

A GRADIENT COMPUTATIONAL TECHNIQUE
FOR A CLASS OF OPTIMAL CONTROL
PROBLEMS SUBJECT TO
INEQUALITY CONSTRAINTS

Thesis for the Degree of Ph. D.
MICHIGAN STATE UNIVERSITY
SHYH JONG WANG
1969



This is to certify that the

thesis entitled

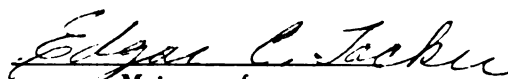
A GRADIENT COMPUTATIONAL TECHNIQUE
FOR A CLASS OF OPTIMAL CONTROL PROBLEMS
SUBJECT TO INEQUALITY CONSTRAINTS

presented by

Shyh Jong Wang

has been accepted towards fulfillment
of the requirements for

Ph. D. degree in E. E.


Major professor

Date 9/18/69

ABSTRACT

A GRADIENT COMPUTATIONAL TECHNIQUE FOR A CLASS OF OPTIMAL CONTROL PROBLEMS SUBJECT TO INEQUALITY CONSTRAINTS

By

Shyh Jong Wang

A broad class of optimal control problems with isoperimetric constraints and instantaneous algebraic constraints, the control problem of Bolza, are considered. An important subclass of the general control problem of Bolza which contains the bang-bang control problems and problems with continuous control variables as well as discrete control variables are also considered.

Local linearization and perturbation techniques are used to obtain the computational algorithms for the solutions of these problems. Iterative procedures for these algorithms are given in detail. A sample program for the general algorithm written in FORTRAN is presented in APPENDIX A.

Isoperimetric inequality constraints are transformed into equality constraints by introducing additional control parameters, and a penalty function technique is used for treating the instantaneous algebraic constraints.

Three numerical examples are given to illustrate the application of the optimization process, and to demonstrate the influences of the alternative choices of the iterative parameters and their respective updating schemes. The solutions of these examples are presented in curves and in tabular forms, and the results are discussed.

A GRADIENT COMPUTATIONAL TECHNIQUE
FOR A CLASS OF OPTIMAL CONTROL PROBLEMS SUBJECT TO
INEQUALITY CONSTRAINTS

Shyh Jong Wang

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering
and System Science

1969

2011
10-10

TO MY PARENTS

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to his advisor, Dr. Edgar C. Tacker, for his constant support, motivation, and guidance throughout this work. The author also wishes to thank the other members of his Guidance Committee, Dr. Harry G. Hedges, Dr. John B. Kreer, Dr. Richard C. Dubes, and Dr. J. Sutherland Frame, for their guidance of his doctoral program and for their helpful suggestions during the final preparation of this thesis. Thanks are also due to Mr. Robert J. Greiner for his help in programming and for many important suggestions.

The author gratefully acknowledges the support provided to his work by the Division of Engineering Research. Use of the Michigan State University computing facilities was made possible through support, in part, from the National Science Foundation.

Finally, the author wishes to thank his wife, Nancy, for her patience, understanding, and encouragement throughout his graduate work.

TABLE OF CONTENTS

	Page
ABSTRACT	
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	vii
LIST OF TABLES	ix
LIST OF APPENDICES	x
I INTRODUCTION	1
II THE CLASSICAL VARIATIONAL THEORY AND OPTIMAL CONTROL PROBLEMS	4
2.1 The First Variation of a Functional	4
2.2 Extremum of a Functional; A Necessary Condition	6
2.3 The Second Variation of a Functional; A Necessary Condition Involving the Second Variation	8
2.4 The Euler-Lagrange Equation	9
2.5 A Generalization of Euler-Lagrange Equation to the n-Dimensional Fixed End-Point Problem	10
2.6 Weierstrass Necessary Condition (For a Strong Extremum)	13
2.7 Legendre's Necessary Condition	15
2.8 Corner Conditions	15
2.9 Jacobi's Necessary Condition; Conjugate Points	16

	Page
2.10 Necessary Conditions; Sufficient Conditions	17
2.11 Variable End-Point Problem	18
2.12 A General Problem of the Calculus of Variations—the Problem of Bolza	21
2.13 The Multiplier Rule for the Problem of Bolza	23
2.14 The Control Problem of Bolza	26
2.15 Summary	33
 III A GRADIENT COMPUTATIONAL TECHNIQUE FOR THE CONTROL PROBLEM OF BOLZA, ALGORITHM I	 34
3.1 A Formulation of the Control Problem of Bolza	34
3.2 A Simple Transformation	36
3.3 Derivation of Variational Formulas for some Quantities of Interest	37
3.4 Constrained Effort and Successive Optimization, Algorithm I	52
3.5 An Iterative Procedure for Algorithm I	59
3.6 Some Measures for the Validity of Linear Approximation	60
3.7 The Instantaneous Algebraic Constraints and the Penalty Function Technique	61
3.8 Conclusion	65
 IV A SUBCLASS OF THE CONTROL PROBLEM OF BOLZA, ALGORITHM II	 67
4.1 Formulation of the Problem	67
4.2 A Multiple-stage Formulation of the Problem	70
4.3 The Variations of the Multiple-stage Problem with Fixed Number of Stages	73

	Page
4.4 Successive Optimization Process for the Problem with Fixed Number of Stages	76
4.5 A Computational Procedure for the Problem with Fixed Number of Stages	78
4.6 Optimization Process for Adding a New Stage and the Corresponding Computational Procedure	80
4.7 A Computational Procedure for Algorithm II	85
V NUMERICAL EXAMPLES	87
5.1 Brachistochrone Problem with Inequality Stage Constraint	87
5.2 Orbit Transfer of a Solar Sail Ship	100
5.3 Low Thrust Trajectory Optimization Problem	110
VI SUMMARY AND CONCLUSION	121
REFERENCES	125

LIST OF FIGURES

	Page
Figure	
1 The arcs x^* and X	14
2 Neighboring curves for variable end-point problem	19
3 A nominal solution arc and its neighboring arc with variable end-points	44
4 Brachistochrone problem	89
5 Paths of brachistochrone problem - Case I . . .	95
6 t_f vs. N , brachistochrone - Case I	96
7 t_f vs. N , brachistochrone - Case II	96
8 t_f vs. N , brachistochrone - Case IV	98
9 Control functions, brachistochrone - Case IV .	98
10 Paths of brachistochrone problem - Case IV . .	99
11 Orbit transfer of a solar sail ship	102
12 Trajectories of solar sail ship - Case II . . .	108
13 Sail angle of solar sail problem - Case II . .	109
14 Orbit transfer of a low thrust rocket	111

Figure		Page
15	Trajectories of low thrust rocket - Case I . .	116
16	Thrust angle of low thrust rocket - Case I . .	117
17	Trajectories of low thrust rocket - Case II . .	119
18	Thrust angle of low thrust rocket - Case II . .	120

LIST OF TABLES

Table		Page
1	Numerical results, solar sail - Case I	106
2	Numerical results, solar sail - Case II	107
3	Numerical results, low thrust - Case I	115
4	Numerical results, low thrust - Case II	118

LIST OF APPENDICES

		Page
Appendix		
A	A Sample Program	129

I INTRODUCTION

The behavior of a physical system can usually be described mathematically in terms of the parameters of the system. One may call this representation the mathematical model of the system. The behavior of a large class of such physical systems can be represented or approximated by a set of simultaneous ordinary differential equations of the first order, and the scope of this thesis is limited to this class.

Once a mathematical model is given or identified, one can, at least in principle, choose the parameters, or the control variables of the system such that the performance of the system is optimized in a sense which is specified beforehand. The object or the functional which is to be extremized subject to the mathematical constraints of the system is called the performance index, or the cost functional of the system.

A complete understanding and mastering of the problem is not a simple matter; it has been the major work of a branch of applied mathematics, the calculus of variations, for almost three hundred years.

The calculus of variations has been studied and developed by many mathematicians, necessary and sufficient conditions for many problems have been developed. However,

only relatively few problems can be solved using these conditions directly.

The primary interest of engineers and scientists is perhaps the realization of a given problem, that is to obtain a specific solution of the problem. In most cases, practical optimization problems are sufficiently complicated that solutions in compact form can not be obtained. Therefore numerical techniques have been studied extensively since high speed computers became available.

One of the most general problems in the calculus of variations is the so-called problem of Bolza. This problem was first formulated by Bolza in 1913 [BL-1], has stimulated a great deal of research and study since 1930, and has become a very important problem in optimal control theory. Hestenes [HE-1, -2] formulated an equivalent problem to the problem of Bolza which is in a more desirable form for the study of optimal control theory, we shall call this problem the control problem of Bolza. In Chapter II of this thesis, we shall review briefly the work of the classical variational theory and modern optimal control theory developed in the past three hundred years. This material will serve as background for the later development of this thesis.

The contributions of this work are in Chapter III and Chapter IV. The contribution in Chapter III is the extension of the method of gradients [KE-1, BR-1] to the

control problem of Bolza with various constraints, and that in Chapter IV is the formulation of an important subclass of the control problem of Bolza, and the derivation of a computational algorithm for the solution of the problem.

Three numerical examples are given in Chapter V to illustrate the application of the computational algorithm for the general problem, and to show the effects of the iteration parameters on the speed of convergence.

In Chapter VI, we summarize this work and discuss the possible extensions for further research.

A sample program is given in Appendix A.

II THE CLASSICAL VARIATIONAL THEORY AND OPTIMAL CONTROL PROBLEMS

The variational calculus has been developing since the late seventeenth century, having its beginning with Johann Bernoulli who posed the brachistochrone problem in 1696. This problem was solved by the Bernoulli brothers, Newton, de l'Hospital, and others. In 1697, Johann Bernoulli solved another well known problem—the problem of geodesics, later L. Euler and J. Lagrange solved the general problem of this type.

Besides these two problems there is another problem—the classical isoperimetric problem. The general method of solving this problem was given by L. Euler. These three problems have had a great influence on the development of the variational calculus.

For convenience as well as to introduce notation, we shall now give a brief discussion of the classical variational theory, indicating the various mathematical arguments employed and the results obtained. This will lead us to the consideration of optimal control problems.

2.1 The First Variation of a Functional

Let X be a normed linear space, whose elements are real-valued functions defined on a closed interval I of R , where R is the real line. A functional $J(\cdot)$ is defined as

a mapping which maps X into R . We shall particularly be interested in the functional

$$J(x) = \int_{t_0}^{t_1} F(t, x, \dot{x}) dt \quad (2.1.1)$$

where $\dot{x} \triangleq \frac{dx}{dt}$, and F is a real-valued function of its arguments. The space X will be considered to be endowed with the norm $(\|\cdot\|)$ defined by

$$\|x\| = \sup_{t \in I} |x(t)| \quad (2.1.2)$$

where $I = [t_0, t_1]$, and $x \in X$. The notion of distance between two elements x, y of the space X can be defined as the norm of their difference (i.e., the distance induced by $\|\cdot\|$),

$$d(x, y) = \|x - y\| \quad (2.1.3)$$

Give a fixed function $x(t)$ and its increment $\xi(t)$ of the space X , the corresponding increment of the functional $J(x)$ is a functional of ξ ,

$$\Delta J(\xi) = J(x + \xi) - J(x) \quad (2.1.4)$$

Suppose we can write

$$\Delta J(\xi) = l(\xi) + \varepsilon \|\xi\| \quad (2.1.5)$$

where $l(\xi)$ is a linear functional of ξ , and $\varepsilon \rightarrow 0$ as $\|\xi\| \rightarrow 0$. The functional J is differentiable if it has the above

property. $J'(\xi)$ is called the first variation of J along x , and we shall denote it by $J'(x, \xi)$. One can show that $J'(x, \xi)$ is unique (GE-1).

2.2 Extremum of a Functional; A Necessary Condition

Let X_1 be a normed linear space of all real-valued continuous functions on I having piecewise continuous derivatives. It is clear that an element of X_1 is also an element of X . The elements of X_1 will be called admissible functions. It is convenient for the subsequent discussion to define another norm, $\|\cdot\|_1$,

$$\|x\|_1 = \|x\| + \|\dot{x}\| = \sup_{t \in I} |x(t)| + \sup_{t \in I} |\dot{x}(t)| \quad (2.2.1)$$

Let $\bar{x}$ be an arbitrary but fixed element of X_1 . The strong neighborhood of $\bar{x}$ corresponding to $\varepsilon > 0$, $N_s(\bar{x}, \varepsilon)$, is the set of all functions x in X_1 such that $\|x - \bar{x}\| < \varepsilon$, while the weak neighborhood of $\bar{x}$ with $\varepsilon > 0$, $N_w(\bar{x}, \varepsilon)$, is the set of all functions x in X_1 such that $\|x - \bar{x}\|_1 < \varepsilon$. It is clear that for a given ε and $\bar{x}$, the strong neighborhood contains the weak neighborhood. We shall say that $x = \bar{x}$ yields a strong relative extremum of J if there is an $\varepsilon > 0$ such that for all $x \neq \bar{x}$ and $x \in N_s(\bar{x}, \varepsilon)$, $J(x) - J(\bar{x})$ has the same sign. On the other hand, we say that $x = \bar{x}$ yields a weak relative extremum of J if there exists a weak neighborhood $N_w(\bar{x}, \varepsilon)$, such that $J(x) - J(\bar{x})$ has the same sign for all $x \neq \bar{x}$ and $x \in N_w(\bar{x}, \varepsilon)$. Since $\|x - \bar{x}\|_1 < \varepsilon$

implies $\|x - \bar{x}\| \leq \varepsilon$, if $\bar{x}$ yields an extremum of J with respect to all x with $\|x - \bar{x}\| \leq \varepsilon$, then $\bar{x}$ yields an extremum of J with respect to all x with $\|x - \bar{x}\|_1 \leq \varepsilon$. Hence it is clear that if $\bar{x}$ furnishes a strong extremum of J it also furnishes a weak extremum. A necessary condition for a weak extremum is also a necessary condition for a strong extremum.

It is important to note that a relative extremum is determined by local properties of the functional in question and it is always associated with a neighborhood of that which yields the extremum. On the other hand, an absolute extremum is determined by global properties of the functional. In the subsequent discussions, only relative extrema are considered; therefore, necessary and sufficient conditions for an extremum are local although, for simplicity, we may not mention it explicitly.

The following lemma is basic in the calculus of variations and is known as the fundamental lemma [HE-4].

