined in Fig. 5. Eq. (5.56) is subject to the boundary conditions

$$\theta(0) = 0, \quad (5.57)$$

$$\text{and} \quad \left[GJ \frac{d\theta}{dX} \right]_{X=L} = 0. \quad (5.58)$$

The first states that the wing is fixed at $X = 0$, while the second indicates that there is no applied torque at the free end with $X = L$.

Armand and Vitte³⁵ assume the torsional stiffness of the wing is dominated by the contribution from the skin so the torsion constant J is directly proportional to the thickness T of the skin

$$J = kT, \quad (5.59)$$

where k is constant. Introducing the dimensionless quantities

$$x = X/L \quad \text{and} \quad t = T/T_0$$

where T_0 is the constant skin thickness of a reference wing with identical cross-section, Eqs. (5.56-5.58) become

$$(t\theta')' + \omega^2 \theta = 0,$$

$$\theta(0) = 0,$$

$$\text{and} \quad [t\theta']_{x=1} = 0.$$

The notation $()'$ denotes differentiation with respect to x and

$$\omega^2 = \frac{qCEa_0}{GkT_0} L^2 = \frac{qCEa_0}{GJ_0} L^2. \quad (5.60)$$

Armand and Vitte³⁵ show that a flutter condition exists for a uniform panel ($t=1$) for $\omega = (2n+1)\pi/2$, $n = 0, 1, 2, \dots$. The lowest value of ω for which flutter can exist is $\pi/2$. This corresponds to the torsional divergence dynamic pressure of

$$q_D = \frac{GJ_0 \pi^2}{4CEa_0 L^2}.$$

The optimization problem can now be stated as:

$$\text{Minimize} \quad \int_0^1 t(x) dx$$

subject to the 2^{nd} -order differential equation

$$(t\theta')' + \omega^2 \theta = 0$$

with boundary conditions

$$\theta(0) = 0 \text{ and } [t\theta']_{x=1} = 0,$$

and subject to the skin thickness constraint

$$t(x) \geq t_{\min},$$

while holding ω equal to $\pi/2$.

The minimization problem can be transformed into more familiar notation by letting

$$t = x,$$

$$u(t) = t(x),$$

$$x_1(t) = \theta(x),$$

$$\text{and } x_2(t) = t\theta'(x).$$

Now the optimal control problem is to determine the scalar control function $u(t)$, $0 \leq t \leq 1$ which minimizes the cost functional

$$J = \int_0^1 u(t) dt$$

subject to the 2nd-order dynamical system

$$\dot{x}_1(t) = x_2(t)/u(t), \quad x_1(0) = 0,$$

$$\dot{x}_2(t) = -\omega^2 x_1(t), \quad x_2(1) = 0, \quad \omega = \pi/2,$$

and subject to the control inequality constraint

$$u(t) - u_{\min} \geq 0.$$

The analytical solution obtained by Armand and Vitte³⁵ for the above optimal control problem is

$$u(t) = u_{\min} + \frac{\omega^2}{2} (t_1^2 - t^2), \quad 0 \leq t \leq t_1$$

$$u(t) = u_{\min}, \quad t_1 < t \leq 1$$

where t_1 , the initial contact point of the constrained arc, is given by the solution of the transcendental equation

$$t_1 = \frac{\sqrt{u_{\min}}}{\omega} \cotan\left[\frac{\omega}{\sqrt{u_{\min}}}(1-t_1)\right].$$

The optimal control is shown to be unique, but this control determines only the shape of the state variables and not the magnitudes.

The nonuniqueness of the state variables could be treated (see McCart, Haug, and Streeter²⁰) by specifying the value of $x_2(0)$. In an effort to keep the problem solution as general as possible and also to utilize the material in Chapter 3 of this report concerning missing initial conditions of state variables the value of $x_2(0)$ was not fixed. It is interesting to note, however, that after the first three or four steps of each solution attempted the value of $x_2(0)$ is essentially constant and the gradient element associated with this variable is essentially zero. Treating $x_2(0)$ as a variable does add an additional term to the projection operator derived in this chapter which must be obtained from Chapter 3. However, this is a straightforward process.

Five solutions were attempted for the proposed optimal control problem. They are denoted as solution one, solution two, etc. In addition, two different nominal controls were used, denoted as nominal control one and nominal control two. Both initial conditions have the minimum control level u_{\min} equal to 0.5 and $x_2(0)$ equal to 0.3. The control time history $u(t)$, $0 \leq t \leq t_1$, and the contact time t_1 for both initial controls are shown in Fig. 11. These two nominals were chosen to demonstrate the

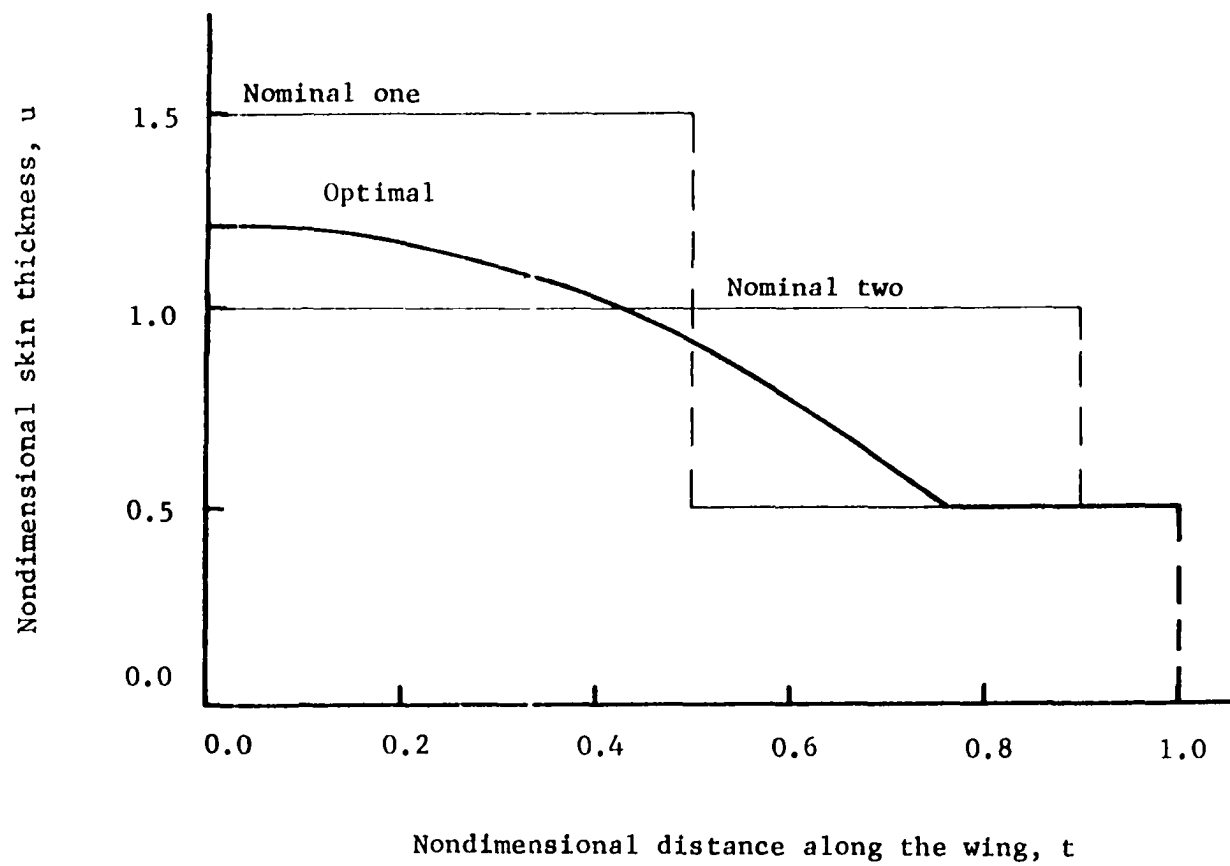


Fig. 11. Optimal and nominal controls for rectangular wing problem.

ability of the proposed algorithm to obtain the correct optimal skin thickness distribution whether the nominal value of t_1 was to the right or to the left of the optimal value of t_1 .

The use of the constrained one-dimensional minimization can now be explained by referring to Fig. 11. Due to the physical interpretation of the control being the skin thickness the nominal controls one and two have been drawn as two solid straight lines connected by a dashed line. However, there is actually a discontinuity of the control at t_1 . The numerical iterative process should drive the control at t_1^- , $u(t_1^-)$, to equal the value of the control at t_1^+ , $u(t_1^+)$, which is fixed at the minimum skin thickness value. During the one-dimensional minimization process it is obviously not allowable to let the control between 0 and t_1 take on values less than the minimum value 0.5. Referring to Fig. 9, the use of the constrained one-dimensional minimization stepsize θ_3 assures that this does not happen. It is interesting to note that the constraint on the one-dimensional minimization is active, i.e. $\theta_3 < \theta_2$, for only the first step of solutions one, two, and three, and only infrequently for solutions four and five. Although it is conceivable that any point in the control time history between 0 and t_1 might try to travel below the minimum thickness constraint, every case where the constraint is active θ_3 has been determined by the value of the control and direction of search at t_1^- .

Solution one was obtained using steepest descent in conjunction with the projection operator developed using the linear time transformation (4.4). The convergence properties of the first twenty steps of the proposed algorithm are shown in Table 6. The optimal values determined by

Table 6. Convergence of solution one of the rectangular wing problem: steepest descent from nominal control one.

Step number	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_1	\tilde{g}_{t_1}	$u(t_1^-)$	$\tilde{g}(t_1^-)$	NFEVAL	ANFECC
0	1.000000	$(.92)10^{-1}$	0.5000	$(-.37)10^{-1}$	1.5000	$(.33)$	1	
0_{ψ}	0.955754	$(.66)10^{-1}$	0.4601	$(-.50)10^{-1}$	1.4933	$(.30)$	3	3
1	0.866995	$(.32)10^{-1}$	0.7242	$(.86)10^{-1}$	0.7702	$(.33)$	19	3.75
2	0.860400	$(.61)10^{-3}$	0.6193	$(-.19)10^{-1}$	0.6285	$(.15)10^{-1}$	10	2.67
4	0.859782	$(.61)10^{-4}$	0.7214	$(-.60)10^{-2}$	0.5688	$(-.49)10^{-3}$	7	1.33
6	0.859689	$(.11)10^{-4}$	0.7348	$(-.27)10^{-2}$	0.5454	$(-.27)10^{-2}$	6	1
8	0.859667	$(.30)10^{-5}$	0.7426	$(-.12)10^{-2}$	0.5300	$(-.17)10^{-2}$	6	1
10	0.859663	$(.19)10^{-5}$	0.7447	$(-.11)10^{-2}$	0.5258	$(-.24)10^{-2}$	6	1
12	0.859661	$(.13)10^{-5}$	0.7461	$(-.89)10^{-3}$	0.5234	$(-.21)10^{-2}$	6	1
14	0.859660	$(.92)10^{-6}$	0.7472	$(-.75)10^{-3}$	0.5214	$(-.18)10^{-2}$	6	1
16	0.859659	$(.66)10^{-6}$	0.7482	$(-.64)10^{-3}$	0.5196	$(-.15)10^{-2}$	6	1
18	0.859658	$(.49)10^{-6}$	0.7491	$(-.55)10^{-3}$	0.5181	$(-.12)10^{-2}$	6	1
20	0.859658	$(.37)10^{-6}$	0.7499	$(-.48)10^{-3}$	0.5167	$(-.10)10^{-2}$	6	1
opt.	0.859655	0	0.7590	0	0.5000	0		

the analytical solution and denoted by 'opt.' are given below the values presented for the twentieth step. These first twenty steps required a total computer execution time of 23.45 seconds and 149 function evaluations where a function evaluation is still defined as one forward and one backward integration of the differential equations in blocks one and two of the flowchart in Fig. 2. The value of ϵ used to determine constraint satisfaction was $(.2)10^{-7}$.