Lemma 2.2.1 Let $M(t)$, $N(t)$ be piecewise continuous functions on I . Then

$$\int_{t_0}^{t_1} (M(t)\xi(t) + N(t)\dot{\xi}(t))dt = 0 \quad (2.2.2)$$

holds for all ξ in X_1 and $\xi(t_0) = \xi(t_1) = 0$ if and only if there exists a constant c such that

$$N(t) = \int_{t_0}^t M(T)dT + c \quad t_0 \leq t \leq t_1 \quad (2.2.3)$$

Theorem 2.2.1 A Necessary Condition. If J is differentiable, and if $\bar{x}$ furnishes an extremum for J , then

$$J'(\bar{x}, \xi) = 0 \quad (2.2.4)$$

for all ξ in X_1 .

The proofs of this theorem and those which will be stated in the subsequent sections can be found in most of the standard texts (see, for example, [BL-2, GE-1, HE-4]).

2.3 The Second Variation of a Functional; A Necessary Condition Involving the Second Variation

We say that the functional J is twice differentiable if its increment can be expressed as

$$\Delta J(\xi) = l(\xi) + q(\xi) + \varepsilon \|\xi\|^2 \quad (2.3.1)$$

where $l(\xi)$ is the first variation of J , $q(\xi)$ is a quadratic functional, and $\varepsilon \rightarrow 0$ as $\|\xi\| \rightarrow 0$. $q(\xi)$ is called the second variation of J along x . The second variation of J is also unique and is denoted by $J''(x, \xi)$.

Theorem 2.3.1 A Necessary Condition. Let $J(x)$ be twice differentiable, then $x = \bar{x}$ yielding a minimum (maximum) of J implies that

$$J''(\bar{x}, \xi) \geq 0 \ (\leq 0) \quad (2.3.2)$$

for all ξ in X_1 .

2.4 The Euler-Lagrange Equation

Consider all the arcs x in X_1 having fixed end points $x(t_0) = x_0$, $x(t_1) = x_1$. We shall refer to these arcs as the admissible arcs of fixed end-point. In this section we are going to state the first necessary condition of the following problem:

Among all the admissible arcs of fixed end-point, find the one yielding a weak extremum for the functional

$$J(x) = \int_{t_0}^{t_1} F(t, x, \dot{x}) dt \quad (2.4.1)$$

We shall assume that F is a known function of $(t, x, \dot{x})$ with continuous first and second partial derivatives with respect to all its arguments. This problem is known as the simplest variational problem.

Theorem 2.4.1 If $x = x^*$ furnishes the functional $J(x)$ an extremum in the class of all admissible arcs connecting its end-points x_0 and x_1 , then there exists a constant c such that

$$F_{\dot{x}}^* = \int_{t_0}^t F_x dt + c \quad (2.4.2)$$

holds at every point of the arc $x = x^*$, where $F_x \triangleq \frac{\partial F}{\partial x}$, and $F_{\dot{x}} \triangleq \frac{\partial F}{\partial \dot{x}}$.

Equation (2.4.2) is known as the integral form of

the Euler-Lagrange equation

$$\frac{d}{dt} F_{\dot{x}} = F_x \quad (2.4.3)$$

An admissible arc satisfying the Euler-Lagrange equation is call an extremaloid.

Note that the extremum of J in Theorem 2.4.1 is a weak extremum. However, as mentioned earlier, a strong extremum is also a weak extremum and any necessary condition for a weak extremum is also a necessary condition for a strong extremum. This implies that Theorem 2.4.1 gives a necessary condition for both weak and strong extrema.

2.5 A Generalization of Euler-Lagrange Equation to the n-Dimensional Fixed End-Point Problem

Let $x = (x^1, \dots, x^n)^T$ be an n -vector and each component of x be a real-valued function on I . Let X_1 be a linear space of all n -dimensional functions x and each component of x is continuous on I and has a piecewise continuous derivative. We shall call members of X_1 admissible arcs. Let $F(t, x, \dot{x})$ be a continuous function having continuous first and second partial derivatives with respect to all its arguments. The problem is stated as: find the necessary conditions (Euler-Lagrange) for the functional J to have a weak extremum furnished by an admissible arc

$x = x^*$ with fixed end-points $x(t_0) = x_0$ and $x(t_1) = x_1$.

J is defined as

$$J(x) = \int_{t_0}^{t_1} F(t, x, \dot{x}) dt \quad (2.5.1)$$

Let $\xi = (\xi^1, \dots, \xi^n)^T$ be an increment of an admissible arc x . We shall call ξ an admissible increment of an arc x if ξ is a member of X_1 and $\xi(t_0) = \xi(t_1) = 0$. The corresponding increment ΔJ is

$$\Delta J(\xi) = J(x + \xi) - J(x)$$

$$= \int_{t_0}^{t_1} (F(t, x + \xi, \dot{x} + \dot{\xi}) - F(t, x, \dot{x})) dt \quad (2.5.2)$$

The first variation of ΔJ can be obtained by expanding (2.5.2) using Taylor's Theorem,

$$J'(x, \xi) = \int_{t_0}^{t_1} (F_x \xi + F_{\dot{x}} \dot{\xi}) dt \quad (2.5.3)$$

where F_x and $F_{\dot{x}}$ are the row vectors of partial derivatives

with respect to x and $\dot{x}$ respectively. We can write

$$F_x = (F_{x1} \dots F_{xn}) \quad (2.5.4)$$

$$F_{\dot{x}} = (F_{\dot{x}1} \dots F_{\dot{x}n}) \quad (2.5.5)$$

In (2.5.3) choosing all the components of ξ except the i th one to be zero, we have

$$J'(x, \xi) = \int_{t_0}^t (F_{x^i} \xi^i + F_{\dot{x}^i} \dot{\xi}^i) dt \quad i = 1, \dots, n \quad (2.5.6)$$

Suppose x^* extremizes J among the admissible arcs connecting its end-points, by Theorem 2.2.1 the expression (2.5.6) must be zero for $x = x^*$, then by Lemma 2.2.1 there exists a constant c_i such that the equations

$$F_{\dot{x}^i} = \int_{t_0}^t F_{x^i} dt + c_i \quad i = 1, \dots, n \quad (2.5.7)$$

hold along x^* . We thus have the following theorem:

Theorem 2.5.1 If the arc x^* extremizes J among all the admissible arcs connecting its end-points, then there exist n constants c_i such that

$$F_{\dot{x}^i} = \int_{t_0}^t F_{x^i} dt + c_i \quad i = 1, \dots, n$$

hold along x^* .

Equation (2.5.7) is the integral form of the Euler-Lagrange equations

$$\frac{d}{dt} F_{\dot{x}}^i = F_x^i \quad i = 1, \dots, n \quad (2.5.8)$$

2.6 Weierstrass Necessary Condition (For a Strong Extremum)

Let $x = x^*$ be a (strong) minimizing arc for J among all admissible arcs with fixed end-points. The arcs considered are n -dimensional. The Weierstrass E -function is defined as

$$E(t, x, \dot{x}, \dot{X}) = F(t, x, \dot{X}) - F(t, x, \dot{x}) - F_{\dot{x}}(t, x, \dot{x})(\dot{X} - \dot{x}) \quad (2.6.1)$$

In (2.6.1) $(t, x, \dot{x})$ is a point on the arc x^* at time t and having derivative $\dot{x}$. Similarly, $(t, x, \dot{X})$ is a point on an (neighboring) admissible arc x (see Figure 1).

The following theorem is due to Weierstrass,

Theorem 2.6.1 Weierstrass Necessary Condition. If the arc x^* furnishes a strong minimum for J among the class of admissible arcs connecting its end-points, then

$$E(t, x, \dot{x}, \dot{X}) \geq 0 \quad (2.6.2)$$

holds for every $(t, x, \dot{x}, \dot{X})$ with $(t, x, \dot{x})$ on x^* and $(t, x, \dot{X})$ on an admissible arc and $\dot{X} \neq \dot{x}$.

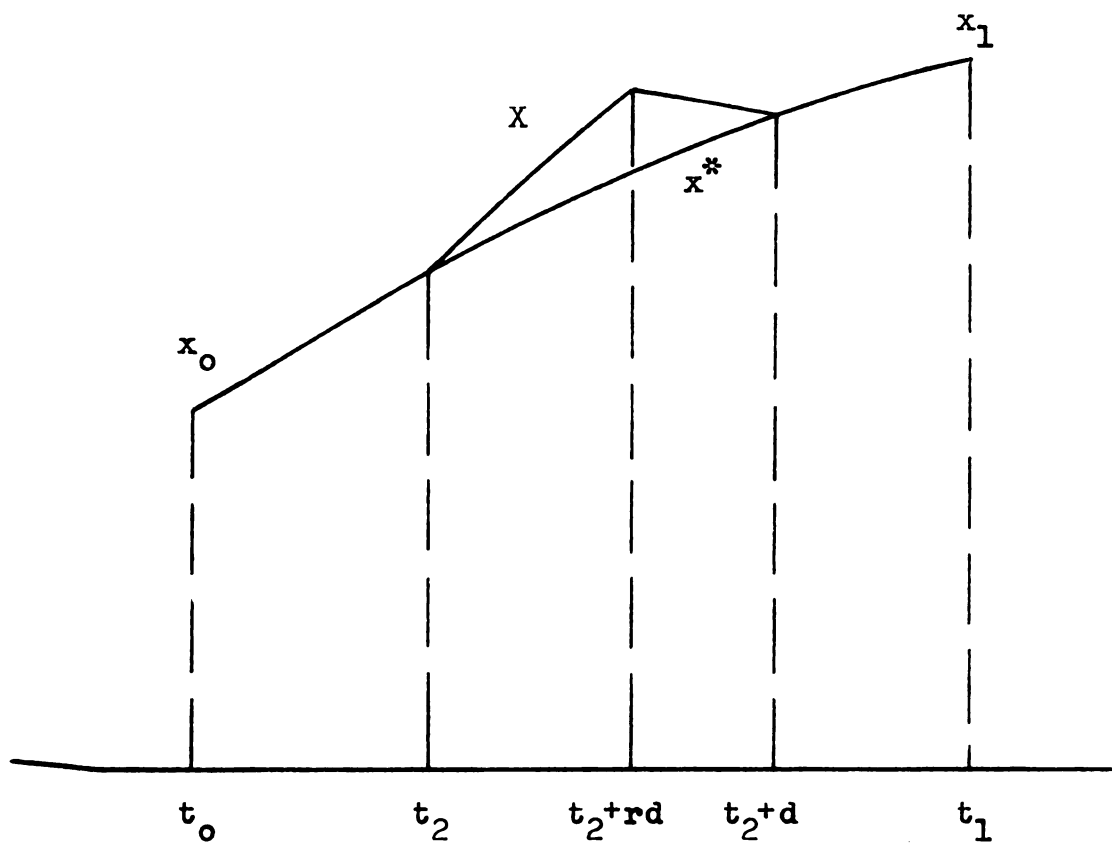


Figure 1 The arcs x^* and X

Remark 1: The condition (2.6.2) is trivially satisfied if $\dot{X} = \dot{x}$.

Remark 2: If x^* (strongly) maximizes J the inequality sign in (2.6.2) should be reversed.

2.7 Legendre's Necessary Condition

Theorem 2.7.1 If the arc x^* (weakly) minimizes the functional J in the class of admissible arcs connecting its end-points, then

$$a^T F_{\ddot{x}\ddot{x}}(t, x, \dot{x}) a \geq 0 \quad (2.7.1)$$

along x^* for each non-zero vector $a = (a^1, \dots, a^n)^T$. Where

$$F_{\ddot{x}\ddot{x}} = \begin{bmatrix} F_{\ddot{x}\ddot{x}}^{1.1} & \cdot & \cdot & \cdot & \cdot & F_{\ddot{x}\ddot{x}}^{1.n} \\ \vdots & & & & & \\ \vdots & & & & & \\ \vdots & & & & & \\ F_{\ddot{x}\ddot{x}}^{n.1} & \cdot & \cdot & \cdot & \cdot & F_{\ddot{x}\ddot{x}}^{n.n} \end{bmatrix}$$

This condition is a consequence of the Weierstrass necessary condition, although Legendre proved it very differently and at a much earlier time.

2.8 Corner Conditions

The following is known as the Weierstrass-Erdmann corner conditions,

Theorem 2.8.1 Let x^* be an admissible arc satisfying the necessary conditions (2.5.7), the Euler-Lagrange equations, then (I) $F_x(t, x, \dot{x})$ is continuous along x^* , and (II) If x^* also satisfies the Weierstrass necessary condition then $F - F_x \dot{x}$ is continuous along x^* .

2.9 Jacobi's Necessary Condition; Conjugate Points

An admissible arc x^* without corners which is a solution of (2.5.7) is called an extremal. Any solution of (2.5.7) consists of a finite number of extremals. An admissible arc x along which the determinant of F_{xx} is non-zero is said to be nonsingular.

An extremal which is also nonsingular is called a nonsingular extremal. From the hypothesis on F and the above theorem, a nonsingular extremal has a continuous second derivative.

We say that a point $(t_2, x(t_2))$, $t_0 < t_2 \leq t_1$, on the extremal x^* is conjugate to $(t_0, x(t_0))$, if there is an accessory extremal ξ such that $\xi(t_0) = \xi(t_2) = 0$ but $\xi(t) \neq 0$ on $t_0 < t < t_2$.

Theorem 2.9.1 Jacobi's Necessary Condition. If x^* is a nonsingular minimizing arc without corners for the functional $J(x)$ defined by equation (2.5.1) in the class of admissible arcs connecting its end-points, then there does not exist any point $(t_2, x(t_2))$ with $t_0 < t_2 < t_1$ on x^* conjugate to the point $(t_0, x(t_0))$.

2.10 Necessary Conditions; Sufficient Conditions

I. Necessary Conditions for a Weak Extremum: If the admissible arc x^* on I provides the functional

$$J(x) = \int_{t_0}^{t_1} F(t, x, \dot{x}) dt \quad (2.10.1)$$

a weak (relative) extremum among the class of admissible arcs connecting its end-points x_0 and x_1 , then

- (Ia) The Euler-Lagrange equations are satisfied by x^* (Theorem 2.5.1).
- (Ib) The Legendre's necessary condition holds along x^* (Theorem 2.7.1).
- (Ic) The Jacobi's necessary condition holds (Theorem 2.9.1).

II. Necessary Conditions for Strong Extremum: If x^* on I provides a strong (relative) extremum for the functional (2.10.1) among the class of admissible arcs connecting its end-points, then the conditions (Ia), (Ib), (Ic) and (IIa) are satisfied, where (IIa) is

- (IIa) The Weierstrass necessary condition (Theorem 2.6.1).