Solution two was obtained in the same manner as solution one except for using nominal control two instead of nominal control one. The convergence properties of the first twenty steps of the proposed algorithm are shown in Table 7. These twenty steps required a total computer execution time of 21.62 seconds and 133 function evaluations.

Solution three was obtained in the same manner as solution two except that the conjugate gradient algorithm (5.55) is used in place of steepest descent. It is standard procedure in using a conjugate direction algorithm to restart the algorithm, i.e. perform a steepest descent step, after a specified number of steps. The algorithm used here was restarted after every six steps.

The convergence properties of the first twenty steps of the proposed algorithm are shown in Table 8. In addition, the control iterate at the twentieth step is plotted on Fig. 11. This skin thickness distribution is indistinguishable from the analytical solution on that scale. These twenty steps required a total computer execution time of 22.75 seconds and 138 function evaluations.

Tripathi and Narendra²⁵ have recently treated optimal control problems using a combination of the two techniques mentioned earlier in this

Table 7. Convergence of solution two of the rectangular wing problem: steepest descent from nominal control two.

Step number	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_1	\tilde{g}_{t_1}	$u(t_1^-)$	$\tilde{g}(t_1^-)$	NFEVAL	ANFECC
0	0.950000	(.43)	0.9000	(.47)	1.0000	(.87)	1	
0_{ψ}	0.951130	(.43)	0.9001	(.47)	1.0001	(.87)	2	2
1	0.860772	$(.37)10^{-2}$	0.7288	$(.58)10^{-2}$	0.6538	(.20)	12	3
2	0.859784	$(.25)10^{-3}$	0.7264	$(-.83)10^{-2}$	0.5504	$(-.23)10^{-1}$	9	2
4	0.859702	$(.29)10^{-4}$	0.7323	$(-.39)10^{-2}$	0.5449	$(-.12)10^{-1}$	6	1
6	0.859686	$(.16)10^{-4}$	0.7359	$(-.29)10^{-2}$	0.5390	$(-.10)10^{-1}$	6	1
8	0.859676	$(.99)10^{-5}$	0.7386	$(-.23)10^{-2}$	0.5346	$(-.87)10^{-2}$	6	1
10	0.859670	$(.64)10^{-5}$	0.7408	$(-.19)10^{-2}$	0.5312	$(-.74)10^{-2}$	6	1
12	0.859667	$(.43)10^{-5}$	0.7425	$(-.15)10^{-2}$	0.5284	$(-.63)10^{-2}$	6	1
14	0.859664	$(.30)10^{-5}$	0.7440	$(-.13)10^{-2}$	0.5260	$(-.54)10^{-2}$	6	1
16	0.859662	$(.22)10^{-5}$	0.7452	$(-.11)10^{-2}$	0.5240	$(-.46)10^{-2}$	6	1
18	0.859661	$(.16)10^{-5}$	0.7463	$(-.94)10^{-3}$	0.5223	$(-.40)10^{-2}$	6	1
20	0.859660	$(.12)10^{-5}$	0.7472	$(-.82)10^{-3}$	0.5208	$(-.34)10^{-2}$	6	1
opt.	0.859655	0	0.7590	0	0.5000	0		

Table 8. Convergence of solution three of the rectangular wing problem: conjugate gradient technique from nominal control two.

Step number	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_1	\tilde{g}_{t_1}	$u(t_1^-)$	$\tilde{g}(t_1^-)$	NFEVAL	ANFECC
0	0.950000	(.43)	0.9000	(.47)	1.0000	(.87)	1	
0 _ψ	0.951130	(.43)	0.9001	(.47)	1.0001	(.87)	2	2
1	0.860772	$(.38)10^{-2}$	0.7288	$(.58)10^{-2}$	0.6538	(.20)	12	3
2	0.859817	$(.36)10^{-3}$	0.7256	$(-.95)10^{-2}$	0.5521	$(-.21)10^{-1}$	9	2
4	0.859692	$(.40)10^{-4}$	0.7366	$(-.12)10^{-2}$	0.5509	$(.24)10^{-1}$	7	1.33
6	0.859658	$(.32)10^{-5}$	0.7503	$(-.73)10^{-4}$	0.5179	$(.44)10^{-2}$	7	1.33
8	0.859657	$(.55)10^{-6}$	0.7508	$(-.26)10^{-3}$	0.5167	$(.40)10^{-2}$	6	1
10	0.859656	$(.30)10^{-6}$	0.7559	$(-.11)10^{-3}$	0.5062	$(.85)10^{-3}$	6	1
12	0.859656	$(.28)10^{-7}$	0.7565	$(-.73)10^{-4}$	0.5049	$(.24)10^{-3}$	6	1
14	0.859656	$(.40)10^{-7}$	0.7568	$(-.11)10^{-3}$	0.5043	$(.36)10^{-3}$	6	1
16	0.859656	$(.18)10^{-7}$	0.7577	$(-.89)10^{-4}$	0.5024	$(-.44)10^{-4}$	6	1
18	0.859656	$(.14)10^{-7}$	0.7582	$(-.67)10^{-4}$	0.5017	$(.27)10^{-3}$	6	1
20	0.859656	$(.99)10^{-8}$	0.7582	$(-.79)10^{-4}$	0.5015	$(.78)10^{-4}$	6	1
opt.	0.859655	0	0.7590	0	0.5000	0		

chapter for handling unspecified final or mid-point times. They develop what they call a modified multiplier method and treat optimal control problems with unspecified terminal times using a penalty function to satisfy terminal state constraints. Their method uses a gradient vector which is a simplified version of Eq. (5.50) to determine the factor by which the time scale should be expanded or compressed as done by the linear time translations (4.4) or (4.5). No mathematical foundation is given which would authenticate such a combination of techniques which seems to violate both techniques discussed in this report. The rationale behind its use is evidently the numerical experience indicating that it indeed works. This is perhaps the best endorsement that any numerical technique could receive and justifies an attempt to use it here in conjunction with the projection operator technique.

Solution four was obtained in accordance with the Tripathi and Narendra²⁵ approach. That is, the gradient information for the steepest descent algorithm is defined by Eq. (5.50), the projection operator used with this gradient information is the one defined by Eq. (5.42), and the time point t_1 is varied by a compression or expansion of the time scale. Nominal solution one was used for this solution. The convergence properties of the first eighteen steps of the proposed algorithm are shown in Table 9. These eighteen steps required a total computer execution time of 30 seconds and 227 function evaluations. As should be expected from the combining of the two techniques, the ability of this new technique to correct a control back to the constraint surface is penalized. To lessen the degree of this penalty the value of ϵ used to determine constraint satisfaction was $(.2)10^{-6}$ rather than $(.2)10^{-7}$.

Table 9. Convergence of solution four of the rectangular wing problem: steepest descent from nominal control one.

Step number	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_1	\tilde{g}_{t_1}	$u(t_1^-)$	$\tilde{g}(t_1^-)$	NFEVAL	ANFECC
0	1.500000	(.16)	0.5000	$(-.89)10^{-1}$	1.5000	(.64)	1	
0_{ψ}	0.953571	(.13)	0.4618	(-.10)	1.4873	(.63)	3	3
1	0.864624	$(.13)10^{-1}$	0.6579	$(-.41)10^{-1}$	0.8184	(.31)	20	4
2	0.861523	$(.80)10^{-2}$	0.6956	$(-.14)10^{-1}$	0.5337	(-.33)	8	3
4	0.859918	$(.49)10^{-3}$	0.7147	$(-.13)10^{-1}$	0.5618	$(-.77)10^{-1}$	15	3.75
6	0.859751	$(.11)10^{-3}$	0.7255	$(-.75)10^{-2}$	0.5491	$(-.51)10^{-1}$	11	2.67
8	0.859705	$(.59)10^{-4}$	0.7323	$(-.48)10^{-2}$	0.5379	$(-.46)10^{-1}$	9	2
10	0.859681	$(.63)10^{-4}$	0.7388	$(-.25)10^{-2}$	0.5226	$(-.60)10^{-1}$	9	2
12	0.859666	$(.27)10^{-4}$	0.7441	$(-.14)10^{-2}$	0.5180	$(-.39)10^{-1}$	10	2
14	0.859661	$(.21)10^{-4}$	0.7472	$(-.85)10^{-3}$	0.5132	$(-.36)10^{-1}$	10	2
16	0.859658	$(.12)10^{-4}$	0.7510	$(-.38)10^{-3}$	0.5082	$(-.27)10^{-1}$	13	2
18	0.859656	$(.44)10^{-6}$	0.7514	$(-.25)10^{-3}$	0.5132	$(-.49)10^{-2}$	14	1.33
opt.	0.859655	0	0.7590	0	0.5000	0		

Solution five was obtained in exactly the same manner as solution four with the exception of starting from nominal control two. The convergence properties of the first eighteen steps of the proposed algorithm are shown in Table 10. These eighteen steps required a total computer execution time of 30 seconds and 230 function evaluations.

Panel flutter problem

Consider the initially flat panel of infinite span and length L shown in Fig. 12. The panel is assumed simply supported and has one side exposed to a parallel, high Mach number supersonic flow. The problem of minimizing the weight of a panel of sandwich construction constrained to a specific flutter speed has been treated numerically by McIntosh, Weisshaar, and Ashley³⁶ and Weisshaar.³⁷ The same problem with a panel of solid construction rather than sandwich construction has been numerically treated by Armand and Vitte³⁵ and Pierson.³⁸

Neglecting midplane stress and aerodynamic damping and assuming elastic bending and simple harmonic motion, the 4th-order ordinary non-dimensional differential equation governing this system is

$$[\delta(x)w''(x)]'' + \lambda_0 w'(x) - (\alpha\pi)^4 \mu(x)w(x) = 0 \quad (5.56)$$

where $x = X/L$, $X =$ distance along the panel,

$w = W/L$, $W =$ panel deflection,

$\delta(x) = D(x)/D_0$, $D =$ panel stiffness,

$D_0 =$ constant stiffness of reference panel,

$\alpha = \omega/\omega_f$, $\omega =$ fundamental flutter frequency,

Table 10. Convergence of solution five of the rectangular wing problem: steepest descent from nominal control two.

Step number	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_1	\tilde{g}_{t_1}	$u(t_1^-)$	$\tilde{g}(t_1^-)$	NFEVAL	ANFECC
0	0.950000	(.45)	0.9000	(.47)	1.0000	(.97)	1	
0 _ψ	0.951127	(.45)	0.9001	(.47)	1.0001	(.97)	2	2
1	0.861144	$(.40)10^{-2}$	0.6927	$(-.25)10^{-1}$	0.6734	(.15)	15	4
2	0.860065	$(.59)10^{-3}$	0.7049	$(-.20)10^{-1}$	0.5948	$(-.21)10^{-1}$	12	3
4	0.859744	$(.75)10^{-4}$	0.7259	$(-.76)10^{-2}$	0.5560	$(-.22)10^{-1}$	11	2.67
6	0.859697	$(.33)10^{-4}$	0.7335	$(-.45)10^{-2}$	0.5410	$(-.26)10^{-1}$	9	2
8	0.859680	$(.23)10^{-4}$	0.7378	$(-.31)10^{-2}$	0.5323	$(-.28)10^{-1}$	9	2
10	0.859670	$(.30)10^{-4}$	0.7423	$(-.18)10^{-2}$	0.5208	$(-.41)10^{-1}$	9	2
12	0.859662	$(.14)10^{-4}$	0.7460	$(-.11)10^{-2}$	0.5172	$(-.28)10^{-1}$	10	2
14	0.859660	$(.12)10^{-4}$	0.7484	$(-.72)10^{-3}$	0.5130	$(-.28)10^{-1}$	10	2
16	0.859658	$(.99)10^{-5}$	0.7515	$(-.34)10^{-3}$	0.5079	$(-.24)10^{-1}$	13	2
18	0.859656	$(.20)10^{-6}$	0.7518	$(-.25)10^{-3}$	0.5129	$(-.28)10^{-2}$	18	1.25
opt.	0.859655	0	0.7590	0	0.5000	0		

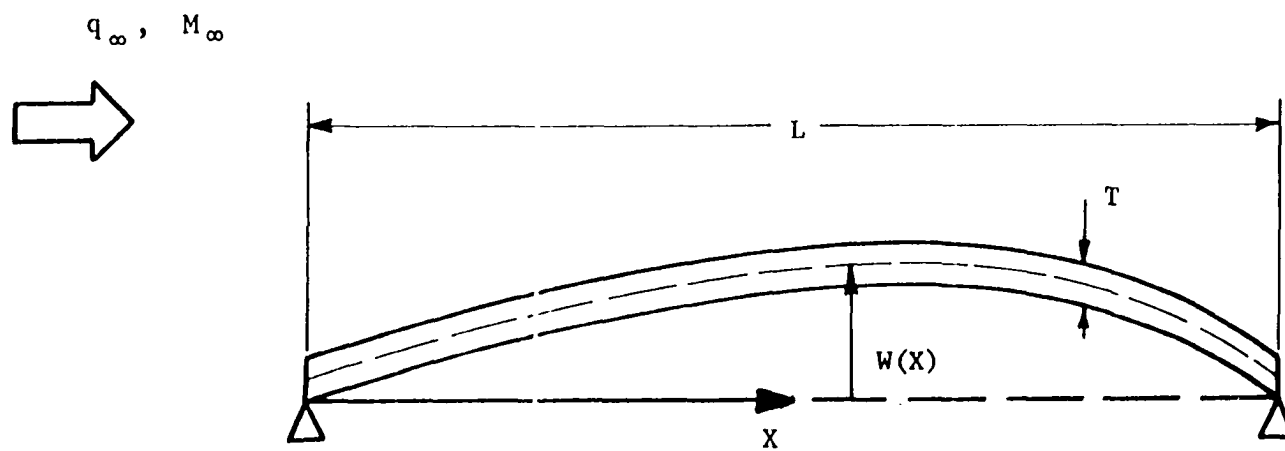


Fig. 12. Simply supported panel of infinite span in supersonic flow.

ω_f = fundamental frequency of reference panel
for free vibration,

$$\mu(x) = m(x)/m_o,$$

$m(x)$ = mass per unit length

m_o = mass per unit length of reference
panel,

and
$$\lambda_o = \frac{2q L^3}{D_o \sqrt{M^2 - 1}},$$

q = dynamic pressure,

M = Mach number.

The notation $()'$ again represents differentiation with respect to x . The 4th-order differential equation is subject to the boundary conditions

$$w(0) = w(1) = 0,$$

and
$$[\delta(x)w''(x)]_{x=0} = [\delta(x)w''(x)]_{x=1} = 0.$$

The first states that there is zero deflection of the panel at either end, while the second indicates that there is no moment applied at either end or that the structure is simply supported.

The values of λ_o and α for the reference panel of uniform thickness are³⁸ 343.36 and 1.8127, respectively. From the definition of λ_o it can be seen that λ_o characterizes the flutter speed of the panel. Fixing λ_o constitutes a dynamic constraint on the problem. Throughout the minimization process, λ_o must be held fixed while allowing α and the panel thickness to vary. Although the projection operator could be revised to treat α as a control parameter, this investigation fixes it at the uniform panel value to simplify the problem solution. Since other numerical experience^{37,38} indicates that the optimal value of α and the uniform

panel value of α are not very different, this simplification should not greatly penalize the result of the minimizing technique.

If $t(x)$ is defined as the ratio between the panel thickness and the uniform thickness of the reference panel, $\delta(x)$ for the solid panel equals $t^3(x)$ and $\mu(x)$ for the solid panel equals $t(x)$. In addition, after defining the new variables

$$\begin{aligned} t &= x, \\ u(t) &= t(x), \\ x_1(t) &= w(x), \\ x_2(t) &= w'(x), \\ x_3(t) &= t^3(x) w''(x), \\ \text{and} \quad x_4(t) &= [t^3(x) w''(x)]', \end{aligned}$$

the problem of minimizing the mass of the panel with a minimum thickness constraint on the skin thickness can be stated as:

$$\text{Minimize} \quad J = \int_0^1 u(t) \, dt$$

subject to the 4th-order dynamical system

$$\dot{x}_1(t) = x_2(t), \quad x_1(0) = x_1(1) = 0,$$

$$\dot{x}_2(t) = x_3(t)/u^3(t),$$

$$\dot{x}_3(t) = x_4(t), \quad x_3(0) = x_3(1) = 0,$$

$$\dot{x}_4(t) = -\lambda_0 x_2(t) + (\alpha \pi)^4 u(t) x_1(t), \quad \lambda_0 = 343.36, \alpha = 1.8127,$$

and subject to the control inequality constraint

$$u(t) - u_{\min} \geq 0.$$

Although no one has been able to obtain an analytical solution for this optimal control problem, Armand and Vitte³⁵ have shown the optimal control to be symmetrical about the mid-chord ($t=\frac{1}{2}$). Weisshaar³⁷ has obtained a numerical solution for the sandwich construction panel with a 0.5 minimum thickness constraint. His solution contains three constrained arcs, one at either end and one in the middle. Taking this information into consideration the skin thickness or rather the time-varying portion of the control ω chosen as the nominal skin thickness is shown in Fig. 13. The time points t_1 through t_4 equal 0.5, 0.45, 0.55, and 0.95, respectively, for the nominal control time history.

The optimal control problem as stated requires the optimization process to account for the two unspecified initial conditions on state variables x_2 and x_4 . This requires again that the material from Chapter 3 concerning missing initial conditions on state variables be utilized. The nominal values of $x_2(0)$ and $x_4(0)$ used in this investigation were 0.0 and 0.1. More information concerning this topic is given with the discussion of the particular trial solution.

This example problem utilizes the constrained one-dimensional minimization for the same purpose as discussed for the first example problem in this chapter. In addition, it is also used to constrain the motion of the skin thickness along either unconstrained arc by not more than 0.1 on any step. This constraint is active only for the first few steps of

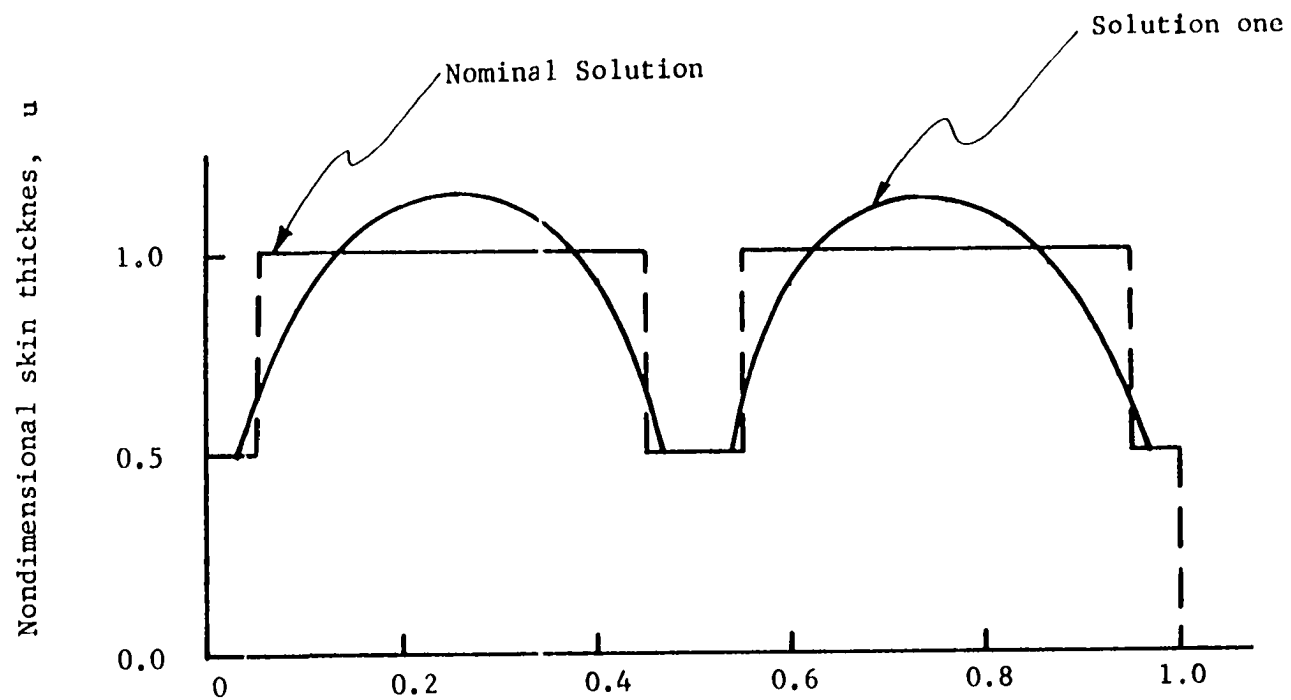


Fig. 13. Nominal and optimal solutions of panel flutter problem.

any of the trial solutions and was imposed to reduce the amount of function evaluations required to correct the controls back to the constraint surface during the one-dimensional minimization.