III. Sufficient Conditions for a Weak Extremum: If for an admissible arc x^* , the following conditions hold simultaneously, then x^* provides a weak (relative) minimum for the functional (2.10.1):

- (IIIa) The arc x^* is an extremal, that is, x^* satisfies (Ia) and x^* has no corners.

(IIIb) The strengthened Legendre's necessary condition, that is, the inequality of (2.7.1) is replaced by strict inequality.

(IIIc) The strengthened Jacobi's necessary condition, that is, there is no point in $(t_0, t_1]$ on x^* conjugate to $(t_0, x(t_0))$.

IV. Sufficient Conditions for a Strong Minimum: If the admissible arc satisfies (IIIa), (IIIb), (IIIc) and (IVa) then x^* provides a strong (relative) minimum for the functional (2.10.1) with fixed end-points, where (IVa) is

(IVa) There exists a neighborhood of the elements $(t, x^*, \dot{x}^*)$ on x^* such that the strict inequality $E(t, x, \dot{x}, \dot{X}) > 0$ holds for all $(t, x, \dot{x}, \dot{X})$ with $(t, x, \dot{x})$ in the neighborhood, $(t, x, \dot{X})$ admissible, and $\dot{X} \neq \dot{x}$.

For further discussion of sufficient conditions, see [HE-4, GE-1, BL-2].

2.11 Variable End-point Problem

Let x and y be admissible arcs defined on $I = [t_0, t_1]$ and $I' = [t'_0, t'_1]$ respectively, with end-points $x_0 = x(t_0)$, $x_1 = x(t_1)$, $y_0 = y(t'_0)$, $y_1 = y(t'_1)$. Extending x and y to some interval containing $I \cup I'$. The distance between the arcs x and y is defined as

$$d(x, y) = \sum_{i=1}^n d(x^i, y^i) \quad (2.11.1)$$

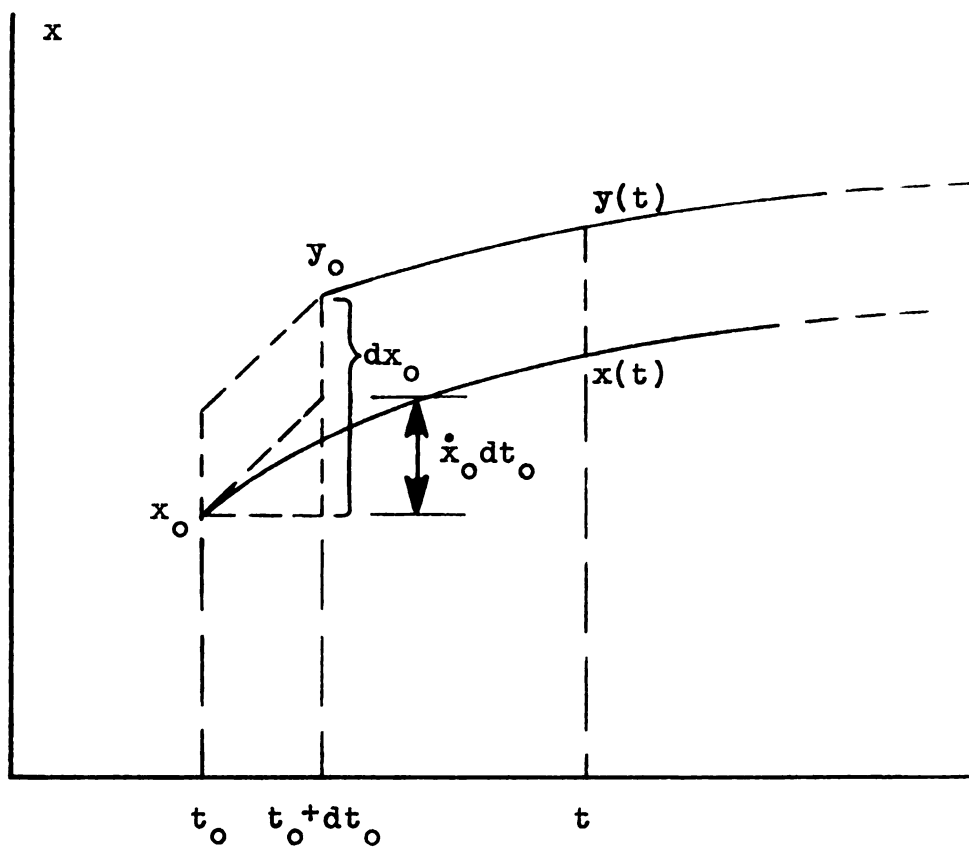


Figure 2 Neighboring curves for variable end-point problem

where

$$d(x^i, y^i) = \sup_t |x^i(t) - y^i(t)| + \sup_t |\dot{x}^i(t) - \dot{y}^i(t)| \\ + |x^i(t_0) - y^i(t_0)| + |\dot{x}^i(t_1) - \dot{y}^i(t_1)| \quad (2.11.2)$$

Let x and y be neighboring arcs, let ξ be their difference

$$\xi(t) = y(t) - x(t) \quad (2.11.3)$$

and write the end-points of y as

$$y_0 = (t_0 + dt_0, x_0 + dx_0)$$

$$y_1 = (t_1 + dt_1, x_1 + dx_1)$$

The functional $J(x)$ is defined as

$$J(x) = \int_{t_0}^{t_1} F(t, x, \dot{x}) dt \quad (2.11.4)$$

where F is a known function having continuous partial derivatives up to the second order with respect to all its arguments.

The first variation of J is defined as the linear part of the increment $\Delta J(\xi)$ with respect to $\xi, \dot{\xi}, dt_0, dx_0$ and dx_1 , and is denoted by $J'(x, \xi)$. We state the following theorem,

Theorem 2.11.1 If the arc x^* minimizes the functional J on the class of admissible arcs with variable end-points, then x^* satisfies the necessary conditions stated in Section 2.10, and the boundary condition

$$\left[(F - F_{\dot{x}} \dot{x}) dt + F_{\dot{x}} dx \right]_{t_0}^{t_1} = 0 \quad (2.11.5)$$

The boundary condition (2.11.5) is called the transversality condition.

2.12 A General Problem of the Calculus of Variations— the Problem of Bolza

The problem of Bolza can be stated as follows:
Find in the class of arcs x satisfying the equality constraints

$$\phi^s(t, x, \dot{x}) = 0 \quad s = 1, \dots, q < n \quad (2.12.1)$$

and the boundary constraints

$$\theta^\gamma(t_0, x(t_0), t_f, x(t_f)) = 0 \quad \gamma = 1, \dots, p \leq (2n+1) \quad (2.12.2)$$

the one which minimizes the functional

$$J(x) = g(t_0, x(t_0), t_f, x(t_f)) + \int_{t_0}^{t_f} L(t, x, \dot{x}) dt \quad (2.12.3)$$

where ϕ^s , θ^γ , g , and L are real-valued functions of their arguments. The continuity properties of these functions will be given in Section 2.13.

This form of the problem of Bolza was formulated by Bliss [BL-3]. There are two other equally general problems, the problem of Mayer, and the problem of Lagrange. These two latter problems can be obtained from the problem of Bolza by setting, respectively, $L \equiv 0$ and $g \equiv 0$ in (2.12.3). On the other hand, one can obtain the problem of Bolza from the problem of Mayer and that of Lagrange through suitable transformations as introduced below: Consider the $n+1$ dimensional arc x, x^{n+1} defined on $I = [t_0, t_f]$ satisfying the constraints

$$\begin{aligned} \phi^s(t, x, \dot{x}) &= 0 & s &= 1, \dots, q \\ \dot{x}^{n+1} - L(t, x, \dot{x}) &= 0 \end{aligned} \quad (2.12.4)$$

and

$$\begin{aligned} \theta^\gamma(t_0, x(t_0), t_f, x(t_f)) &= 0 & \gamma &= 1, \dots, p \\ x^{n+1}(t_0) &= 0 \end{aligned} \quad (2.12.5)$$

With this transformation, the functional J becomes

$$J(x) = g(t_0, x(t_0), t_f, x(t_f)) + x^{n+1}(t_f) , \quad (2.12.6)$$

which defines the problem of Mayer.

If we consider the class of arcs x, x^{n+1} on I satisfying the constraints

$$\begin{aligned}\phi^s(t, x, \dot{x}) &= 0 & s &= 1, \dots, q \\ \dot{x}^{n+1}(t) &= 0\end{aligned}\tag{2.12.7}$$

and

$$\begin{aligned}\theta^\gamma(t_0, x(t_0), t_f, x(t_f)) &= 0 & \gamma &= 1, \dots, p \\ x^{n+1}(t_f) - \frac{1}{t_f - t_0} g(t_0, x(t_0), t_f, x(t_f)) &= 0\end{aligned}\tag{2.12.8}$$

The functional J becomes

$$J(x) = \int_{t_0}^{t_f} [L(t, x, \dot{x}) + x^{n+1}(t)] dt\tag{2.12.9}$$

This is the problem of Lagrange. Hence we have proved the equivalence of the problems of Bolza, Mayer, and Lagrange.

2.13 The Multiplier Rule for the Problem of Bolza

Let x be an n -dimensional point which represents the "state" of an arc x at time $t \in I$, thus $(t, x, \dot{x})$ is a $(2n+1)$ -tuple. Let R_{2n+1} be an open region (arcwise connected open set) of elements $(t, x, \dot{x})$, and let ϕ^s and L be continuous on R_{2n+1} and have continuous first partial derivatives. Assume that the matrix $\phi_{\dot{x}}^s$ has its maximum rank q on the

subset R_{2n+1}^0 of R_{2n+1} defined by the constraints (2.12.1). Note that $\emptyset$ is the vector notation for $\emptyset^s$, $s = 1, \dots, q$. Let $x(t_0) = x_0$, $x(t_f) = x_f$, and let S_{2n+2} be an open region of $(2n+2)$ -tuples (t_0, x_0, t_f, x_f) on which θ^γ , and g are continuous and have continuous first partial derivatives, and the matrix $(\theta_{t_0} \quad \theta_{x_0} \quad \theta_{t_f} \quad \theta_{x_f})$ has its maximum rank p at each point of S_{2n+2} . Points in R_{2n+1} , S_{2n+2} are admissible if they are also in R_{2n+1}^0 . An arc x which is continuous and has piecewise continuous derivatives is admissible if its elements $(t, x, \dot{x})$ and (t_0, x_0, t_f, x_f) are admissible. With these assumptions we can state the multiplier rule for the problem of Bolza formulated in the last section [BL-2, HE-4],

Theorem 2.13.1 The Multiplier Rule. Let x^* be an admissible arc defined on $[t_0, t_f]$, which minimizes the functional J , then there exist multipliers λ^0 , $\lambda^s(t)$, $s = 1, \dots, q < n$, and h^γ , $\gamma = 1, \dots, p$ such that (1) λ^0 and h^γ are constants and are not all zero; (2) $\lambda^s(t)$ are continuous on $[t_0, t_f]$ except possibly at those t corresponding to corners of x^* where the left and right limits exist, and λ^0 , and $\lambda^s(t)$ do not vanish at the same time on $[t_0, t_f]$; (3) If we define

$$F(t, x, \dot{x}, \lambda) = \lambda^0 L(t, x, \dot{x}) + \lambda^T \emptyset(t, x, \dot{x}) \quad (2.13.1)$$

then

$$F_{\dot{x}^i} = \int_{t_0}^t F_{\dot{x}^i} dT + c^i \quad i = 1, \dots, n \quad (2.13.2)$$

$$\emptyset^s(t, x, \dot{x}) = 0 \quad s = 1, \dots, q \quad (2.13.3)$$

hold along x^* , where c^i are constants.

(4) The conditions

$$[(F - F_{\dot{x}} \dot{x})dt + F_{\dot{x}} dx]_{t_0}^{t_f} + \lambda^0 dg + h^T d\theta = 0 \quad (2.13.4)$$

and

$$\theta^\gamma(t_0, x_0, t_f, x_f) = 0 \quad \gamma = 1, \dots, p \quad (2.13.5)$$

hold at the end-points of x^* for every choice of dt_0 , dt_f , dx_0 and dx_f . And furthermore (5) The Weierstrass condition

$$E(t, x, \dot{x}, \dot{X}, \lambda) \geq 0 \quad (2.13.6)$$

hold for every $(t, x, \dot{x})$ on x^* and for each $(t, x, \dot{X})$ in R_{2n+1}^0 such that $\dot{X} \neq \dot{x}$, where

$$E(t, x, \dot{x}, \dot{X}, \lambda) = F(t, x, \dot{X}, \lambda) - F(t, x, \dot{x}, \lambda) - F_{\dot{x}}(t, x, \dot{x}, \lambda)(\dot{X} - \dot{x}) \quad (2.13.7)$$

It is easy to see, from (2.13.2), that along x^* $F_{\dot{x}}$ is continuous and has piecewise continuous derivatives, and

$$\frac{d}{dt} F_{\dot{x}} = F_x \quad (2.13.8)$$

This is the Euler-Lagrange equation (in row-vector form).

2.14 The Control Problem of Bolza

We shall formulate a very important class of optimal control problems which have their origin in the classical problem of Bolza. We shall call the new problem the control problem of Bolza. Later in this section, we shall show that these two problems of Bolza are equivalent.

Let the cost functional J be defined as

$$J = g(t_f, x(t_f), a) + \int_{t_0}^{t_f} L(t, x, u, a) dt \quad (2.14.1)$$

where t_0 is the initial time and t_f is the final or terminal time, which can either be fixed or variable; x is an n -vector, $(x^1, \dots, x^n)^T$, called the state variable (vector); u is an m -vector, $(u^1, \dots, u^m)^T$, called the control variable (vector), and a is an r -vector, $(a^1, \dots, a^r)^T$, called the control parameter (vector). We shall also use

$$x : t, x(t), u(t), a, t \in [t_0, t_f]$$

to represent the solution arc of the differential constraints

$$\dot{x} = f(t, x, u, a) \quad (2.14.2)$$

Equation (2.14.2) may be considered as the mathematical model of a system which is nonlinear in general. Let E be a region in an $(n+m+r+1)$ -dimensional Euclidean space whose elements

are the $(n+m+r+1)$ -tuples (t, x, u, a) . We assume that the real-valued functions f^i , $i = 1, \dots, n$, are continuous and have continuous first partial derivatives on E . Let E_0 be a subset of E defined by the side constraints

$$\phi^s(t, x, u, a) \leq 0 \quad s = 1, \dots, q' \quad (2.14.3a)$$

$$\phi^s(t, x, u, a) = 0 \quad s = q'+1, \dots, q \quad (2.14.3b)$$

therefore the elements of E_0 are those in E that also satisfy (2.14.3). Assume that each ϕ^s is continuous and has continuous first partial derivatives on E , and the $q \times (m+q)$ -matrix $[\phi_u \quad I\phi]$ has its maximum rank q at each point of E_0 , where $\phi = (\phi^1, \dots, \phi^q)^T$, $\phi_u = \frac{\partial \phi}{\partial u}$, and I is the $q \times q$ identity matrix. Elements of E_0 are called admissible elements. We shall assume that the control functions $u(t)$, $t \in [t_0, t_f]$ are piecewise continuous. A solution arc x of (2.14.2) with $x(t_0)$ in a given initial manifold is called an admissible arc if all its elements (t, x, u, a) are admissible.