Six solutions were attempted for this optimal control problem. They are denoted as solution one, solution two, etc. The results of these six trial solutions are presented in tabular form in Tables 11 through 17. The twelfth iterate of solution one is presented in Fig. 13 and labeled as 'optimal'. Tables 11 through 16 do not in general include values of the initial conditions on state variables x_2 and x_4 since they remain essentially constant after the first few steps. The value of ϵ used to determine constraint satisfaction was $(0.5)10^{-6}$. The techniques used to obtain trial solutions one through six are described below.

Solution one was obtained using the second projection operator derived in this chapter letting the contact times t_1 through t_4 play the role of control parameters and then using the method proposed by Tripathi and Narendra²⁵ to implement the contact time changes. Although the proposed algorithm was unable to decrease the value of the cost functional J below the value obtained by the first twelve steps given in Table 11, the twelfth iterate of this trial solution does represent one of the two best approximations to the actual optimal solution. These first twelve steps of solution one required a total computer execution time of 58.26 seconds and required 173 function evaluations.

Since the state variables evaluated along the optimal control are not unique for this particular optimal control problem, it would possibly be advantageous to fix either $x_2(0)$ or $x_4(0)$ thereby requiring the state

Table 11. Convergence of solution one of the panel flutter problem.

Step number	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_1	\tilde{g}_{t_1}	$u(t_1^+)$	$\tilde{g}(t_1^-)$	NFEVAL	ANFECC
0	0.900000	$(.15)10^{+1}$	0.0500	$(-.30)$	1.0000	$(.91)$	1	
0	0.933332	$(.31)$	0.0369	$(-.28)$	1.0056	$(.91)$	5	5
1	0.916424	$(.57)10^{-1}$	0.0588	$(.98)10^{-1}$	0.9115	$(.69)$	10	3.5
2	0.911119	$(.43)10^{-1}$	0.0456	$(-.33)10^{-1}$	0.8193	$(.70)$	11	4
3	0.907554	$(.28)10^{-1}$	0.0502	$(.93)10^{-1}$	0.7195	$(.36)$	13	5
4	0.905840	$(.22)10^{-1}$	0.0340	$(-.36)10^{-1}$	0.6561	$(.57)$	15	6
5	0.904746	$(.13)10^{-1}$	0.0404	$(.31)10^{-1}$	0.5551	$(-.26)$	15	6
6	0.903637	$(.32)10^{-2}$	0.0364	$(.11)10^{-1}$	0.5884	$(.20)$	17	4.67
7	0.903325	$(.14)10^{-2}$	0.0333	$(.44)10^{-2}$	0.5215	$(-.11)$	20	5.67
8	0.903208	$(.37)10^{-3}$	0.0325	$(.27)10^{-2}$	0.5406	$(.92)10^{-1}$	17	4.67
9	0.903187	$(.16)10^{-3}$	0.0320	$(.24)10^{-2}$	0.5202	$(-.30)10^{-1}$	16	4.33
10	0.903185	$(.18)10^{-3}$	0.0310	$(.32)10^{-3}$	0.5309	$(.11)$	11	4
11	0.903177	$(.13)10^{-4}$	0.0310	$(.11)10^{-2}$	0.5176	$(-.11)10^{-1}$	12	3
12	0.903177	$(.78)10^{-5}$	0.0307	$(.82)10^{-3}$	0.5144	$(.25)10^{-2}$	11	4

Table 11. Continued.

Step no.	t_2	\tilde{g}_{t_2}	$u(t_2^-)$	$\tilde{g}(t_2^-)$	t_3	\tilde{g}_{t_3}	$u(t_3^+)$	$\tilde{g}(t_3^+)$
0	.4500	(.40)	1.0000	(.79)	.5500	(-.25)	1.0000	(.82)
0 _{ψ}	.4600	(.12)	1.0157	(.54)	.5337	(-.26)	1.0136	(.60)
1	.4523	(.11) 10^{-1}	0.9628	(.50)	.5545	(.10) 10^{-1}	0.9551	(.49)
2	.4511	(-.51) 10^{-1}	0.8970	(.42)	.5529	(.16) 10^{-1}	0.8901	(.45)
3	.4585	(.13) 10^{-1}	0.8368	(.43)	.5506	(.26) 10^{-1}	0.8258	(.39)
4	.4557	(-.55) 10^{-1}	0.7596	(.28)	.5463	(.66) 10^{-2}	0.7556	(.36)
5	.4654	(.23) 10^{-1}	0.7100	(.37)	.5453	(.22) 10^{-1}	0.6920	(.25)
6	.4622	(-.20) 10^{-1}	0.6612	(.20)	.5426	(.11) 10^{-1}	0.6588	(.23)
7	.4678	(-.12) 10^{-2}	0.5955	(.19)	.5396	(.54) 10^{-2}	0.5838	(.12)
8	.4676	(-.55) 10^{-2}	0.5593	(.51) 10^{-1}	.5387	(.43) 10^{-2}	0.5609	(.72) 10^{-1}
9	.4686	(-.25) 10^{-2}	0.5475	(.58) 10^{-1}	.5378	(.29) 10^{-2}	0.5448	(.42) 10^{-1}
10	.4696	(-.13) 10^{-2}	0.5218	(.62) 10^{-2}	.5367	(.11) 10^{-2}	0.5255	(.18) 10^{-1}
11	.4698	(-.99) 10^{-3}	0.5223	(.10) 10^{-1}	.5365	(.11) 10^{-2}	0.5231	(.10) 10^{-1}
12	.4700	(-.67) 10^{-3}	0.5193	(.12) 10^{-1}	.5363	(.83) 10^{-3}	0.5200	(.82) 10^{-2}

Table 11. Continued.

Step no.	t_4	\tilde{g}_{t_4}	$u(t_4^-)$	$\tilde{g}(t_4^-)$	$x_2(0)$	\tilde{g}_{x_2}	$x_4(0)$	\tilde{g}_{x_4}
0	.9500	(.77)	1.0000	(.11) 10^{+1}	.00000	(.24) 10^{-1}	0.1000	(.16)
0 _y	.9417	(-.58) 10^{-1}	0.9960	(.79)	-.00135	(.57) 10^{-2}	0.0902	(.90) 10^{-4}
1	.9464	(.50) 10^{-2}	0.9079	(.75)	-.00208	(-.44) 10^{-2}	0.0874	(-.11) 10^{-3}
2	.9460	(-.76) 10^{-1}	0.8082	(.58)	-.00150	(.30) 10^{-2}	0.0874	(.52) 10^{-4}
3	.9568	(.56) 10^{-2}	0.7260	(.57)	-.00208	(-.34) 10^{-2}	0.0874	(-.82) 10^{-4}
4	.9556	(-.51) 10^{-1}	0.6248	(.12)	-.00132	(.27) 10^{-2}	0.0874	(.42) 10^{-4}
5	.9646	(.55) 10^{-2}	0.6036	(.36)	-.00217	(-.28) 10^{-2}	0.0873	(-.68) 10^{-4}
6	.9639	(-.11) 10^{-1}	0.5560	(.40) 10^{-1}	-.00169	(.46) 10^{-3}	0.0874	(.12) 10^{-4}
7	.9669	(-.24) 10^{-2}	0.5381	(.85) 10^{-1}	-.00185	(-.67) 10^{-3}	0.0870	(-.13) 10^{-4}
8	.9674	(-.24) 10^{-2}	0.5216	(-.20) 10^{-1}	-.00171	(.22) 10^{-3}	0.0870	(.56) 10^{-5}
9	.9678	(-.16) 10^{-2}	0.5244	(.31) 10^{-1}	-.00176	(-.13) 10^{-3}	0.0870	(-.13) 10^{-5}
10	.9684	(-.83) 10^{-3}	0.5093	(-.49) 10^{-1}	-.00170	(.26) 10^{-3}	0.0870	(.62) 10^{-5}
11	.9686	(-.86) 10^{-3}	0.5151	(.53) 10^{-2}	-.00174	(-.23) 10^{-4}	0.0870	(.20) 10^{-5}
12	.9688	(-.68) 10^{-3}	0.5133	(.39) 10^{-2}	-.00174	(-.92) 10^{-7}	0.0870	(.13) 10^{-5}

Table 12. Convergence of solution two of the panel flutter problem.

Step no.	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_3	t_4	$u(t_3^+)$	$\tilde{g}(t_3^+)$	$u(t_4^-)$	$\tilde{g}(t_4^-)$	NFEVAL
0	0.900000	(.73)	0.5500	0.9500	1.0000	(.42)	1.0000	(.39)	1
0_{ψ}	0.931122	(.34)	0.5492	0.9628	1.0089	(.22)	1.0020	(.38)	14
1	0.927551	(.12)	0.5378	0.9467	1.0035	(.25)	0.9917	(.34)	8
2	0.925028	(.20)	0.5496	0.9553	0.9934	(.20)	0.9766	(.35)	6
4	0.921239	(.12)	0.5514	0.9522	0.9765	(.18)	0.9502	(.33)	7
8	0.916483	(.30)	0.5570	0.9560	0.9354	(.19)	0.8869	(.33)	11
12	0.911480	$(.26)10^{-1}$	0.5519	0.9501	0.8935	(.16)	0.8235	(.25)	17
16	0.910309	$(.18)10^{-1}$	0.5513	0.9502	0.8766	(.16)	0.7996	(.24)	13
20	0.909314	$(.14)10^{-1}$	0.5509	0.9508	0.8601	(.15)	0.7770	(.22)	13
24	0.908445	$(.12)10^{-1}$	0.5505	0.9515	0.8436	(.14)	0.7553	(.20)	22
28	0.907712	$(.10)10^{-1}$	0.5501	0.9522	0.8280	(.14)	0.7356	(.19)	12
32	0.907088	$(.17)10^{-1}$	0.5504	0.9539	0.8117	(.13)	0.7162	(.18)	9
36	0.906612	$(.15)10^{-1}$	0.5499	0.9544	0.7989	(.12)	0.7016	(.16)	14
40	0.906197	$(.11)10^{-1}$	0.5493	0.9548	0.7868	(.12)	0.6884	(.15)	8
44	0.905854	$(.73)10^{-2}$	0.5487	0.9551	0.7758	(.12)	0.6770	(.14)	7
48	0.905533	$(.57)10^{-2}$	0.5481	0.9556	0.7642	(.11)	0.6655	(.12)	7

Table 13. Convergence of solution three of the panel flutter problem.

Step no.	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_3	t_4	$u(t_3^+)$	$\tilde{g}(t_3^+)$	$u(t_4^-)$	$\tilde{g}(t_4^-)$	NFEVAL
0	0.900000	(.74)	0.5500	0.9500	1.0000	(.42)	1.0000	(.39)	1
0 _{1/2}	0.930260	(.18)	0.5456	0.9586	1.0090	(.20)	1.0019	(.36)	12
1	0.926799	(.17)	0.5351	0.9418	1.0003	(.28)	0.9852	(.34)	8
2	0.924118	(.14)	0.5487	0.9533	0.9911	(.19)	0.9728	(.34)	6
4	0.921258	(.19)	0.5519	0.9537	0.9752	(.19)	0.9481	(.33)	8
8	0.916012	(.31)	0.5573	0.9562	0.9306	(.19)	0.8796	(.33)	11
12	0.909806	(.33)	0.5584	0.9604	0.8426	(.17)	0.7508	(.29)	14
16	0.907588	$(.20)10^{-1}$	0.5511	0.9533	0.8268	(.13)	0.7303	(.19)	10
20	0.907041	$(.15)10^{-1}$	0.5505	0.9537	0.8134	(.13)	0.7145	(.17)	11
24	0.906579	$(.13)10^{-1}$	0.5499	0.9543	0.8008	(.12)	0.7003	(.16)	8
28	0.906192	$(.97)10^{-2}$	0.5494	0.9547	0.7891	(.12)	0.6878	(.15)	8
32	0.905846	$(.65)10^{-2}$	0.5487	0.9550	0.7779	(.12)	0.6762	(.13)	7
36	0.905520	$(.55)10^{-2}$	0.5482	0.9557	0.7658	(.11)	0.6645	(.12)	7
40	0.905235	$(.47)10^{-2}$	0.5477	0.9562	0.7541	(.11)	0.6537	(.11)	7
44	0.904985	$(.40)10^{-2}$	0.5473	0.9568	0.7428	(.11)	0.6438	(.10)	7
48	0.904769	$(.35)10^{-2}$	0.5468	0.9574	0.7320	(.10)	0.6350	$(.90)10^{-1}$	7

Table 14. Convergence of solution four of the panel flutter problem.

Step no.	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_3	t_4	$u(t_3^+)$	$\tilde{g}(t_3^+)$	$u(t_4^-)$	$\tilde{g}(t_4^-)$	NFEVAL
0	0.900000	(.81)	0.5500	0.9500	1.0000	(.88)	1.0000	(.11) 10^{+1}	1
0 ψ	0.927076	(.10)	0.5451	0.9424	1.0264	(.59)	0.9919	(.81)	6
1	0.916873	(.56) 10^{-1}	0.5486	0.9461	0.9591	(.50)	0.8983	(.72)	6
2	0.910811	(.28) 10^{-1}	0.5504	0.9481	0.8949	(.42)	0.8059	(.57)	6
4	0.905847	(.21) 10^{-2}	0.5536	0.9548	0.7347	(.13)	0.6147	(-.64) 10^{-2}	7
8	0.905270	(.30) 10^{-3}	0.5541	0.9615	0.6902	(.16) 10^{-1}	0.5639	(-.39) 10^{-1}	8
12	0.905141	(.21) 10^{-3}	0.5535	0.9632	0.6859	(.13) 10^{-1}	0.5547	(-.21) 10^{-1}	7
16	0.905031	(.97) 10^{-3}	0.5521	0.9638	0.6833	(.37) 10^{-1}	0.5374	(-.13)	10
20	0.904897	(.43) 10^{-3}	0.5514	0.9649	0.6784	(.23) 10^{-1}	0.5353	(-.81) 10^{-1}	7
24	0.904812	(.26) 10^{-3}	0.5507	0.9655	0.6753	(.17) 10^{-1}	0.5346	(-.53) 10^{-1}	7
28	0.904719	(.95) 10^{-3}	0.5492	0.9656	0.6724	(.35) 10^{-1}	0.5222	(-.16)	8
32	0.904612	(.31) 10^{-3}	0.5485	0.9663	0.6684	(.20) 10^{-1}	0.5270	(-.72) 10^{-1}	7
36	0.904558	(.71) 10^{-3}	0.5475	0.9663	0.6667	(.32) 10^{-1}	0.5213	(-.12)	6
40	0.904470	(.25) 10^{-3}	0.5469	0.9669	0.6629	(.18) 10^{-1}	0.5239	(-.64) 10^{-1}	7
44	0.904416	(.14) 10^{-3}	0.5463	0.9673	0.6605	(.12) 10^{-1}	0.5249	(-.36) 10^{-1}	6
48	0.904362	(.96) 10^{-4}	0.5457	0.9676	0.6580	(.83) 10^{-2}	0.5248	(-.19) 10^{-1}	9

Table 15. Convergence of solution five of the panel flutter problem.

Step no.	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_3	t_4	$u(t_3^+)$	$\tilde{g}(t_3^+)$	$u(t_4^-)$	$\tilde{g}(t_4^-)$	NFEVAL
0	0.900000	(.81)	0.5500	0.9500	1.0000	(.88)	1.0000	(.11) 10^{+1}	1
0 _{ψ}	0.927076	(.10)	0.5451	0.9424	1.0264	(.59)	0.9919	(.81)	6
1	0.916873	(.56) 10^{-1}	0.5486	0.9461	0.9591	(.50)	0.8983	(.72)	6
2	0.910652	(.27) 10^{-1}	0.5512	0.9489	0.8948	(.41)	0.8062	(.59)	7
3	0.907347	(.94) 10^{-2}	0.5527	0.9511	0.8307	(.32)	0.7138	(.36)	8
4	0.906213	(.47) 10^{-2}	0.5535	0.9533	0.7559	(.18)	0.6175	(-.42) 10^{-1}	8
5	0.905990	(.56) 10^{-2}	0.5546	0.9575	0.6764	(-.60) 10^{-1}	0.5630	(-.26)	8
6	0.905911	(.45) 10^{-2}	0.5551	0.9596	0.6522	(-.18)	0.5576	(-.19)	10
7	0.905438	(.90) 10^{-3}	0.5551	0.9610	0.6935	(-.59) 10^{-2}	0.6007	(.17)	13
8	0.905343	(.58) 10^{-3}	0.5544	0.9610	0.7044	(.64) 10^{-1}	0.5666	(-.49) 10^{-1}	9
9	0.905273	(.67) 10^{-3}	0.5549	0.9633	0.6954	(.22) 10^{-1}	0.5568	(.42) 10^{-2}	10
10	0.904923	(.17) 10^{-2}	0.5522	0.9708	0.6287	(-.14)	0.5046	(.67) 10^{-1}	7
11	0.904892	(.15) 10^{-2}	0.5521	0.9713	0.6285	(-.14)	0.5006	(.68) 10^{-1}	6
12	0.904886	(.14) 10^{-2}	0.5521	0.9714	0.6288	(-.14)	0.5001	(.68) 10^{-1}	5

Table 16. Convergence of solution six of the panel flutter problem.

Step no.	J	$\langle \tilde{g}, \tilde{g} \rangle$	t_3	t_4	$u(t_3^+)$	$\tilde{g}(t_3^+)$	$u(t_4^-)$	$\tilde{g}(t_4^-)$	NFEVAL
0	0.900000	(.57)	0.5500	0.9500	1.0000	(.45)	1.0000	(.36)	1
0_{ψ}	0.937342	(.21)	0.5324	0.9500	1.0046	(.30)	1.0025	(.38)	4
1	0.930496	(.13)	0.5412	0.9527	0.9948	(.24)	0.9900	(.36)	6
2	0.924639	$(.65)10^{-1}$	0.5466	0.9496	0.9829	(.21)	0.9718	(.33)	6
6	0.911655	$(.39)10^{-1}$	0.5485	0.9423	0.8896	(.22)	0.8262	(.25)	9
12	0.908184	$(.32)10^{-1}$	0.5473	0.9456	0.8292	(.20)	0.7426	(.19)	9
18	0.905447	$(.26)10^{-1}$	0.5449	0.9507	0.7476	(.18)	0.6504	$(.89)10^{-1}$	11
24	0.904167	$(.27)10^{-2}$	0.5463	0.9574	0.6933	(.10)	0.6072	$(.49)10^{-1}$	7
30	0.903873	$(.33)10^{-2}$	0.5448	0.9586	0.6683	$(.95)10^{-1}$	0.5913	$(.32)10^{-1}$	6
36	0.903527	$(.53)10^{-2}$	0.5418	0.9607	0.6245	$(.84)10^{-1}$	0.5655	$(.56)10^{-2}$	13
42	0.903209	$(.15)10^{-3}$	0.5396	0.9673	0.5622	$(.81)10^{-2}$	0.5294	$(.15)10^{-1}$	6
48	0.903201	$(.64)10^{-4}$	0.5391	0.9674	0.5589	$(.13)10^{-1}$	0.5267	$(.12)10^{-1}$	6
54	0.903179	$(.37)10^{-4}$	0.5372	0.9690	0.5395	$(.13)10^{-1}$	0.5130	$(.76)10^{-2}$	15
60	0.903175	$(.90)10^{-4}$	0.5362	0.9705	0.5255	$(.52)10^{-2}$	0.5033	$(.12)10^{-1}$	14
66	0.903173	$(.30)10^{-4}$	0.5356	0.9707	0.5213	$(.11)10^{-1}$	0.5000	$(.85)10^{-2}$	5

variables to be unique. Trial solution two is obtained by fixing the value of $x_4(0)$ at 0.1 and using the linear time transformation (4.4) with the first projection operator developed in this chapter. Some pertinent results of the first 48 steps of solution two are presented in Table 12. These 48 steps required a total computer execution time of 172 seconds and required 503 function evaluations.

Trial solution three was obtained in a similar manner to solution two except for allowing the value of $x_4(0)$ to vary. Table 13 contains the results of the first 48 steps of this solution. These 48 steps of solution three required a total computer execution time of 154 seconds and required 460 function evaluations.

Trial solution four was obtained by again allowing both $x_2(0)$ and $x_4(0)$ to vary and using the linear time transformation (4.5) instead of the linear time transformation (4.4) as in the two previous trial solutions. Table 14 contains the results of the first 48 steps of this solution. These 48 steps of solution 4 required a total computer execution time of 120 seconds and required 367 function evaluations.

Trial solution five was obtained in a similar manner to solution four except for the method of picking the direction of search. Solution four, as well as all previous solutions, have used the steepest descent direction of search, while solution five used the conjugate gradient technique described in this chapter for picking the direction of search. The algorithm was again restarted after every six steps. The results of the first twelve steps of this trial solution are presented in Table 15. As can be seen from Table 15, these twelve steps force the control at t_4^- ,

$u(t_4^-)$, very close to the control constraint. Even though the value of the control at t_3^+ , $u(t_3^+)$, is greater than 0.5, the algorithm can no longer proceed since the constrained one-dimensional minimization is controlled by the value of the control and direction of search at t_4^- . These twelve steps of solution five required a total execution time of 33.49 seconds and required 104 function evaluations.