Consider the isoperimetric constraints

$$I^\gamma \leq 0 \quad \gamma = 1, \dots, p' \quad (2.14.4a)$$

$$I^\gamma = 0 \quad \gamma = p'+1, \dots, p \quad (2.14.4b)$$

where

$$I^\gamma = \theta^\gamma(t_f, x(t_f), a) + \int_{t_0}^{t_f} M^\gamma(t, x, u, a) dt \quad (2.14.5)$$

We shall use I , θ , and M to represent the vectors $(I^1, \dots, I^p)^T$, $(\theta^1, \dots, \theta^p)^T$, and $(M^1, \dots, M^p)^T$ respectively. Assume that θ^Y , and M^Y are continuous and have continuous first partial derivatives on E . Note that terminal constraints are special cases of isoperimetric constraints and hence are included in (2.14.4).

Now we state the control problem of Bolza as follows:
Find in the class of admissible arcs

$$x : t, x(t), u(t), \alpha \qquad t \in [t_0, t_f]$$

satisfying the constraints (2.14.2), (2.14.3) and (2.14.4),
the one which minimizes the cost function J .

A problem, perhaps somewhat less general than the one given here, was first formulated by Hestenes in 1949 [HE-1,2,3]. Since the new formulation and the classical formulation of the problem of Bolza are equivalent, necessary conditions for the new problem can be obtained from the classical one through translation [HE-1, BE-1]. Necessary conditions for this problem have been obtained directly by Pontryagin, and the result is called the maximum principle, see [PO-1]. The methods of McShane [MCS-1] and Pontryagin were generalized by Hestenes, who obtained more general results [HE-3], and the results were also obtained by Guinn with weaker hypotheses [GU-1]. We shall state the first necessary condition for the control problem of Bolza obtained by Hestenes.

Setting

$$F(t, x, u, \alpha, \lambda, h) = \lambda^0 L + \lambda^T \phi + h^T M \quad (2.14.6)$$

and

$$H(t, x, u, \alpha, p, \lambda, h) = p^T f - F \quad (2.14.7)$$

$$G = \lambda^0 g + h^T \theta \quad (2.14.8)$$

where

$$\lambda^T = (\lambda^1, \dots, \lambda^q), \quad h^T = (h^1, \dots, h^p), \quad p^T = (p^1, \dots, p^n) \quad (2.14.9)$$

we have the following:

Theorem 2.14.1 Let x^* be a solution to the control problem of Bolza, then there exist multipliers λ^0 , $\lambda^s(t)$, $p^i(t)$, and h^γ such that (1) $\lambda^0 \geq 0$, $h^1, \dots, h^p$ are constants and for $\gamma = 1, \dots, p'$, $h^\gamma \geq 0$ if $I_\gamma = 0$, $h^\gamma = 0$ if $I_\gamma < 0$ along x^* . Furthermore λ^0 , h^γ , $p^i(t)$ do not vanish simultaneously at any $t \in [t_0, t_f]$. (2) $\lambda^1(t), \dots, \lambda^q(t)$ are continuous except possibly at the discontinuities of $u^*(t)$ at which the left and right limits exist. For $s = 1, \dots, q'$, $\lambda^s(t) \geq 0$ if $\phi_s = 0$ and $= 0$ if $\phi_s < 0$ along x^* . (3) $p^1(t), \dots, p^n(t)$ are continuous and along the arc x^* H is continuous and

$$\dot{x}^T = H_p \quad \dot{p}^T = -H_x \quad (\text{Euler-Lagrange Equations}) \quad (2.14.10)$$

$$H_u = 0 \quad \frac{d}{dt} H = H_t \quad (2.14.11)$$

hold along x^* .

(4) The Weierstrass condition

$$H(t, x^*(t), u, a^*, p(t), 0, h) \leq H(t, x^*(t), u^*(t), a^*, p(t), 0, h) \quad (2.14.12)$$

holds for all $t \in [t_0, t_f]$, and for all u with

$(t, x^*(t), u, a^*) \in E_0$, that is, for all u satisfying the constraints

$$\emptyset_s(t, x^*(t), u, a^*) \leq 0 \quad s = 1, \dots, q'$$

$$\emptyset_s(t, x^*(t), u, a^*) = 0 \quad s = q'+1, \dots, q$$

(5) The transversality condition

$$dG + [-Hdt + p^T dx]_{t_0}^{t_f} - \int_{t_0}^{t_f} (H_u da) dt \equiv 0$$

holds on x^* for every choice of dt_0 , dt_f , dx_0 , dx_f and da .

To show the equivalence of the control problem of Bolza and the classical problem of Bolza, consider the case with equality constraints only, i.e.,

$$\emptyset_s(t, x, u, a) = 0 \quad s = 1, \dots, q \quad (2.14.13)$$

$$I^\gamma = \theta^\gamma(t_f, x(t_f), a) + \int_{t_0}^{t_f} M^\gamma(t, x, u, a) dt \quad \gamma = 1, \dots, p \quad (2.14.14)$$

Let $y^i = x^i$, $i = 1, \dots, n$; $\dot{y}^{n+\gamma} = M^\gamma$, $y^{n+\gamma}(t_0) = 0$,
 $\gamma = 1, \dots, p$. And replace u^j by $\dot{y}^{n+p+j}$, $j = 1, \dots, m$, and for
definiteness, set $y^{n+p+j} = 0$, for $j = 1, \dots, m$. Substituting
these new variables into equations (2.14.1), (2.14.2),
 (2.14.13) and (2.14.14), we have

$$J = g(t_f; y^1, \dots, y^n; a) + \int_{t_0}^{t_f} L(t; y^1, \dots, y^n; \dot{y}^{n+p+1}, \dots, \dot{y}^{n+p+m}; a) dt \quad (2.14.15)$$

$$\ddot{y}^i - f^i(t; y^1, \dots, y^n; \dot{y}^{n+p+1}, \dots, \dot{y}^{n+p+m}; a) = 0 \quad i = 1, \dots, n \quad (2.14.16)$$

$$\emptyset^s(t; y^1, \dots, y^n; \dot{y}^{n+p+1}, \dots, \dot{y}^{n+p+m}; a) = 0 \quad s = 1, \dots, q \quad (2.14.17)$$

$$I^\gamma = \theta^\gamma(t_0; y^1(t_0), \dots, y^n(t_0); t_f; y^1(t_f), \dots, y^n(t_f)) + y^{n+\gamma}(t_f) \quad \gamma = 1, \dots, p \quad (2.14.18)$$

If we set $y = (y^1, \dots, y^{n+p+m})^T$, and

$$\phi^i = \ddot{y}^i - f^i(t, y, \dot{y}, a) \quad i = 1, \dots, n \quad (2.14.19a)$$

$$\phi^{n+\gamma} = \dot{y}^{n+\gamma} - M^\gamma(t, y, \dot{y}, a) \quad \gamma = 1, \dots, p \quad (2.14.19b)$$

$$\phi^{n+p+s} = \emptyset^s(t, y, \dot{y}, a) \quad s = 1, \dots, q \quad (2.14.19c)$$

and

$$\Theta^\gamma = \theta^\gamma(t_0, y(t_0), t_f, y(t_f)) + y^{n+\gamma}(t_f) \quad \gamma = 1, \dots, p \quad (2.14.20a)$$

$$\Theta^{p+j} = y^{n+j}(t_0) \quad j = 1, \dots, p+m \quad (2.14.20b)$$

Then, from equations (2.14.15) to (2.14.20), we have the following,

$$\Phi^s(t, y, \dot{y}, a) = 0 \quad s = 1, \dots, n+p+q \quad (2.14.21)$$

$$\Theta^\gamma(t_0, y(t_0), t_f, y(t_f)) = 0 \quad \gamma = 1, \dots, 2p+m \quad (2.14.22)$$

$$J = g(t_f, y(t_f), a) + \int_{t_0}^{t_f} L(t, y, \dot{y}, a) dt \quad (2.14.23)$$

We thus have showed that the control problem of Bolza can be expressed in the same form as that of classical problem of Bolza. Similarly, one can show the reverse. Note that one may have to use the condition that Φ_x^0 has maximum rank so that at least one set, $x^{i_1}, \dots, x^{i_q}$, can be chosen from $x^1, \dots, x^n$ such that (2.12.1) can be written as

$$\dot{x}^{i_r} = f^{i_r}(t; x^{i_1}, \dots, x^{i_q}; z^1, \dots, z^m; u^1, \dots, u^m)$$

where $m = n - q$, and u^j , $j = 1, \dots, m$ are the rest of the variables among $x^1, \dots, x^n$, and z^j satisfies the constraints

$$\dot{z}^j - u^j = 0 \quad j = 1, \dots, m$$

where x^{i_r} , $r = 1, \dots, q$ are the state variables, and u^j , $j = 1, \dots, m$, the control variables.

2.15 Summary

The variational calculus and the optimal control theory is a very extensive subject which has proven to be increasingly important to modern technology. In this chapter we have briefly reviewed the history and the development of the subject, from the classical treatment of the so called simplest problem of variational calculus to a reasonably sophisticated modern control problem. For further generalization of the subject, see [BE-2, GU-1, HE-4, PO-1, DR-1, BR-2].

In the next chapter we shall develop a computational method to obtain numerical solutions for the control problem of Bolza.

III A GRADIENT COMPUTATIONAL TECHNIQUE FOR THE CONTROL PROBLEM OF BOLZA, ALGORITHM I

In this chapter, we shall first restate the control problem of Bolza in a form which will be more convenient for computational purposes. A technique similar to that used by Valentine [VA-1] is used to convert the inequality isoperimetric constraints into equality constraints. The instantaneous side constraints are not treated directly; instead, the penalty function technique is used to treat the constraints in an indirect fashion.

Variations of various quantities of the control problem of Bolza are derived in terms of adjoint variables, or influence functions, and the variations of control function, control parameters, and initial time and state. A technique due to Bryson and Denham [BR-1] is then employed to obtain the optimal variations of various quantities. A computational procedure is given, and measures for prediction errors in dJ and dI are defined.

3.1 A Formulation of the Control Problem of Bolza

Let t , x , u , α be the independent variable—time, the n -state vector, m -control vector, and r -control parameter respectively. Let E be a region in the Euclidean space with

elements (t, x, u, α) . Let f^i ($i = 1, \dots, n$), L , M^0 , M^γ ($\gamma = 1, \dots, p$) be defined on E , and assume that f^i , L , M^0 , M^γ are of $C^1(E)$. Furthermore, let g , θ^0 , θ^γ ($\gamma = 1, \dots, p$) be functions of (t, x, α) , and of C^1 for each (t, x, α) such that $(t, x, u, \alpha) \in E$. We shall assume that the control function $u^j(t)$ ($j = 1, \dots, m$) is piecewise continuous on $[t_0, t_f]$, where t_0 and t_f are the initial and terminal times respectively. We shall call $u(t)$ an admissible control function, if it is piecewise continuous, and the corresponding (t, x, u, α) is in E for each $t \in [t_0, t_f]$.

Define the cost functional, the differential constraints, and the isoperimetric constraints as those defined in Chapter II,

$$J = g(t_f, x(t_f), \alpha) + \int_{t_0}^{t_f} L(t, x, u, \alpha) dt \quad (3.1.1)$$

$$\dot{x} = f(t, x, u, \alpha) \quad (3.1.2)$$

$$\bar{I}^\gamma \leq 0 \quad \gamma = 1, \dots, p' \quad (3.1.3a)$$

$$\bar{I}^\gamma = 0 \quad \gamma = p'+1, \dots, p \quad (3.1.3b)$$

$$\bar{I}^\gamma = \theta^\gamma(t_f, x(t_f), \alpha) + \int_{t_0}^{t_f} M^\gamma(t, x, u, \alpha) dt \quad (3.1.3c)$$

For computational convenience, we shall define a stopping functional,

$$S(t_f) = \theta^0(t_f, x(t_f), \alpha) + \int_{t_0}^{t_f} M^0(t, x, u, \alpha) dt \quad (3.1.4)$$

where $S = 0$ defines the value of t_f , and we assume that

$\dot{s} \neq 0$ for all $t \in [t_0, t_f]$.

Let x be a solution arc of the system (3.1.2) which can be written explicitly as,

$$x : t, x(t), u(t), a \qquad t \in [t_0, t_f]$$

We shall call elements of E admissible elements, and we shall call x an admissible arc if x is a solution arc of (3.1.2) corresponding to an admissible control, and each element of x is in E .

The control problem of Bolza is to determine the admissible control function $u(t)$, the control parameter a , and the corresponding admissible arc x , such that the cost functional J is minimized and the constraints (3.1.3) are satisfied.

Since the instantaneous constraints (2.14.3) are treated indirectly, the admissible set E_0 has been relaxed to E . This is satisfactory especially when iterative computation is concerned.

3.2 A Simple Transformation

It is convenient from the computational point of view to transform the constraints (3.1.3) into equality constraints at the expense of introducing additional control parameters.

Let β be the additional parameter vector with components $\beta^1, \dots, \beta^{p'}$, and let Z be a p -vector whose

components are defined as,

$$z^\gamma = (\beta^\gamma)^2 \quad \gamma = 1, \dots, p' \quad (3.2.1a)$$

$$z^\gamma = 0 \quad \gamma = p'+1, \dots, p \quad (3.2.1b)$$

If we set $I^\gamma = \bar{I}^\gamma + z^\gamma$, then constraints (3.1.3) are equivalent to

$$I^\gamma = 0 \quad \gamma = 1, \dots, p \quad (3.2.2)$$

The control problem defined in Section 3.1 can be restated in terms of the new constraints as follows:

Determine the admissible control $u(t)$, the control parameters α , β , and the corresponding admissible arc x such that the cost functional J is minimized and the constraints (3.2.2) are satisfied.