One well-known method of increasing numerical accuracy is the practice of nondimensionalizing the state variables. This is possible if the 4th-order ordinary differential equation (5.56) is written as

$$\beta \left(\frac{t^3(x) w''(x)}{\beta} \right)' + \lambda_0 w'(x) - (\alpha \pi)^4 t(x) w(x) = 0$$

where β is an arbitrary scaling constant. Making the definitions analogous to those in the earlier development,

$$t = x,$$

$$u(t) = t(x),$$

$$x_1(t) = w(x),$$

$$x_2(t) = w'(x),$$

$$x_3(t) = t^3(x) w''(x) / \beta,$$

$$\text{and} \quad x_4(t) = [t^3(x) w''(x) / \beta]',$$

the dynamical system becomes

$$\dot{x}_1(t) = x_2(t), \quad x_1(0) = x_1(1) = 0,$$

$$\dot{x}_2(t) = \beta x_3(t) / u^3(t),$$

$$\dot{x}_3(t) = x_4(t), \quad x_3(0) = x_3(1) = 0,$$

$$\dot{x}_4(t) = -\lambda_0 x_2(t)/\beta + (\alpha\pi)^4 u(t)x_1(t)/\beta, \quad \lambda_0 = 343.36, \alpha = 1.8127.$$

Specifying the value of β to equal 50 causes the state variables x_1 and x_3 to take on the same order of magnitude. Consequently so do state variables x_2 and x_4 . Solution six is obtained by using the projection operator derived for linear time transformation (4.4) with the newly defined set of state variables and the steepest descent technique for picking the direction of search. The results of the first sixty-six steps of solution six are presented in Table 16. These sixty-six steps of solution six required a total computer execution time of 198 seconds and required 616 function evaluations.

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY

The task of evaluating the results or drawing conclusions from any numerical study is not an easy one. The attempt to be completely objective is immediately hampered by necessarily incomplete data. Especially in the field of numerical optimization, where the success or failure of any numerical algorithm is so highly problem dependent, the degree of objectivity can be only as valid as the numerical experience gained during the study. Inferences made about a particular problem as a result of limited numerical experience can often be confusing and misleading. The initial testing of any numerical technique is best done using simple example problems for practical considerations.

The evaluation of the results of Chapter 3 is greatly facilitated by the availability of analytical solutions for the two example problems. There is no doubt that the agreement between the analytical and numerical solutions for both problems is quite good. Since no previous numerical results for the harmonically forced rod problem are available, it is difficult to measure the computational effort required by the algorithm to solve the problem. The performance of any algorithm is based not only on its ability to obtain correct answers but also on the computational effort required to obtain those answers. For that reason, any evaluation of the proposed algorithm based solely on this first example problem is somewhat incomplete. It is fair to say, however, that the computer time required for the three solutions to this first example problem does not appear to be unduely high.

The rocket launch problem with unspecified launch site has an optimal control time history identical to the rocket launch problem with specified launch site. Fortunately, this latter problem has been given much attention.^{8,9,21,30} Lasdon, Mitter, and Waren⁸ use the penalty function approach to satisfy terminal state constraints and a conjugate gradient algorithm to determine the direction of search. Due to oscillations of the gradient, which were attributed to the large penalty constant needed to satisfy the terminal state constraints, a converged solution was not obtained.

Pierson³⁰ has obtained a solution for the rocket launch problem using a discrete-variable technique which requires that the dynamical system be replaced by a set of appropriate difference equations. The control time history obtained by Pierson is comparable to that obtained in this investigation. However, his solution required many times the computer time required by this investigation. Since the discrete-variable technique was programmed on a somewhat slower computer (IBM 7090) the relative time requirements lose significance. Since the comparable discrete-variable solution required on the order of six minutes, the 18.0 seconds required in this investigation using the projection operator technique does give a favorable recommendation to the use of the projection operator technique.

Pierson and Rajtora⁹ have obtained three solutions to the rocket launch problem using the penalty function approach to satisfy terminal state constraints and the Davidon technique to determine the direction of search. The three different solutions were obtained by restarting the Davidon algorithm after every three, four, and six steps. The best results

and the shortest computation time (21 seconds) were achieved by restarting the algorithm after every three steps. Since identical computers were used in both investigations, there is at least some capability for a comparison of results. Difficulties are present, however, since Pierson and Rajtora divided the time interval into 100 segments for numerical integration purposes while in this investigation the time interval was divided into only 50 segments. If the Davidon technique had been tested using only half as many intervals for the numerical integration, the computation time would probably be less. It is also possible, though, that the numerical sensitivity present as a result of the rather large penalty constants would increase the computation time rather than decrease it. The value of the penalty constant used by Pierson and Rajtora was the same value as that used by Lasdon, Mitter, and Waren⁸ but still not large enough to satisfy the terminal boundary constraints to the same extent possible with the projection technique. Raising the penalty constant would increase the complexity of the problem for the Davidon technique and as a result would require larger computation times. Although the difference in computation times is not large enough to make any final judgment concerning the comparative capability of the penalty function approach using the Davidon technique versus the projection operator technique using steepest descent, the proposed projection algorithm is at least competitive with the former technique.

The Rajtora and Pierson²¹ solution of the rocket launch problem is very similar to the solution of the variable launch site rocket launch problem solved in this investigation. The only differences are the number

of integration steps used and, of course, the changing of the initial conditions in the one case and not in the other. For the fixed launch site and 100 integration segments, the required computation time was 19.2 seconds, while for the variable launch site and 50 integration segments the required computation time was 18.0 seconds. These figures indicate that the variable launch site problem is probably a slightly more difficult problem than the fixed launch site problem. The additional computational effort appears to be fairly modest considering that the variable launch site problem has three terminal state constraints as opposed to only two for the fixed launch site problem.

Referring to Tables 2 and 3 it can be seen that the greatest computational effort required to correct the controls back to the constraint surface during the one-dimensional minimization occurred during the first few steps. Rather obviously, the further the nominal guess departs from the optimal solution the greater this effect is on the total computational effort required to solve the problem. This is perhaps a partial justification for the use of a constrained one-dimensional minimization as implemented in Chapter 5 of this investigation.

Although the two example problems presented in Chapter 3 were meant to be of a simple and straightforward nature, the very satisfactory results obtained by the proposed algorithm certainly give a favorable outlook to future implementation. Additional numerical experience using the proposed algorithm in connection with unspecified initial conditions is presented in Chapter 5. The results obtained there give further encouragement and optimism for the concept of the one-dimensional minimization along the constraint surface for problems involving unspecified initial conditions.

As stated earlier in Chapter 3, both Mehra and Bryson¹³ and Tripathi and Narendra²⁵ have solved the same V/STOL minimum time-to-climb optimization problem involving an unknown initial flight path angle and have obtained surprisingly different results. Mehra and Bryson concluded that the minimum time to climb was 53 seconds while Tripathi and Narendra obtained a trajectory requiring only 44.5 seconds to climb to 20,000 feet and attain a zero flight path angle. Neither Ref. 13 or 25 gives any indication what total amount of computational effort was expended to obtain these solutions, but the problem is still a very good candidate for future testing of the algorithm proposed in this report. The highly nonlinear characteristics of the aerodynamic forces affecting the dynamics of the system have long been a source of discomfort to people involved in flight path optimization studies. Proof of the capability of the proposed algorithm to solve this type of problem would certainly enhance the credibility of the algorithm.

The obvious discrepancy between the results obtained by Mason³¹ for the Command Service Module abort problem and the results obtained in this study makes the task of evaluating the results of Chapter 4 doubly difficult. Even if both investigations had obtained almost identical results and Mason had given figures concerning the computation effort required for his solution (which he didn't), the task of comparing a direct method to an indirect method is still difficult. Mason's indirect solution technique requires comparatively good initial guesses of the unknown state and multiplier values at the start of each branch. The projection operator technique has demonstrated that even a crude guess suffices to obtain the optimal control in a fairly modest time (46 seconds).

Mason's³¹ program implementing his indirect approach requires that the operator stop the program and change various parameters during the execution of the program should the perturbed trajectories used in updating the missing initial state and multiplier values vary too significantly or insignificantly from the unperturbed solution. He admits that although the technique is useful as a research tool it would be highly inefficient for production computations. In contrast, the projection operator algorithm has been demonstrated here to be highly automated and efficient.

Assuming the results presented here for the Command Service Module abort problem are correct, there is one observation which can be made from Table 5 which should be emphasized. Correcting the nominal control back to the constraint surface decreases the branch 3 burn time from 250 seconds to 208.32 seconds. At the end of the third iteration the branch 3 burn time is 207.93 seconds. Thus, the majority of the optimization process occurs during the initial correction to the constraint surface. The parameter α used in the linear time transformation for branch 3 appears to have very rapid rotation when it is not near its optimal value and then slows down when the optimal value has been almost reached.

Most indirect optimization techniques are well known for their instability. That is, given some set of guessed missing initial conditions, the errors at the final time for the specified values of the state and multiplier variables increase rather than decrease as the process continues. However, it is also commonly accepted that a solution arrived at using an indirect technique is normally more exact than a solution obtained by a

direct technique. It is doubtful, however, that the rather large discrepancy encountered in Chapter 4 can be attributed to this fact. Since the documentation of both the problem statement and results in Refs. 31 and 32 is incomplete, there is a good possibility that both investigators have obtained correct solutions to different problems. Obviously, no recommendation for further study would be complete without some suggestion to either validate or disprove the results of this report. If the results presented here are correct they should be able to be used to provide extremely good estimates of the unknown initial values of the missing state and multiplier values at the initial time of the three branches for use with an indirect optimization technique.

In addition to the two branched trajectory problems for which Mason³¹ gives results, he develops the background for two additional branched trajectory problems. One of these branched trajectories involves lunar lander/orbiter maneuvers for which one of the branches has a coasting arc. This problem would certainly be an interesting challenge for the projection operator technique. Although Mason mentions that an attempt was made to solve the lunar lander/orbit problem, no results are presented. The second problem involves air traffic control near airports and is interesting solely due to its importance in today's society.

The rectangular wing problem presented in Chapter 5 is an excellent example problem to demonstrate the feasibility of the projection operator algorithm applied to inequality-constrained problems. It is simple enough to allow an almost exact numerical solution, and yet difficult enough to challenge the algorithm and show the potential of conjugate direction methods

for picking the direction of search. Since the exact solution for this problem is known, the obvious criterion on which to judge a particular numerical solution is the value of the cost functional obtained. Since even for the poorest solution, the cost functional J differs less than $(.6)10^{-6}$ percent from that for the exact solution, it is important to determine other performance criteria which would be used for any problem involving inequality constraints.

The second obvious performance criterion is the computational time required to obtain a solution. The computer execution times for solutions one through five are, respectively 23.45 sec, 21.62 sec, 22.75, 30 sec, and 30 sec. Since solutions one and two have the highest values of the cost functional at the last step, it is difficult to interpret these figures. However, the conjugate gradient solution, solution three, exhibits slightly superior performance capabilities.

Since the necessary condition for optimality is that the projected gradient vanish, the parameter $\langle \tilde{g}, \tilde{g} \rangle$ listed in Tables 6 through 10 provides a good measure of convergence of the solution. Comparison of solutions one, two, four and five shows approximately the same behavior with respect to this parameter. The final value of $\langle \tilde{g}, \tilde{g} \rangle$ for solution three, the conjugate gradient solution, is more than an order of magnitude less than the other solutions.

Two additional performance criteria are: 1) the value of the time point t_1 in comparison with its value given by the analytical solution, and 2) the value of $|u(t_1^+) - u(t_1^-)|$. Solutions four and five have a slight advantage over solutions one and two with respect to these criteria, but

this advantage is not overly significant. Once again the conjugate gradient solution has a substantial advantage over the other four solutions in this regard.