3.3 Derivation of Variational Formulas for Some Quantities of Interest

Let x^* be an admissible arc, we shall call it a nominal solution arc,

$$x^* : t^*, x^*(t^*), u^*(t^*), \alpha^*, \beta^* \quad t^* \in [t_0^*, t_f^*]$$

Let ε be an arbitrary small positive number, and let $d(x, x^*)$ be a distance measure between the arcs x , and x^* defined by

$$\begin{aligned}
d(x, x^*) = & \sup_{t \in T \cup T^*} |t - t^*| + \max_{i=1, \dots, n} |x^i(t_0) - x^{*i}(t_0^*)| \\
& + \max_{i=1, \dots, n} |x^i(t_f) - x^{*i}(t_f^*)| \\
& + \sup_{t \in T \cup T^*} \max_{j=1, \dots, m} |u^j(t) - u^{*j}(t^*)| \\
& + \max_{k=1, \dots, r} |\alpha^k - \alpha^{*k}| + \max_{\gamma=1, \dots, p} |\beta^\gamma - \beta^{*\gamma}|
\end{aligned}$$

where $T = [t_0, t_f]$, and $T^* = [t_0^*, t_f^*]$. Let x be a neighboring arc of x^*

$$x : t, x(t), u(t), \alpha, \beta \quad t \in [t_0, t_f]$$

such that $d(x, x^*) < \varepsilon$. Let Δt , and $\Delta x(t^*)$ denote the increments

$$\Delta t = t - t^* \quad (3.3.1)$$

$$\Delta x(t^*) = x(t) - x^*(t^*) \quad (3.3.2)$$

Let δt and $\delta x(t^*)$ be the variations corresponding to Δt and $\Delta x(t^*)$ respectively, then for arbitrarily small ε , we can write

$$\Delta t = \delta t + o(\varepsilon) \quad (3.3.3)$$

$$\Delta x(t^*) = \delta x(t^*) + o(\varepsilon) \quad (3.3.4)$$

where $o(\varepsilon)$ satisfies

$$\lim_{\varepsilon \rightarrow 0} \frac{o(\varepsilon)}{\varepsilon} = 0 \quad (3.3.5)$$

Let $\Delta \bar{x}(t^*)$ be the increment

$$\Delta \bar{x}(t^*) = x(t^*) - x^*(t^*) = \delta \bar{x}(t^*) + o(\varepsilon) \quad (3.3.6)$$

where $\delta \bar{x}(t^*)$ is the corresponding variation. In a later development we shall need the relation between $\delta x(t^*)$ and $\delta \bar{x}(t^*)$ which can be obtained as

$$\begin{aligned} \Delta x(t^*) &= x(t) - x^*(t^*) \\ &= [x(t^*) - x^*(t^*)] + [x(t) - x(t^*)] \\ &= \delta \bar{x}(t^*) + \left. \frac{dx}{dt^*} \right|_{t^*} \Delta t + o(\varepsilon) \\ &= \delta \bar{x}(t^*) + \left. \frac{dx^*}{dt^*} \right|_{t^*} \delta t + o(\varepsilon) \end{aligned} \quad (3.3.7)$$

In the above derivation we used the facts that $o(\varepsilon) + o(\varepsilon) = o(\varepsilon)$ and $\frac{dx}{dt^*} = \frac{dx^*}{dt^*} + o(\varepsilon)$. Comparing (3.3.4) with (3.3.7) and noting that both $\delta x(t^*)$ and $\delta \bar{x}(t^*) + \dot{x}^*(t^*)\delta t$ are principal linear parts of $\Delta x(t^*)$, we have

$$\delta x(t^*) = \delta \bar{x}(t^*) + \dot{x}^*(t^*)\delta t \quad (3.3.8)$$

where $\dot{x}^*(t^*) \triangleq \left. \frac{dx^*}{dt^*} \right|_{t^*}$. Now we shall derive the variational

equation for $\delta x(t^*)$. The state equations (3.1.2) corresponding to x^* , and x are

$$\dot{x}^*(t^*) = f(t^*, x^*(t^*), u^*(t^*), a^*) \quad t^* \in [t_0^*, t_f^*]$$

$$\dot{x}(t) = f(t, x, u, a) \quad t \in [t_0, t_f]$$

Their difference is

$$\begin{aligned} \Delta \dot{x}(t^*) &= f(t, x(t), u(t), a) - f(t^*, x^*(t^*), u^*(t^*), a^*) \\ &= \left(\frac{\partial f}{\partial t} \right)_* \Delta t + \left(\frac{\partial f}{\partial x} \right)_* \Delta x(t^*) + \left(\frac{\partial f}{\partial u} \right)_* \Delta u(t^*) \\ &\quad + \left(\frac{\partial f}{\partial a} \right)_* \Delta a + o(\varepsilon) \end{aligned} \quad (3.3.9)$$

where

$$\left(\frac{\partial f}{\partial x} \right)_* = \begin{bmatrix} \frac{\partial f^1}{\partial x^1} & \dots & \frac{\partial f^1}{\partial x^n} \\ \vdots & & \\ \frac{\partial f^n}{\partial x^1} & \dots & \frac{\partial f^n}{\partial x^n} \end{bmatrix}_*$$

and $\left(\frac{\partial f}{\partial t} \right)_*$, $\left(\frac{\partial f}{\partial u} \right)_*$ and $\left(\frac{\partial f}{\partial a} \right)_*$ are defined similarly. The symbol $(.)_*$ means that the quantity in the parentheses is evaluated along the nominal arc x^* .

If the notions of the variations δt , $\delta x(t^*)$, $\delta u(t^*)$, and δa are introduced in (3.3.9) and the result compared with

$$\Delta \dot{x}(t^*) = \delta \dot{x}(t^*) + o(\varepsilon) \quad (3.3.10)$$

we have the variational equation:

$$\delta \dot{x}(t^*) = \left(\frac{\partial f}{\partial t} \right)_* \delta t + \left(\frac{\partial f}{\partial x} \right)_* \delta x(t^*) + \left(\frac{\partial f}{\partial u} \right)_* \delta u(t^*) + \left(\frac{\partial f}{\partial a} \right)_* \delta a \quad (3.3.11)$$

Let Q be the quantity of interest, and be of the general form

$$Q = G(t_f, x(t_f), a) + Z(\beta) + \int_{t_0}^{t_f} F(t, x, u, a) dt \quad (3.3.12)$$

Adjoining the differential constraint (3.1.2) to (3.3.12), and forming the difference of Q along the arcs x and x^* , we have

$$\begin{aligned} \Delta Q = & G(t_f, x(t_f), a) - G(t_f^*, x^*(t_f^*), a^*) + Z(\beta) - Z(\beta^*) \\ & + \int_{t_0}^{t_f} [F(t, x(t), u(t), a) + \lambda^T(t)(f(t, x(t), u(t), a) - \dot{x}(t))] dt \\ & - \int_{t_0^*}^{t_f^*} [F(t^*, x^*(t^*), u^*(t^*), a^*) + \lambda^{T^*}(t^*)(f(t^*, x^*(t^*), \\ & \quad \quad \quad u^*(t^*), a^*) - \dot{x}^*(t^*))] dt^* \end{aligned} \quad (3.3.13)$$

So far we have treated the matter in a fairly general way, from now on we shall focus our attention on our specific problem. Letting

$$t_0 = t_0^* + \Delta t_0 \quad (3.3.14a)$$

$$t_f = t_f^* + \Delta t_f \quad (3.3.14b)$$

$$t = t^* \quad \text{for } t \in [t_0^* + \Delta t_0, t_f^*] \quad (3.3.14c)$$

and extending the definitions of the arcs x and x^* to $[t_0^*, t_f^* + \Delta t_f]$, (3.3.13) becomes,

$$\begin{aligned} \Delta Q = & G(t_f + \Delta t_f, x(t_f + \Delta t_f), a) - G(t_f, x^*(t_f), a^*) + Z(\beta) - Z(\beta^*) \\ & + \int_{t_0}^{t_f} [F(t, x, u, a) - F(t, x^*, u^*, a^*) + \lambda^T (f(t, x, u, a) \\ & \quad - f(t, x^*, u^*, a^*) - \dot{x} + \dot{x}^*)] dt \\ & + \int_{t_f}^{t_f + \Delta t_f} [F(t, x, u, a) + \lambda^T (f(t, x, u, a) - \dot{x})] dt \\ & - \int_{t_0}^{t_0 + \Delta t_0} [F(t, x, u, a) - \lambda^T (f(t, x, u, a) - \dot{x})] dt \end{aligned} \quad (3.3.15)$$

Note that in (3.3.15) we have, for simplicity, dropped the * associated with t . Expanding (3.3.15), we have

$$\begin{aligned} \Delta Q = & \left(\frac{\partial G}{\partial t} \right)_{*, t_f} \Delta t_f + \left(\frac{\partial G}{\partial x} \right)_{*, t_f} \Delta x(t_f) + \left(\frac{\partial G}{\partial a} \right)_{*, t_f} \Delta a + \left(\frac{\partial Z}{\partial \beta} \right)_{*} \Delta \beta \\ & + \int_{t_0}^{t_f} \left[\left(\frac{\partial F}{\partial x} \right)_{*} \Delta \bar{x}(t) + \left(\frac{\partial F}{\partial u} \right)_{*} \Delta \bar{u}(t) + \left(\frac{\partial F}{\partial a} \right)_{*} \Delta \bar{a} \right. \\ & \quad \left. + \lambda^T \left(\left(\frac{\partial f}{\partial x} \right)_{*} \Delta \bar{x}(t) + \left(\frac{\partial f}{\partial u} \right)_{*} \Delta \bar{u}(t) + \left(\frac{\partial f}{\partial a} \right)_{*} \Delta \bar{a} - \Delta \dot{\bar{x}}(t) \right) \right] dt \\ & + (F)_{*, t_f} \Delta t_f - (F)_{*, t_0} \Delta t_0 + o(\varepsilon) \end{aligned} \quad (3.3.16)$$

In (3.3.16) we have used the relations

$$(F)|_{x, t_0} \Delta t_0 = (F)_{*, t_0} \Delta t_0 + o(\varepsilon) \quad (3.3.17a)$$

and

$$(F)|_{x, t_f} \Delta t_f = (F)_{*, t_f} \Delta t_f + o(\varepsilon) \quad (3.3.17b)$$

If we replace the differences by their corresponding variations and combine the higher order terms relative to ε with $o(\varepsilon)$ in (3.3.16), and compare the result with

$$\Delta Q = \delta Q + o(\varepsilon) \quad (3.3.18)$$

we have

$$\begin{aligned} \delta Q = & \left(\frac{\partial G}{\partial t} + F \right)_{*, t_f} \delta t_f + \left(\frac{\partial G}{\partial x} \right)_{*, t_f} \delta x(t_f) - (F)_{*, t_0} \delta t_0 + \left(\frac{\partial Z}{\partial \beta} \right)_{*} \delta \beta \\ & + \int_{t_0}^{t_f} \left[\left(\lambda^T \left(\frac{\partial f}{\partial x} \right)_{*} + \left(\frac{\partial F}{\partial x} \right)_{*} \right) \delta \bar{x} - \lambda^T \delta \dot{\bar{x}} \right] dt \\ & + \int_{t_0}^{t_f} \left[\lambda^T \left(\frac{\partial f}{\partial u} \right)_{*} + \left(\frac{\partial F}{\partial u} \right)_{*} \right] \delta \bar{u} dt \\ & + \left[\int_{t_0}^{t_f} \left(\lambda^T \left(\frac{\partial f}{\partial \alpha} \right)_{*} + \left(\frac{\partial F}{\partial \alpha} \right)_{*} \right) dt + \left(\frac{\partial G}{\partial \alpha} \right)_{*, t_f} \right] \delta \alpha \end{aligned} \quad (3.3.19)$$

where $\delta \bar{x}(t)$ satisfies the variational equation

$$\delta \dot{\bar{x}} = \left(\frac{\partial f}{\partial x} \right)_{*} \delta \bar{x} + \left(\frac{\partial f}{\partial u} \right)_{*} \delta \bar{u} + \left(\frac{\partial f}{\partial \alpha} \right)_{*} \delta \alpha \quad (3.3.20a)$$

with

$$\delta \bar{x}(t_0) = \delta x(t_0) - (\dot{x})_{*, t_0} \delta t_0 \quad (3.3.20b)$$

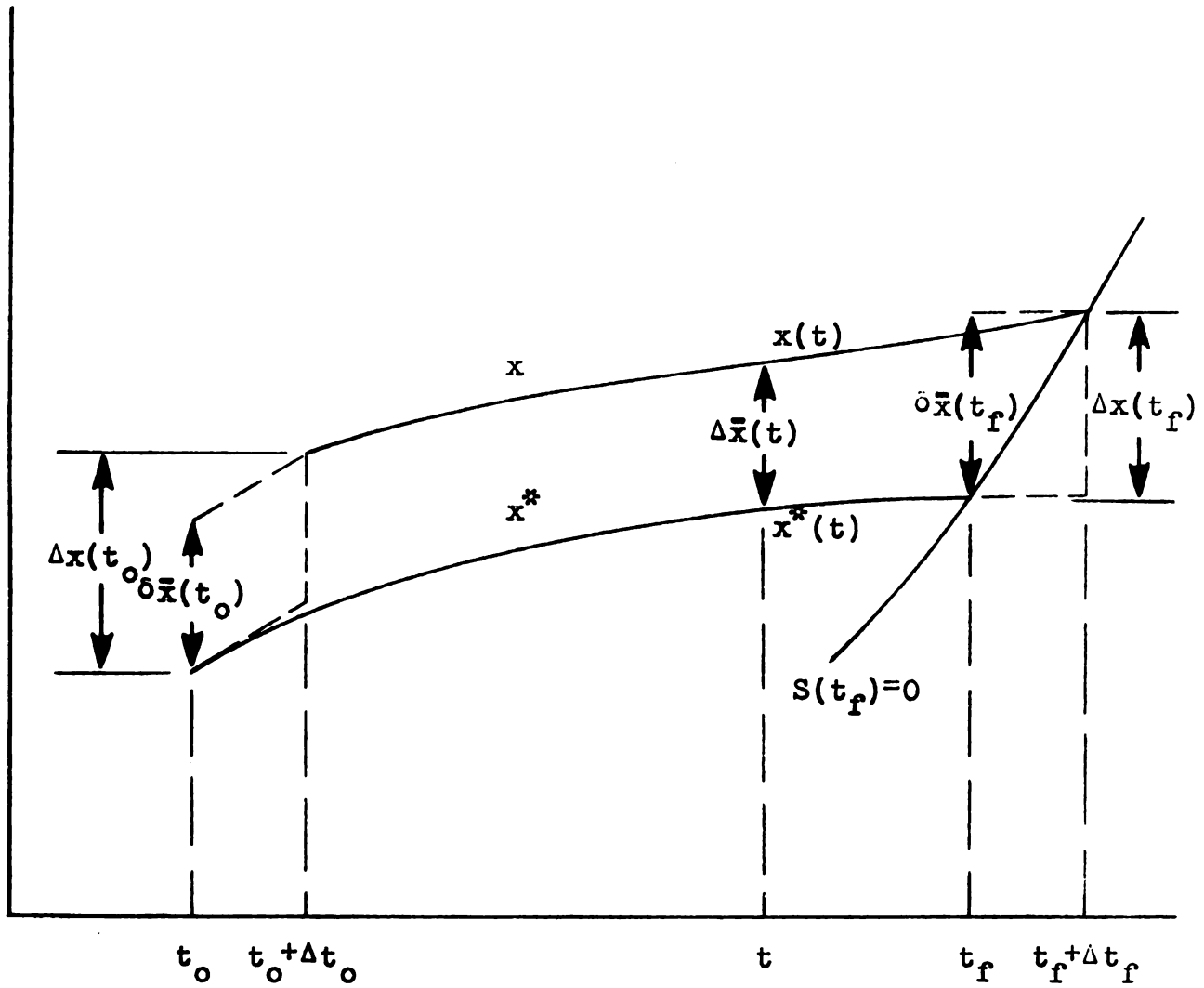


Figure 3 A nominal solution arc and its neighboring arc with variable end-points

$$\delta \bar{x}(t_f) = \delta x(t_f) - (\dot{x})_{*,t_f} \delta t_f \quad (3.3.20c)$$

In deriving (3.3.20), the relations (3.3.17) with f replacing F are used. In this case, the variational equations (3.3.20) and (3.3.11) are equivalent.

Integrating $\int_{t_0}^{t_f} \lambda^T \delta \dot{x} dt$ by parts and choosing $\lambda(t)$ to satisfy the following adjoint equation and the terminal condition

$$\dot{\lambda}^T = - \lambda^T \left(\frac{\partial f}{\partial x} \right)_{*} - \left(\frac{\partial F}{\partial x} \right)_{*} \quad (3.3.21a)$$

$$\lambda^T(t_f) = \left(\frac{\partial G}{\partial x} \right)_{*,t_f} \quad (3.3.21b)$$

and substituting these relations into (3.3.19), we have

$$\begin{aligned} \delta Q = & (\dot{G} + F)_{*,t_f} \delta t_f + (\lambda^T \delta x)_{*,t_0} - (\lambda^T \dot{x} + F)_{*,t_0} \delta t_0 \\ & + \left(\frac{\partial Z}{\partial \beta} \right)_{*} \delta \beta + \int_{t_0}^{t_f} \left[\lambda^T \left(\frac{\partial f}{\partial u} \right)_{*} + \left(\frac{\partial F}{\partial u} \right)_{*} \right] \delta u dt \\ & + \left[\int_{t_0}^{t_f} \left(\lambda^T \left(\frac{\partial f}{\partial a} \right)_{*} + \left(\frac{\partial F}{\partial a} \right)_{*} \right) dt + \left(\frac{\partial G}{\partial a} \right)_{*,t_f} \right] \delta a \end{aligned} \quad (3.3.22)$$

where

$$(\dot{G})_{*,t_f} = \left(\frac{\partial G}{\partial t} + \frac{\partial G}{\partial x} \dot{x} \right)_{*,t_f} = \left(\frac{\partial G}{\partial t} + \lambda^T \dot{x} \right)_{*,t_f}$$

We shall call $\delta \bar{y}$ the variation of y , and δy the total variation of y . Therefore δQ is the total variation of Q

corresponding to the total variations δt_0 , δt_f , $\delta x(t_0)$, $\delta x(t_f)$, $\delta \alpha$ and $\delta \beta$, and the variations $\delta \bar{u}(t)$ and $\delta \bar{x}(t)$. For convenience we shall replace δy by dy , and $\delta \bar{y}$ by δy , thus (3.3.22) and (3.3.20) become, respectively,

$$\begin{aligned} dQ = & (\dot{G} + F)_{*,t_f} dt_f + (\lambda^T dx)_{*,t_0} - (\lambda^T \dot{x} + F)_{*,t_0} dt_0 + \left(\frac{\partial Z}{\partial \beta}\right)_{*} d\beta \\ & + \int_{t_0}^{t_f} \left[\lambda^T \left(\frac{\partial f}{\partial u}\right)_{*} + \left(\frac{\partial F}{\partial u}\right)_{*} \right] \delta u dt \\ & + \left[\int_{t_0}^{t_f} \left(\lambda^T \left(\frac{\partial f}{\partial \alpha}\right)_{*} + \left(\frac{\partial F}{\partial \alpha}\right)_{*} \right) dt + \left(\frac{\partial G}{\partial \alpha}\right)_{*,t_f} \right] d\alpha \end{aligned} \quad (3.3.23)$$

and

$$\delta \dot{x} = \left(\frac{\partial f}{\partial x}\right)_{*} \delta x + \left(\frac{\partial f}{\partial u}\right)_{*} \delta u + \left(\frac{\partial f}{\partial \alpha}\right)_{*} d\alpha \quad (3.3.24a)$$

with

$$\delta x(t_0) = dx(t_0) - (\dot{x})_{*,t_0} dt_0 \quad (3.3.24b)$$

$$\delta x(t_f) = dx(t_f) - (\dot{x})_{*,t_f} dt_f \quad (3.3.24c)$$

The solution of the variational equation (3.3.24) is not needed directly. Instead, the solution of the adjoint equation (3.3.21) is required to evaluate the total variation of Q . A different approach to the derivation of the variation of Q will show clearly that the adjoint variable $\lambda(t)$ defines the variation of Q in terms of the variations or perturbations of the control variable $u(t)$, the control parameters α , β ,

and the initial and final times and states. For this reason, $\lambda(t)$ is frequently called the influence function.

It is also important to point out that the adjoint variable $\lambda(t)$ is a linear combination of the column elements of the transition matrix of the equivalent Mayer problem. Therefore, one may call $\lambda(t)$ the modified transition function [EV-1]. To show this, let $X = (x^0 \ x^T)^T$, where x^0 satisfies

$$\dot{x}^0 = F(t, x, u, a) \quad x^0(t_0) = 0 \quad (3.3.25)$$

The variational equation corresponding to the new state variable X is

$$\dot{\delta X} = \left(\frac{\partial \bar{f}}{\partial X} \right)_* \delta X + \left(\frac{\partial \bar{f}}{\partial u} \right)_* \delta u + \left(\frac{\partial \bar{f}}{\partial a} \right)_* \delta a \quad (3.3.26a)$$

with

$$\delta X(t_0) = dX(t_0) - \dot{X}(t_0) dt_0 \quad (3.3.26b)$$

where

$$\bar{f} = \begin{bmatrix} F \\ f \end{bmatrix} \quad (3.3.27)$$

the corresponding adjoint equation is

$$\dot{\Lambda}^T = - \Lambda^T \left(\frac{\partial \bar{f}}{\partial X} \right)_* \quad (3.3.28a)$$

where

$$\Lambda = \begin{bmatrix} \lambda^0 \\ \lambda \end{bmatrix} \quad (3.3.28b)$$

The transition matrix $\Phi(t, t')$ of (3.3.26) is nonsingular and satisfies the homogeneous linear differential equation (see, for instance, Coddington and Levinson [CO-1]),

$$\dot{\Phi}(t, t') = \left(\frac{\partial \bar{f}}{\partial X} \right)_* \Phi(t, t') \quad \Phi(t, t') = I \quad (3.3.29)$$

Differentiating the inverse of $\Phi(t, t')$, we have

$$\begin{aligned} \frac{d}{dt} \Phi^{-1}(t, t') &= - \Phi^{-1}(t, t') \dot{\Phi}(t, t') \Phi^{-1}(t, t') \\ &= - \Phi^{-1}(t, t') \left(\frac{\partial \bar{f}}{\partial X} \right)_* \end{aligned} \quad (3.3.30)$$

Using the relations

$$\Phi^{-1}(t, t') = \Phi(t', t) \quad (3.3.31a)$$

and

$$\Phi^T(t, t) = \Phi(t, t) = I \quad (3.3.31b)$$

(3.3.30) can be written as

$$\dot{\Phi}^T(t', t) = - \left(\frac{\partial \bar{f}}{\partial X} \right)_*^T \Phi^T(t', t) \quad \Phi^T(t, t) = I \quad (3.3.32)$$

Therefore $\Phi^T(t', t)$ is the transition matrix of the adjoint equation (3.3.28). If we assign t_f for t , and t for t' ,

$\Lambda(t)$ can be expressed as

$$\Lambda(t) = \bar{\Phi}^T(t, t_f) \Lambda(t_f) \quad (3.3.33)$$

or

$$\Lambda^T(t) = \Lambda^T(t_f) \bar{\Phi}(t, t_f) \quad (3.3.34)$$

Thus we have shown that $\Lambda(t)$ (and thus $\lambda(t)$) is a linear combination of the column elements of $\bar{\Phi}$.

Now, we shall return to our main subject. In the expression (3.3.23) for dQ , there is a term which involves dt_f explicitly. This term can be eliminated by using the stopping condition $S(t_f) = 0$. Since S is defined as a stopping function, $S = 0$ is always satisfied at $t = t_f$, and therefore the total variation of S , dS , is zero at t_f . Let $Q = S(t_f)$, and dropping the term involving β in (3.3.23), we have

$$\begin{aligned} dS(t_f) &= (\dot{\theta}^0 + M^0)_{*, t_f} dt_f + (\lambda_S^T dx)_{*, t_0} - (\lambda_S^T \dot{x} + M^0)_{*, t_0} dt_0 \\ &\quad + \int_{t_0}^{t_f} [\lambda_S^T (\frac{\partial f}{\partial u})_{*} + (\frac{\partial M^0}{\partial u})_{*}] \delta u dt \\ &\quad + \int_{t_0}^{t_f} (\lambda_S^T (\frac{\partial f}{\partial \alpha})_{*} + (\frac{\partial M^0}{\partial \alpha})_{*}) dt + \frac{\partial \theta^0}{\partial \alpha} d\alpha = 0 \quad (3.3.35) \end{aligned}$$

Since we have assumed that $\dot{S} \neq 0$, one can solve dt_f from (3.3.35). For convenience we set

$$(\dot{S})_{*,t_f} = (\dot{\theta}^0 + M^0)_{*,t_f} \quad (3.3.36a)$$

$$(\dot{Q})_{*,t_f} = (\dot{G} + F)_{*,t_f} \quad (3.3.36b)$$

Solving for dt_f and substituting it into (3.3.23), we have

$$\begin{aligned} dQ = & \int_{t_0}^{t_f} \left(\frac{\partial H_{QS}}{\partial u} \right)_{*} \delta u dt + \left[\int_{t_0}^{t_f} \left(\frac{\partial H_{QS}}{\partial \alpha} \right)_{*} dt + Y_Q \right] d\alpha \\ & + \left(\frac{\partial Z}{\partial \beta} \right)_{*} d\beta + K_Q db \end{aligned} \quad (3.3.37)$$

where

$$H_{QS} = \lambda_{QS}^T(t) f(t, x, u, \alpha) + F(t, x, u, \alpha) - \left(\frac{\dot{Q}}{\dot{S}} \right)_{*,t_f} M^0(t, x, u, \alpha) \quad (3.3.38)$$

$$\lambda_{QS} = \lambda_Q - \lambda_S \left(\frac{\dot{Q}}{\dot{S}} \right)_{*,t_f}^T \quad (3.3.39)$$

Note that λ_S is the $(n \times 1)$ influence function corresponding to S , and λ_S satisfies the adjoint equation (3.3.21a) and the terminal condition (3.3.21b) with F replaced by M^0 , and G by θ^0 . λ_Q is the influence function corresponding to Q , and λ_Q satisfies (3.3.21). If we assign J to Q , then F is replaced by L , and G by g , and λ_J is an n -vector, H_{JS} is a scalar; on the other hand, λ_I is an $(n \times p)$ -matrix function which satisfies (3.3.21) with F replaced by M , and G by θ , and consequently H_{IS} is a p -vector-valued function, where

$$\left(\frac{\partial H}{\partial u}\right)_{QS}^* = \lambda_{QS}^T \left(\frac{\partial f}{\partial u}\right)_* + \left(\frac{\partial F}{\partial u}\right)_* - \left(\frac{\dot{Q}}{\dot{S}}\right)_{*,t_f} \left(\frac{\partial M^0}{\partial u}\right)_* \quad (3.3.40)$$

$$K_Q = [-(H_{QS})_{*,t_0} \quad \lambda_{QS}(t_0)] \quad (3.3.41)$$

$$db^T = [dt_0 \quad dx^T(t_0)] \quad (3.3.42)$$

$$Y_Q = \left(\frac{\partial G}{\partial \alpha}\right)_{*,t_f} - \left(\frac{\dot{Q}}{\dot{S}}\right)_{*,t_f} \left(\frac{\partial \theta^0}{\partial \alpha}\right)_{*,t_f} \quad (3.3.43)$$

where the partial derivatives with respect to vector variables are defined in the usual way. For example,

$$\left(\frac{\partial L}{\partial u}\right)_* = \left[\frac{\partial L}{\partial u^1} \cdot \cdot \cdot \cdot \cdot \cdot \frac{\partial L}{\partial u^m} \right]_* \quad (3.3.44)$$

$$\left(\frac{\partial M}{\partial \alpha}\right)_* = \left[\begin{array}{cccccc} \frac{\partial M^1}{\partial \alpha^1} & \cdot & \cdot & \cdot & \cdot & \cdot & \frac{\partial M^1}{\partial \alpha^r} \\ \vdots & & & & & & \\ \frac{\partial M^p}{\partial \alpha^1} & \cdot & \cdot & \cdot & \cdot & \cdot & \frac{\partial M^p}{\partial \alpha^r} \end{array} \right]_* \quad (3.3.45)$$

$$\left(\frac{\partial Z}{\partial \beta}\right)_* = \left[\begin{array}{cccccc} 2\beta^1 & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & & & & & \cdot \\ \cdot & & \cdot & & & & \cdot \\ \cdot & & & \cdot & & & \cdot \\ \cdot & & & & \cdot & & \cdot \\ \cdot & & & & & 2\beta^{p'} & \cdot \\ \cdot & & & & & 0 & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \end{array} \right]_* \quad (3.3.46)$$

3.4 Constrained Effort and Successive Optimization, Algorithm I

In the previous section we derived the total variation of the generic quantity Q in terms of the influence functions λ_Q and λ_S , and the variations of controls and initial time and state. If we successively assign J and I to Q , we have, respectively

$$dJ = \int_{t_0}^{t_f} \left(\frac{\partial H_{JS}}{\partial u} \right)_* \delta u dt + \left[\int_{t_0}^{t_f} \left(\frac{\partial H_{JS}}{\partial \alpha} \right)_* dt + Y_J \right] d\alpha + K_J db \quad (3.4.1)$$

$$dI = \int_{t_0}^{t_f} \left(\frac{\partial H_{IS}}{\partial u} \right)_* \delta u dt + \left[\int_{t_0}^{t_f} \left(\frac{\partial H_{IS}}{\partial \alpha} \right)_* dt + Y_I \right] d\alpha \\ + \left(\frac{\partial Z}{\partial \beta} \right)_* d\beta + K_I db \quad (3.4.2)$$

Note that dJ and dI are obtained by adjoining the differential constraint (3.1.2) to J and I and by choosing the adjoint variables λ_J , and λ_I to satisfy the adjoint equations (3.3.21). Thus far J and I are not related except that they are both constrained by the state equation (3.1.2).