Although it has been fairly well established that solution three is the best of the five solutions, there are a number of additional observations involving the gradient projection algorithm. It should be noted that the optimal solution is obtained from a fairly poor initial guess of the optimal control. The control converges whether the initial contact time t_1 is to the right or left of the optimal contact time. One very interesting observation is that the contact time always converges to the optimal contact time from the unconstrained side, that is, from the left. If the guessed contact time of the nominal control is to the right of the true optimal contact time it jumps on the first step to the left of the true optimal contact time and again converges from the left.

The fact that Tripathi and Narendra's method²⁵ for updating the contact time t_1 works (see solutions four and five) is not surprising in itself, but the fact that it does as well as or perhaps even better than the 'pure' method used to obtain solution one and two is an item of interest. It appears to have a definite advantage in its ability to move the time point t_1 to the right, which allows the value of the discontinuity at t_1 , i.e. the value of $|u(t_1^+) - u(t_1^-)|$, to decrease. The behavior of the time point t_1 using the linear time transformation to handle the unspecified time point is certainly different than that described earlier for the branched trajectory problem. When the initial nominal control for the branched trajectory problem was corrected back to the constraint, the value of the unspecified

final time moved very quickly toward its optimal value. For the rectangular wing problem, the value of t_1 moved very little when the initial nominal control was corrected back to the constraint. The inability of solutions one and two to move the time point t_1 to the right is perhaps due to inherent properties of the linear time transformation. Whatever the cause, the use of the conjugate gradient algorithm alleviates much of the difficulty.

One last observation concerns the choice of nominal controls used for structural optimization problems. Numerical experience gained through this investigation indicates that it is much easier to reduce the skin thickness to satisfy the terminal state constraints than it is to increase the skin thickness to satisfy the terminal state constraints. Therefore, the nominal guess which is picked should have approximately at least as much mass contained in the skin as the uniform reference structure. This was also the reason for increasing the complexity of the one-dimensional minimization with regard to the panel flutter problem.

No analytical solution exists for the panel flutter problem with a panel of solid construction or even for the simpler problem involving a panel of sandwich construction. This greatly complicates the task of evaluating the results of this investigation. Some of the earliest numerical attempts in this field involved the use of finite element models. Turner³⁹ attempted to solve the problem using a model employing only three uniform panel segments. The thickness distribution obtained from this work does not compare well with thickness distributions which are now considered to be optimal. Weisshaar³⁷ using four tapered elements has been able to at least approximate a skin thickness distribution which is considered to be optimal.

The computational effort involved in obtaining this approximate answer is not known. Even if this information were known, the problem of comparing the computational effort of an approximate solution to that of an 'exact' solution using two completely different techniques would be very difficult.

Pierson³⁸ has had some success at solving both the sandwich construction panel flutter problem and the solid construction panel flutter problem. His results compare very favorably with other results in this field. The foremost disadvantage of the discrete-variable technique employed by Pierson is the large amount of computational effort required to obtain a solution. In addition, the discrete-variable technique has not yet been used to solve the problem with a minimum skin thickness constraint. This complicates a comparison of the results obtained by Pierson and the results obtained by this investigation. The optimal unconstrained panel thickness is zero at both ends and possibly in the middle so that the inequality constraint is certainly desirable from a physically realistic standpoint.

Weisshaar³⁷ has also been able to solve the sandwich panel construction panel flutter problem with an inequality constraint on the skin thickness using an indirect optimization technique. The value for the cost functional he obtained using the optimal skin thickness was 0.9020. This figure compares favorably with the results of this investigation. In addition, the general shape of the skin thickness distribution obtained here compares very favorably with the shape of the skin thickness distribution obtained by Weisshaar. Weisshaar was able to start the iteration procedure for his indirect technique by setting λ_0 equal to zero, solving that subproblem and

then using that result as an initial guess to solve the next subproblem with a larger value of λ_0 until the desired value of λ_0 was obtained. Unfortunately, he gives no estimate of the total computational effort involved in this procedure.

Although the six trial solutions obtained by this investigation cannot be compared to an analytical solution, a significant amount of information concerning the performance of the proposed algorithm is given by a comparison of the six trial solutions with each other. Even though the exact value of the optimal cost functional is not known, much insight can be obtained by the relative values of the cost functional obtained by the various solutions. The other performance criterion discussed previously, such as computational effort required for a solution, satisfaction of the optimality condition, and the magnitudes of the discontinuities in the control variable at the time points t_1 through t_4 are also relevant.

Each of the six solutions appears to be converging to the same optimal skin thickness distribution. Considering the motion of the time points t_1 through t_4 and the discontinuities of the control at these time points, solutions one and six are definitely superior to solutions two through five. It is interesting to observe how close the values of the cost functional for solutions two through five approximate those values of solutions one and six while seeming to be still very far from the optimal solution. Solutions two through four certainly have difficulty in varying the time points t_1 through t_4 in an optimum manner. It definitely appears that time transformations (4.4) and (4.5) are not especially well suited for solving structural optimization problems. This does not necessarily discredit the linear time

transformation technique's usefulness in conjunction with flight path optimization problems.

The results obtained by solutions one and six are very similar. Even though solution six provides a somewhat lower value of the cost functional J , it also requires almost three times the computer time. The big advantage of these two solutions over the other trial solutions is their ability to uniformly reduce the value of the discontinuities in the control function by motion of time points t_1 through t_4 . Solution five, the conjugate gradient solution, drives the discontinuity at t_4 to zero at a rapid rate but appears to ignore the discontinuities at t_1 , t_2 , and t_3 .

The reason for solution six performing so much better than solution three is not completely understood. Certainly some of the additional performance can be attributed to greater numerical accuracy. The motion of the parameters $x_2(0)$ and $x_4(0)$ appears to be greater for this solution than for the other solutions using a linear time transformation to handle unspecified time points. Comparison of solutions two and three indicates the advisability of allowing the additional degree of freedom afforded by letting both $x_2(0)$ and $x_4(0)$ vary rather than just $x_2(0)$. Perhaps the nondimensionalization process used in obtaining solution six affords even greater flexibility to the problem parameters.

The rather successful attempt to determine the optimal skin thickness distribution by solution one indicates that the solid construction panel flutter problem is possibly not as difficult a problem to solve as originally thought. The twelve steps of solution one required a modest 57 sec. The optimal skin thickness obtained by solution one and plotted in Fig. 13

is symmetrical for all practical purposes. The Tripathi and Narendra technique²⁵ definitely has an advantage over the solutions obtained by using the linear time transformations. This is perhaps due to the gradient elements associated with the motion of the unspecified time points being a direct function of the magnitude of the discontinuity of the control at that time point rather than an integral over a time period.

As stated in Chapter 5, the proposed algorithm could not decrease the value of the cost functional below that value obtained by step twelve even though none of the values of the discontinuities had been driven to zero. Since this phenomenon did not happen to any of the other solutions, it must be attributed to the Tripathi and Narendra technique²⁵ of combining the two methods of handling unspecified time points. There are two probable causes for this phenomenon happening. The direction of search used for solution one was the steepest descent or negative projected gradient direction. If the time intervals were actually shortened or lengthened as the second technique for handling unspecified time points indicates, there is no doubt that to proceed in the negative projected gradient direction would decrease the value of the cost functional. However, by changing the values of the time points t_1 through t_4 by an expansion or compression of the time scale there is no longer the assurance that the direction of search is in a direction which causes a decrease in the value of the cost functional. It was noted earlier that the number of function evaluations needed to correct a control back to the constraints was greater using Tripathi and Narendra's technique than by using the linear time transformation technique. It is quite possible that the former technique does not generate a smooth enough curve over which the one-dimensional minimization algorithm must operate.

The results of Chapter 5 are really quite encouraging. The panel flutter problem solved in Chapter 5 is not of the straightforward caliber as the problems solved in Chapter 3 or 4. There remains, however, many avenues for investigation. It was stated early in Chapter 5 that the control variable inequality constraint is the trivial example of a state variable inequality constraint. The obvious suggestion for further study is then the application of the projection operator technique developed in Chapter 5 to a problem involving a state variable inequality constraint.

One conjugate direction algorithm was used in conjunction with the projection operator developed in Chapter 5. It would be interesting to determine the effect of some other conjugate direction algorithms on the convergence history. The second technique for changing the values of the unspecified time points t_1 through t_4 which should be used in conjunction with the second projection operator developed in Chapter 5 was not programmed due to difficulty of the programming logic. It is possible that the difficulty encountered in programming the technique would be offset by better performance characteristics of the resulting algorithm.

It would be possible to do some further study of the algorithm proposed in Chapter 5 with only a slight change of the panel flutter problem presented here. The panel flutter problem presented here assumed the panel to be simply supported. The panel flutter problem could be solved using the boundary conditions consistent with the panel being clamped at both ends. The panel flutter problem presented here also assumed zero in-plane stress. Pierson³⁸ has noted an decrease in computational effort required to obtain

a solution for increased compression in the panel. It would be interesting to determine the affect of increased compression in the panel on the performance of the proposed projection operator technique.

REFERENCES

1. Athans, M., "The Status of Optimal Control Theory and Applications for Deterministic Systems," IEEE Transactions on Automatic Control, Vol. AC-11, No. 3, July 1966, pp. 580-596.
2. Paiewonsky, B., "Optimal Control: A Review of Theory and Practice," AIAA Journal, Vol. 3, No. 11, Nov. 1965, pp. 1985-2006.
3. Athans, M. and Falb, P. L., Optimal Control: An Introduction to the Theory and Its Applications, McGraw-Hill Book Company, New York, 1966.
4. Payne, J. A., "Computational Methods in Optimal Control Problems," in Advances in Control Systems, edited by C. T. Leondes, Academic Press, New York, 1969, pp. 74-164.
5. Kelley, H. J., "Methods of Gradients," Optimization Techniques, edited by G. Leitmann, Academic Press, New York, 1962, Chap. 6, pp. 206-254.
6. Bryson, A. E. and Denham, W. F., "A Steepest-Ascent Method for Solving Optimum Programming Problems," Journal of Applied Mechanics, Vol. 29, June 1962, pp. 247-257.
7. Fiacco, A. V. and McCormick, G. P., Nonlinear Programming, John Wiley and Sons, Inc., New York, 1968.
8. Lasdon, L. S., Mitter, S., and Waren, A. D., "The Conjugate Gradient Method for Optimal Control Problems," IEEE Transactions on Automatic Control, Vol. AC-12, No. 2, April 1967, pp. 132-138.
9. Pierson, B. L. and Rajtora, S. G., "Computational Experience with the Davidson Method Applied to Optimal Control Problems," IEEE Transactions on Systems Science and Cybernetics, Vol. SSC-6, No. 3, July 1970, pp. 240-242.
10. Denham, W. F. and Bryson, A. E., Jr., "Optimal Programming Problems with Inequality Constraints II: Solution by Steepest-Ascent," AIAA Journal, Vol. 2, No. 1, Jan. 1964, pp. 25-34.
11. Sinnott, J. F., Jr. and Luenberger, D. G., "Solution of Optimal Control Problems by the Method of Conjugate Gradients," Proceedings of the Joint Automatic Control Conference, American Automatic Control Council, 1967, pp. 566-574.
12. Willoughby, J. K. and Pierson, B. L., The Projection Operator Applied to Gradient Methods for Solving Optimal Control Problems with Terminal State Constraints, Tech. Rep. ERI-71140, Engineering Research Institute, Ames, Iowa, Dec. 1971.