In the optimization process we shall derive in this section, dJ is minimized under the constraints dI by choosing $\delta u(t)$, $d\alpha$, $d\beta$ and db optimally in each iteration. Therefore, we are not trying to minimize J with the constraints I directly.

In our previous derivations, perturbations and local linearizations were used. The closeness between the predicted

improvement in J , dJ , and the actual improvement in J , $(J_{\text{new}} - J_{\text{old}})$, and that between the predicted correction in I , dI , and the actual correction $(I_{\text{new}} - I_{\text{old}})$ are heavily dependent upon the degree of validity of these linearizations. In order to avoid excessive error, constraint must be placed on the perturbations $\delta u(t)$, $d\alpha$, $d\beta$, and db . To this end, we shall follow an approach which was first used by Bryson and Denham [BR-1].

Let (dC) be a 'small' positive number, and let $U(t)$, A , B , and W be $m \times m$, $r \times r$, $(1+n) \times (1+n)$, and $p \times p$ symmetric positive definite weighting matrices, respectively. Set

$$(dC)^2 = \int_{t_0}^{t_f} \delta u^T(t) U(t) \delta u(t) dt + d\alpha^T A d\alpha + d\beta^T W d\beta + db^T B db \quad (3.4.3)$$

thus, $(dC)^2$ is the sum of the cumulated weighted squares of the control variations $\delta u(t)$, $d\alpha$, $d\beta$ and db . Since (dC) is a measure of the effort in the (u, α, β, b) -space, we shall call (dC) the weighted control effort or the iterative step-size in the (u, α, β, b) -space—the control effort space.

In order to take account of the constraints dI and $(dC)^2$, adjoin (3.4.2) and (3.4.3) to (3.4.1), to obtain

$$\begin{aligned} dJ = & \int_{t_0}^{t_f} \left[\left(\frac{\partial H}{\partial u} \right)^T_{JS} - h^T \left(\frac{\partial H}{\partial u} \right)^T_{IS} - c \delta u^T U \right] \delta u dt \\ & + \int_{t_0}^{t_f} \left(\left(\frac{\partial H}{\partial \alpha} \right)^T_{JS} - h^T \left(\frac{\partial H}{\partial \alpha} \right)^T_{IS} \right) d\alpha + Y_J - h^T Y_I - c d\alpha^T A d\alpha \end{aligned}$$

$$\begin{aligned}
& - [h^T (\frac{\partial Z}{\partial \beta})_* + c d\beta^T W] d\beta + [K_J - h^T K_I - c d b^T B] db \\
& + h^T dI + c (dC)^2
\end{aligned} \tag{3.4.4}$$

where h is a p -vector multiplier, and c is a scalar multiplier.

The variation of dJ corresponding to the perturbations on δu , $d\alpha$, $d\beta$, and db is

$$\begin{aligned}
\delta(dJ) = & \int_{t_0}^{t_f} [(\frac{\partial H}{\partial u})_{JS}^* - h^T (\frac{\partial H}{\partial u})_{IS}^* - 2c \delta u^T U] \delta^2 u dt \\
& + [\int_{t_0}^{t_f} ((\frac{\partial H}{\partial \alpha})_{JS}^* - h^T (\frac{\partial H}{\partial \alpha})_{IS}^*) dt + Y_J - h^T Y_I - 2c d\alpha^T A] d^2 \alpha \\
& - [h^T (\frac{\partial Z}{\partial \beta})_* + 2c d\beta^T W] d^2 \beta + [K_J - h^T K_I - 2c d b^T B] d^2 b \\
& + h^T \delta(dI) + c \delta(dC)^2
\end{aligned} \tag{3.4.5}$$

The stationarity of dJ due to perturbations in δu , $d\alpha$, $d\beta$, and db implies the vanishing of the variation of dJ , $\delta(dJ)$. By applying a fundamental lemma in the calculus of variations, see, for example, [GE-1], the vanishing of $\delta(dJ)$ implies the vanishing of the coefficient of $\delta^2 u$ for all $t \in [t_0, t_f]$, and the vanishing of the coefficients of $d^2 \alpha$, $d^2 \beta$ and $d^2 b$. Note that dI and $(dC)^2$ are specified quantities. Therefore, the variations of dI and $(dC)^2$ are zero. By equating the coefficients of various quantities to zero, we have, after taking transposes

$$\delta u = \frac{U^{-1}}{2c} \left[\left(\frac{\partial H}{\partial u} \right)_{JS}^T - \left(\frac{\partial H}{\partial u} \right)_{IS}^T h \right] \quad (3.4.6)$$

$$da = \frac{A^{-1}}{2c} \left[\int_{t_0}^{t_f} \left(\left(\frac{\partial H}{\partial a} \right)_{JS}^T - \left(\frac{\partial H}{\partial a} \right)_{IS}^T h \right) dt + Y_J^T - Y_I^T h \right] \quad (3.4.7)$$

$$d\beta = - \frac{W^{-1}}{2c} \left(\frac{\partial Z}{\partial \beta} \right) h \quad (3.4.8)$$

$$db = \frac{B^{-1}}{2c} (K_J^T - K_I^T h) \quad (3.4.9)$$

In order to simplify the expressions, we define the following notations:

$$Q_{II} = U_{II} + A_{II} + B_{II} + W_{II} = Q_{II}^T \quad (p \times p) \quad (3.4.10)$$

$$P_{IJ} = U_{IJ} + A_{IJ} + B_{IJ} = P_{JI}^T \quad (p \times 1) \quad (3.4.11)$$

$$P_{JJ} = U_{JJ} + A_{JJ} + B_{JJ} \quad (\text{scalar}) \quad (3.4.12)$$

$$U_{II} = \int_{t_0}^{t_f} \left(\frac{\partial H}{\partial u} \right)_{IS}^T U^{-1} \left(\frac{\partial H}{\partial u} \right)_{IS}^T dt = U_{II}^T \quad (3.4.13)$$

$$U_{IJ} = \int_{t_0}^{t_f} \left(\frac{\partial H}{\partial u} \right)_{IS}^T U^{-1} \left(\frac{\partial H}{\partial u} \right)_{JS}^T dt = U_{JI}^T \quad (3.4.14)$$

$$U_{JJ} = \int_{t_0}^{t_f} \left(\frac{\partial H}{\partial u} \right)_{JS}^T U^{-1} \left(\frac{\partial H}{\partial u} \right)_{JS}^T dt = U_{JJ}^T \quad (3.4.15)$$

$$A_{II} = R_{IS} A^{-1} R_{IS}^T = A_{II}^T \quad (3.4.16)$$

$$A_{IJ} = R_{IS} A^{-1} R_{JS}^T = A_{JI}^T \quad (3.4.17)$$

$$A_{JJ} = R_{JS} A^{-1} R_{JS}^T = A_{JJ}^T \quad (3.4.18)$$

$$B_{II} = K_I B^{-1} K_I^T = B_{II}^T \quad (3.4.19)$$

$$B_{IJ} = K_I B^{-1} K_J^T = B_{JI}^T \quad (3.4.20)$$

$$B_{JJ} = K_J B^{-1} K_J^T = B_{JJ}^T \quad (3.4.21)$$

$$W_{II} = \left(\frac{\partial Z}{\partial \beta} \right)_* W^{-1} \left(\frac{\partial Z}{\partial \beta} \right)_*^T = W_{II}^T \quad (3.4.22)$$

$$R_{IS} = \int_{t_0}^{t_f} \left(\frac{\partial H_{IS}}{\partial \alpha} \right)_* dt + Y_I \quad (p \times r) \quad (3.4.23)$$

$$R_{JS} = \int_{t_0}^{t_f} \left(\frac{\partial H_{JS}}{\partial \alpha} \right)_* dt + Y_J \quad (1 \times r) \quad (3.4.24)$$

Substituting (3.4.6) to (3.4.9) and their transposes into (3.4.3), and using the notations (3.4.10) to (3.4.24), we have

$$(dC)^2 = \frac{1}{4c} 2 [P_{JJ} - P_{IJ}^T h - h^T P_{IJ} + h^T Q_{II} h] \quad (3.4.25)$$

Similarly, substituting (3.4.6) to (3.4.9) into (3.4.2), and using (3.4.10) to (3.4.24), we have

$$dI = \frac{1}{2c} [P_{IJ} - Q_{II} h] \quad (3.4.26)$$

Solving for h from (3.4.26), we have

$$h = Q_{II}^{-1} [P_{IJ} - 2cdI] \quad (3.4.27)$$

Substituting (3.4.27) and its transpose into (3.4.25), we have

$$(dc)^2 = \frac{1}{4c^2} \left[P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ} \right] + dI^T Q_{II}^{-1} dI \quad (3.4.28)$$

Solve for c using (3.4.28) and assuming that

$$(dc)^2 - dI^T Q_{II}^{-1} dI \neq 0$$

we obtain

$$c = \pm \frac{1}{2} \left[\frac{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}}{(dc)^2 - dI^T Q_{II}^{-1} dI} \right]^{\frac{1}{2}}, \quad (3.4.29)$$

Then substituting c into (3.4.27), we obtain

$$h = Q_{II}^{-1} \left[P_{IJ} - (\pm) \left(\frac{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}}{(dc)^2 - dI^T Q_{II}^{-1} dI} \right)^{\frac{1}{2}} dI \right] \quad (3.4.30)$$

Substituting (3.4.29) and (3.4.30) into (3.4.6) to (3.4.9), we obtain the optimal perturbations.

$$\begin{aligned} \delta u(t) = U^{-1}(t) & \left[\left(\frac{\partial H_{IS}}{\partial u} \right)_*^T Q_{II}^{-1} dI \right. \\ & \left. + \left(\left(\frac{\partial H_{JS}}{\partial u} \right)_*^T - \left(\frac{\partial H_{IS}}{\partial u} \right)_*^T Q_{II}^{-1} P_{IJ} \right) \left(\frac{(dc)^2 - dI^T Q_{II}^{-1} dI}{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}} \right)^{\frac{1}{2}} \right] \end{aligned} \quad (3.4.31)$$

$$\delta \alpha = A^{-1} \left[R_{IS}^T Q_{II}^{-1} dI + \left(R_{JS}^T - R_{IS}^T Q_{II}^{-1} P_{IJ} \right) \left(\frac{(dc)^2 - dI^T Q_{II}^{-1} dI}{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}} \right)^{\frac{1}{2}} \right] \quad (3.4.32)$$

$$d\beta = W^{-1} \left(\frac{\partial Z}{\partial \beta} \right)^T_* Q_{II}^{-1} \left[dI - (\pm) P_{IJ} \left(\frac{(dC)^2 - dI^T Q_{II}^{-1} dI}{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}} \right)^{\frac{1}{2}} \right] \quad (3.4.33)$$

$$db = B^{-1} \left[K_I^T Q_{II}^{-1} dI \pm \left(K_J^T - K_I^T Q_{II}^{-1} P_{IJ} \right) \left(\frac{(dC)^2 - dI^T Q_{II}^{-1} dI}{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}} \right)^{\frac{1}{2}} \right] \quad (3.4.34)$$

Finally, substituting (3.4.31) to (3.4.34) into (3.4.1) we obtain the predicted optimal dJ

$$dJ = P_{IJ}^T Q_{II}^{-1} dI \pm \left((dC)^2 - dI^T Q_{II}^{-1} dI \right)^{\frac{1}{2}} \left(P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ} \right)^{\frac{1}{2}} \quad (3.4.35)$$

Since the quantity

$$\left((dC)^2 - dI^T Q_{II}^{-1} dI \right)^{\frac{1}{2}} \left(P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ} \right)^{\frac{1}{2}}$$

is nonnegative, we have the following rule for choosing the signs in (3.4.31) to (3.4.35):

If J is to be minimized, choose the "-" sign associated with the radical terms in (3.4.31) to (3.4.35); If J is to be maximized, use the "+" sign instead.

Since we are discussing the minimization problem, we shall use the "-" sign in our subsequent development.

If we set $dI = 0$ in (3.4.35), we have

$$dJ = - (dC) (P_{IJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ})^{\frac{1}{2}}$$

or

$$\frac{dJ}{dC} = - (P_{IJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ})^{\frac{1}{2}} \quad (3.4.36)$$

Recalling that J is the cost functional which we want to minimize and that (dC) is the weighted control effort, we therefore, may interpret the right-hand side of (3.4.36) as the negative gradient of J in the function space of control effort.

By examining the optimal perturbations (3.4.31) to (3.4.34), one may recognize that each of the optimal perturbations consists of two parts; one of them is for achieving the error correction dI , and the other part is for the minimization of dJ . The weighting between the two parts depends on how large an isoperimetric error correction dI was specified.