13. Mehra, R. K. and Bryson, A. E., Jr., "Conjugate Gradient Methods with and Application to V/STOL Flight-Path Optimization," Journal of Aircraft, Vol. 6, No. 2, March-April 1969, pp. 123-128.
14. Horwitz, L. B. and Sarachik, P. E., "Davidon's Method in Hilbert Space," SIAM Journal of Applied Mathematics, Vol. 16, No. 4, July 1968, pp. 676-695.
15. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming: Part I. Linear Constraints," J. SIAM, Vol. 8, No. 1, March 1960, pp. 181-217.
16. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming: Part II. Nonlinear Constraints," J. SIAM, Vol. 9, No. 4, Dec. 1961, pp. 514-532.
17. Junkins, J. L., "Equivalence of the Minimum Norm and Gradient Projection Constrained Optimization Techniques", Proceedings of the Astrodynamics Specialists Conference, Ft. Lauderdale, Florida, 1971.
18. Rozendaal, L., "A General Branched Trajectory Optimization Algorithm with Applications to Space Shuttle Vehicle Mission Design," Proceedings of the Astrodynamics Specialists Conference, Ft. Lauderdale, Florida, 1971.
19. Gera, J., "Branched Trajectory Optimization by the Projected Gradient Technique," AIAA Journal, Vol. 8, No. 6, June 1970, pp. 1121-1126.
20. McCart, B. R., Haug, E. J., and Streeter, T. D., "Optimal Design of Structures with Constraints on Natural Frequency," AIAA Journal, Vol. 8, No. 6, June 1970, pp. 1012-1019.
21. Rajtora, S. G. and Pierson, B. L., "An Automated Gradient Projection Algorithm for Optimal Control Problems", AIAA Journal, Vol. 10, No. 7, July 1972, pp. 949-951.
22. Luenberger, D. G., Optimization by Vector Space Methods, John Wiley and Sons, Inc., New York, 1969, Chap. 3.
23. Bryson, A. E. and Ho, Y. C., Applied Optimal Control, Blaisdell, Waltham, Mass., 1969, pp. 39-41, pp. 47-50.
24. Willcoughby, J. K., Adaptations of the Conjugate Gradient Method to Optimal Control Problems with Terminal State Constraints, Tech. Rep. ERI-62500, Engineering Research Institute, Ames, Iowa, Feb. 1970.
25. Tripathi, S. S. and Narendra, K. S., "Constrained Optimization Problems Using Multiplier Methods," Journal of Optimization Theory and Applications, Vol. 9, No. 1, Jan. 1972, pp. 59-70.

26. Icerman, L. J., "Optimal Structural Design for Given Dynamic Deflection," International Journal of Solids and Structures, Vol. 5, No. 5, May 1969, pp. 473-490.
27. Turner, M. J., "Design of Minimum Mass Structures with Specified Natural Frequencies," AIAA Journal, Vol. 3, No. 5, March 1967, pp. 406-412.
28. Taylor, J. E., "Minimum Mass Bar for Axial Vibration at Specified Natural Frequency," AIAA Journal, Vol. 5, No. 10, Oct. 1967, pp. 1911-1913.
29. Ashley, H. and McIntosh, S. C., Jr., "Application of Aeroelastic Constraints in Structural Optimization," Applied Mechanics, edited by M. Hetenyi and W. G. Vincenti, Proceedings 12th International Congress of Applied Mechanics, Stanford University, Springer, Berlin, Aug. 1969, pp. 100-113.
30. Pierson, B. L., "A Discrete-Variable Approximation to Optimal Flight Paths," Astronautica Acta, Vol. 14, No. 2, Feb. 1969, pp. 157-169.
31. Mason, J. D., Some Optimal Branched Trajectories, Tech. Rep. NASA CR-1331, National Aeronautics and Space Administration, Washington, D.C., May 1969.
32. Mason, J. D., Smith, D. B., and Dickerson, W. D., "Secondary Mission Optimization", Journal of Spacecraft and Rockets, Vol. 6, No. 4, April 1969, pp. 409-413.
33. Yodzis, C. W., "Liquid Rockets", Manned Spacecraft: Engineering Design and Operation, edited by P. E. Purser, M. A. Faget, and N. F. Smith, Fairchild Publications, Inc., New York, 1964.
34. Bryson, A. E., Jr., Denham, W. F., and Dreyfus, S. E., "Optimal Programming Problems with Inequality Constraints I: Necessary Conditions for Extremal Solutions, AIAA Journal, Vol. 1, No. 11, Nov. 1963, pp. 2544-2550.
35. Armand, J. and Vitte, W. J., Foundations of Aeroelastic Optimization and Some Applications to Continuous Systems, Tech. Rep. SUDAAR N. 390, Department of Aeronautics and Astronautics, Stanford University, Stanford, California, Jan. 1970.
36. McIntosh, S. C. Jr., Weisshaar, T. A., and Ashley, H., Progress in Aeroelastic Optimization - Analytical Versus Numerical Approaches, Tech. Rep. SUDAAR No. 383, Department of Aeronautics and Astronautics, Stanford University, Stanford, California, July 1969.

37. Weisshaar, T. A., An Application of Control Theory Methods to the Optimization of Structures Having Dynamic or Aeroelastic Constraints, Tech. Rep. SUDAAR No. 412, Department of Aeronautics and Astronautics, Stanford University, Stanford, California, Oct. 1970.
38. Pierson, B. L., A Discrete Variable Approximation to Minimum Weight Panel Designs Subject to a Supersonic Flutter-Speed Constraint, Tech. Rep. ERI-71128, Engineering Research Institute, Ames, Iowa, Dec. 1971.
39. Turner, M. J., "Optimization of Structures to Satisfy Flutter Requirements," AIAA Journal, Vol. 7, No. 5, May 1969, pp. 945-951.

ACKNOWLEDGMENTS

The author wishes to express his sincere gratitude to Dr. Bion L. Pierson for his guidance and encouragement during this study. The author is also grateful to Dr. J. D. Iversen, Dr. E. W. Anderson, and Dr. A. N. Michel for their guidance throughout the author's graduate studies.

Finally, the author would like to thank the National Aeronautics and Space Administration for financial support during this study and Mrs. Sheila Hilts for her patience in the typing of the manuscript.

APPENDIX :

FORTRAN IV LISTING OF QUADRATIC INTERPOLATING
ONE-DIMENSIONAL SEARCH

```

      SUBROUTINE LINMIF(STPEST)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON F,OPTSTP,PC,PSI,GSQ,DFA,T,TI,X(10),D(10),STOX1(101),STOX2(1
101),STOU(101),G(101),S(101),GG( 1),Y(101),STOX3(101),ITAB,NFEVAL,
2NREINT,NV,KOUN,ISTEP,IN,IEND,IX
      COMMON/AUX/SSQ,PSTEP,PSQ
      COMMON/PROK/PX(101),PY(101),PZ(101),PTOU(101),Z(101)
      IWORK=0
      ALPHA=0.00
      FA=F
      BETA=ALPHA+STPEST
      CALL FUNCT(BETA)
      CALL PROBAC
C  SUBROUTINE PROBAC CORRECTS THE CONTROL BACK TO THE CONSTRAINT
      IF(F.GT.FA) GO TO 600
601 DELTA=BETA
      FD=F
      STPEST=4.00*STPEST
      BETA=ALPHA+STPEST
      CALL FUNCT(BETA)
      CALL PROBAC
      IF(F.GT.FD) GO TO 620
      ALPHA=DELTA
      FA=FD
      GO TO 601
600 FB=F
      DELTA=0.700*ALPHA+0.300*BETA
      CALL FUNCT(DELTA)
      CALL PROBAC
      IF(FA.GT.F) GO TO 621
      BETA=DELTA
      GO TO 600
620 FB=F
      GO TO 602
621 FD=F
602 D23=DELTA-BETA
      WRITE(6,100)ALPHA,DELTA,BETA,FA,FD,FB

```

```

100 FORMAT(/////6D20.7)
    C31=BETA-ALPHA
    D12=ALPHA-DELTA
    DS23=DELTA**2-BETA**2
    DS31=BETA**2-ALPHA**2
    DS12=ALPHA**2-DELTA**2
    OPTSTP=.5D0*(DS23*FA+DS31*FD+DS12*FB)/(D23*FA+D31*FD+D12*FB)
    CALL FUNCT(OPTSTP)
    CALL PROBAC
    IF(FD.LT.F)GO TO 603
    COEFA=(FA-FD)/(DS12-2.D0*(OPTSTP*D12)
    COEFB=-2.D0*COEFA*OPTSTP
    COEFC=FA-COEFA*ALPHA**2-COEFB*ALPHA
    FEST=COEFC-0.25D0*COEFB**2/COEFA
    IF(FA.GT.FB)GO TO 604
    PERF=.05D0*(FB-FEST)
    GO TO 605
604 PERF=.05D0*(FA-FEST)
605 IF(DABS(F-FEST).LT.PERF)RETURN
    IF(IWORK.GT.4.AND.DABS(F-FD).LT.1.D-4) GO TO 606
    IF(OPTSTP.LT.DELTA)GO TO 607
    ALPHA=DELTA
    FA=FD
    DELTA=OPTSTP
    FD=F
    GO TO 608
607 BETA=DELTA
    FB=FD
    DELTA=OPTSTP
    FD=F
608 IWORK=IWORK+1
    NREINT=NREINT+1
    GO TO 602
606 WRITE(6,650)
650 FORMAT(///'LAST RESORT EXIT TAKEN IN LINMIF'///)
    RETURN
603 IF(IWORK.GT.4.AND.DABS(F-FD).LT.1.D-4)GO TO 609

```

```

        IF(F.GT.FA.AND.OPTSTP.LT.DELTA)GO TO 610
        IF(OPTSTP.LT.DELTA)GO TO 611
        BETA=OPTSTP
        FB=F
        GO TO 608
611  ALPHA=OPTSTP
        FA=F
        GO TO 608
610  BETA=OPTSTP
        FB=F
        DELTA=0.100*BETA+0.900*ALPHA
        CALL FUNCT(DELTA)
        CALL PRGBAC
        FD=F
        IF(FD.LT.FA)GO TO 608
        BETA=DELTA
        FB= FD
        DELTA=0.100*BETA+0.900*ALPHA
        CALL FUNCT(DELTA)
        CALL PRGBAC
        FD=F
        IF(FD.LT.FA)GO TO 608
        WRITE (6,650)
        OPTSTP= ALPHA
        RETURN
609  F=FD
        OPTSTP=DELTA
        WRITE(6,650)
        RETURN
        END

```