The predicted dJ in (3.4.35) also consists of two parts; the term $P_{IJ}^T Q_{II}^{-1} dI$ is due to the effort made to accomplish the correction of I , dI , and the rest of (3.4.35) is due to the effort made toward minimization of the variation of the cost functional dJ .

3.5 An Iterative Procedure for Algorithm I

The following iterative procedure is suggested:

- (i) Initialization: Guess and store the nominal $u^*(t)$, a^* , β^* and b^* .
- (ii) Forward integration: Integrate the state equation (3.1.2) from t_0 until $S = 0$, and store $x^*(t)$. Also integrate L , M , and M^0 , add to g , $\theta + Z$, and θ^0 at

- t_f , respectively, to obtain J , I , and S .
- (iii) Check the validity of linear approximation. Modify the value of (dC) . Stop the iteration if $dC \leq dC_f$.
 - (iv) Evaluate $(\dot{J}/\dot{S})_{*,t_f}$, $(\dot{I}/\dot{S})_{*,t_f}$, Y_J , and Y_I .
 - (v) Update the symmetric positive definite weighting matrices $U^{-1}(t)$, A^{-1} , B^{-1} , W^{-1} .
 - (vi) Backward integration: Integrate the adjoint equation (3.3.21) from t_f to t_0 to obtain $\lambda_J(t)$, $\lambda_I(t)$, and $\lambda_S(t)$. Simultaneously integrate to obtain U_{II} , U_{IJ} , U_{JJ} , A_{II} , A_{IJ} , and A_{JJ} , and store $(\partial H_{JS}/\partial u)_{*,t}$ and $(\partial H_{IS}/\partial u)_{*,t}$.
 - (vii) Evaluate B_{II} , B_{IJ} , B_{JJ} , and W_{II} to obtain Q_{II} , P_{IJ} and P_{JJ} and invert Q_{II} .
 - (viii) Choose suitable dI and (dC) with $(dC)^2 - dI^T Q_{II} dI \geq 0$.
 - (ix) Compute the optimal $\delta u(t)$, $d\alpha$, $d\beta$, db and the predicted dJ .
 - (x) Simultaneously with (ix) update $u(t)$, α , β , and b , and store both new and old values of $u(t)$, α , β , and b .
 - (xi) Go back to (ii).

3.6 Some Measures for the Validity of Linear Approximation

As mentioned before, this optimization process depends heavily upon linearization. Therefore, it is important to check the validity of this approximation. In this section we shall define some measures of the validity of linear approximation in terms of the predictabilities of dJ

and dI .

Let e_J and e_I be the relative prediction errors of dJ and dI , respectively, and define them as

$$e_J = \frac{dJ - (J_{\text{new}} - J_{\text{old}})}{dJ} \quad (3.6.1)$$

and

$$e_I = \frac{\|V(dI - (I_{\text{new}} - I_{\text{old}}))\|}{\|VdI\|} \quad (3.6.2)$$

where V is a diagonal weighting matrix, and the norm $\|\cdot\|$ is defined as

$$\|v\| = \sup_{i=1, \dots, p} |v^i| \quad (3.6.3)$$

where $v = (v^1 \dots v^p)^T$ is any bounded p -vector.

Let e_{J0} and e_{I0} be specified tolerances for e_J and e_I respectively, then e_J and e_I can be used to define test criteria to determine the step-size for the iterative process.

In practical optimization problems, judicious test criteria must be used, otherwise great computational difficulty may result.

3.7 The Instantaneous Algebraic Constraints and the Penalty Function Technique

So far we have treated the control problem of Bolza with differential constraints and isoperimetric constraints

only, but we have not treated the problem with instantaneous algebraic constraints given in (2.14.3). For convenience, we shall redefine the instantaneous algebraic constraints as,

$$\phi^s(t, x, u, \alpha) \leq 0 \quad s = 1, \dots, q' \quad (3.7.1a)$$

$$\phi^s(t, x, u, \alpha) = 0 \quad s = q'+1, \dots, q \quad (3.7.1b)$$

where ϕ is defined on E and is of $C^1(E)$. And ϕ^s can either be a control variable constraint, or a state variable constraint. Similar problems have been treated analytically by many authors, see, for instance, Gamkrelidze [GA-1], Berkovitz [BE-1,2], Hestenes [HE-3] and Guinn [GU-1]. Problems with state variable constraints are more difficult than those with control variable constraints in both analytical treatment of the problem, and from the computational point of view. Dreyfus [DR-1] considered the control problem of Mayer with terminal equality constraints and K th order state variable inequality constraints, where both the inequality constraint and the control variable are single scalar quantities. Denham and Bryson [DE-1] obtained a steepest-ascent solution to this problem by employing this direct approach to inequality constraints. Denham [DE-2] discussed the case of two inequality constraints with one or two control variables involved. Although this method seems feasible for single constraints and single control variable problems, it becomes more and more involved for systems with increasing numbers of constraints

and control variables. A method known as the penalty function technique was first introduced by Courant and Moser [COU-1] for ordinary minimum problems. Kelley [KE-2] extended the penalty function method to the control problem of Mayer by introducing additional state variables. Okamura [OK-1] applied the penalty function method to the control problem of Lagrange with side constraints and established some mathematical basis for this method.

There are a number of reasons why one would wish to use the penalty function technique: (1) Due to the form of the cost functional of our problem, no alternation is needed in our algorithm, whether or not there are instantaneous algebraic constraints. (2) No additional state variables need to be added to our original system. (3) This method is valid for any finite number of constraints and any finite number of control variables.

Let $P(\emptyset)$ be a vector penalty function for the constraints (3.7.1), the components of $P(\emptyset)$ can be defined as

$$P^s(\emptyset^s) > 0 \quad \text{if} \quad \emptyset^s > 0 \quad \text{for} \quad s = 1, \dots, q' \quad (3.7.2a)$$

$$P^s(\emptyset^s) = 0 \quad \text{if} \quad \emptyset^s \leq 0 \quad \text{for} \quad s = 1, \dots, q' \quad (3.7.2b)$$

$$P^s(\emptyset^s) > 0 \quad \text{if} \quad \emptyset^s \neq 0 \quad \text{for} \quad s = q'+1, \dots, q \quad (3.7.2c)$$

$$P^s(\emptyset^s) = 0 \quad \text{if} \quad \emptyset^s = 0 \quad \text{for} \quad s = q'+1, \dots, q \quad (3.7.2d)$$

where $P^s(\emptyset^s)$ ($s = 1, \dots, q$) are real-valued continuous functions

defined for all finite values of $\emptyset^s$.

Let w be a q -vector, whose components $w^1, \dots, w^q$ are real nonnegative numbers. We shall call w the weighting vector for the vector penalty function $P(\emptyset)$. The weighted penalty function is then defined as $w^T P(\emptyset)$. Let J_p be the penalized cost functional defined as:

$$J_p = J + \int_{t_0}^{t_f} w^T P(\emptyset) dt \quad (3.7.3)$$

or

$$J_p = g + \int_{t_0}^{t_f} [L + w^T P(\emptyset)] dt \quad (3.7.4)$$

If we define

$$L_p = L + w^T P(\emptyset) \quad (3.7.5)$$

then the penalized cost functional J_p can be written as

$$J_p = g + \int_{t_0}^{t_f} L_p dt \quad (3.7.6)$$

Note that J_p and J are of the same form. The quantity

$$\int_{t_0}^{t_f} w^T P(\emptyset) dt$$

is a measure of cumulated violation of the instantaneous algebraic constraints.

By suitably choosing the weighting sequence $\{w\}$ the corresponding sequence $\{J_p\}$ of the approximated control

problem will converge to J of the original problem under some convergence hypotheses on the initial state, terminal time, and control functions associated with the penalized system [OK-1].

Therefore by introducing penalty terms to J , one can apply Algorithm I to the general control problem of Bolza. Further, it should be emphasized that the method applies equally well to problems with control variable constraints, or state variable constraints, or with mixed constraints.

3.8 Conclusion

In this chapter we have first formulated a computational version of the control problem of Bolza. Variations of the various quantities of the problem are derived by using the notion of variable region of integration. An algorithm for the computation of the problem is then developed. A penalty function technique is used to treat the instantaneous algebraic constraints, and additional control parameters are introduced to convert the inequality isoperimetric constraints into equality constraints.

Before closing this chapter, we shall illustrate some advantages of this formulation of the optimal control problem over that of the earlier formulations, such as those given by Bryson and Denham [BR-1], Kelley [KE-1], and Vachino [VAC-1]. The present formulation is based on the problem of Bolza, where the earlier ones are based on the

problem of Mayer. From the theoretical point of view, as demonstrated in Chapter II, the problem of Bolza and that of Mayer are equivalent. However, from the computational point of view, this is not the case. For instance, if one has a control problem of Mayer, the present computational algorithm is readily applicable. On the other hand, if one has a control problem of Bolza with isoperimetric constraints and if the stopping function S also has an integral part, then, in order to use the earlier formulations, one will have to transform the problem by introducing $(p+2)$ additional state variables, and $(p+2)^2$ additional adjoint variables. Thus the number of equations needed to be integrated in each iteration increases from $n \times (p+3) + p+2$ to $(n+p+2) \times (p+3)$, or a net increase of $(p+2)^2$. This will increase the computing time as well as the memory required to store the state history and other quantities. The equations in the present formulation of the problem are not significantly more complex.

In the next chapter we shall identify an important subclass of problems treated in the present chapter; they are so important that a separate chapter is devoted to their study.

IV A SUBCLASS OF THE CONTROL PROBLEM OF BOLZA, ALGORITHM II

The control problem we shall consider in this chapter is a very important subclass of the control problem of Bolza. Since this subclass contains a large number of problems of practical interest, we shall treat this problem separately from the general problem considered in Chapter III, and an algorithm will be derived for the numerical solution of the problem.

Bang-bang control problems and problems with both bang-bang control variables as well as continuous control variables with various constraints are contained in this category.

Denham and Bryson [DE-1] considered a bang-bang control problem with a single control variable, and an a priori specified number of switches of the control variable. Vachino [VAC-2] considered the case of multiple control variables, and without the need of a priori information on the number of switches. We shall consider more general problems and we shall not assume an a priori knowledge of the number of switches.

4.1 Formulation of the Problem

The cost functional is defined as

$$J = g(t_f, x(t_f)) + \int_{t_0}^{t_f} L(t, x, u, v) dt \quad (4.1.1)$$

The state equation (differential constraint) is

$$\dot{x} = f(t, x, u, v) \quad (4.1.2)$$

where $x = (x^1 \dots x^n)^T$ is the state vector, $u = (u^1 \dots u^{m_1})^T$ is the continuous control vector, and $v = (v^1 \dots v^{m_2})^T$ is the discrete control vector. Assume $u(t)$ is piecewise continuous on $[t_0, t_f]$, and $v(t) \in \{k_1, k_2\}$, or $v^j(t) \in \{k_1^j, k_2^j\}$, $j = 1, \dots, m_2$, where k_1^j and k_2^j are any fixed real numbers, and $k_i = (k_i^1, \dots, k_i^{m_2})^T$, $i = 1, 2$. There is no loss of generality in assuming that $k_1^j < k_2^j$ for all $j = 1, \dots, m_2$. The instantaneous algebraic constraints are similarly defined

$$\varnothing^s \leq 0 \quad s = 1, \dots, q' \quad (4.1.3a)$$

$$\varnothing^s = 0 \quad s = q'+1, \dots, q \quad (4.1.3b)$$

where $\varnothing^s = \varnothing^s(t, x, u, v)$. Define the isoperimetric constraints

$$\bar{I}^\gamma \leq 0 \quad \gamma = 1, \dots, p' \quad (4.1.4a)$$

$$\bar{I}^\gamma = 0 \quad \gamma = p'+1, \dots, p \quad (4.1.4b)$$

where

$$\bar{I}^\gamma = \theta^\gamma(t_f, x(t_f)) + \int_{t_0}^{t_f} M^\gamma(t, x, u, v) dt \quad (4.1.4c)$$

The stopping condition is determined from

$$S = 0, \quad (4.1.5a)$$

which defines the terminal time t_f , where

$$S(t_f) = \theta^0(t_f, x(t_f)) + \int_{t_0}^{t_f} M^0(t, x, u, v) dt \quad (4.1.5b)$$

and it is assumed that $\dot{S} \neq 0$ for all $t \in [t_0, t_f]$. The assumptions of continuity properties for the various functions are essentially the same as those defined in Chapter III for the general problem, and notions of admissible quantities are similar to that defined in Chapter III.

As we did in Chapter III, we shall treat the instantaneous algebraic constraints by using the penalty function technique, and we shall transform the inequality isoperimetric constraints to their equivalent equality constraints. Let $\beta = (\beta^1 \dots \beta^{p'})^T$ be an additional control parameter, and let Z be a p -vector with components

$$Z^\gamma = (\beta^\gamma)^2 \quad \gamma = 1, \dots, p' \quad (4.1.6a)$$

$$Z^\gamma = 0 \quad \gamma = p'+1, \dots, p \quad (4.1.6b)$$

Set $I = (I^1 \dots I^p)^T$ and let

$$I^\gamma = \bar{I}^\gamma + Z^\gamma \quad \gamma = 1, \dots, p$$

Then

$$I^\gamma = 0$$

$$\gamma = 1, \dots, p$$

are equivalent to the constraints defined in (4.1.4).

We want to determine the controls $u(t)$, $v(t)$, β , and the corresponding state $x(t)$ such that the cost functional is minimized and the constraints are satisfied.

4.2 A Multiple-stage Formulation of the Problem

The problem formulated in Section 4.1 is a single-stage problem. Due to the discrete nature of the control variable v , it will be convenient to treat the problem as a multiple-stage problem. Since the number of stages is determined by the number of discontinuities of v , and in general the number of the discontinuities is not known for a given problem, we, therefore, have to treat the problem as a variable-multiple-stage problem. The following formulation is similar to that used in [VAC-1].

Let $u(t)$, $t \in [t_0, t_f]$ be a nominal continuous control function, and $v(t)$, $t \in [t_0, t_f]$ be a nominal discrete control variable. Since v (nominal) is discrete, both of its values and switching times have to be specified. Let $t_{r_j}^j$, $r_j = 1, \dots, N_j$, be the nominal switching times for v^j . For convenience, assume $t_0 < t_1^j$ and $t_{N_j}^j < t_f$ for all $j = 1, \dots, m_2$. Therefore t_0 and t_f are not considered as switching times. Let $t(N) = \{t_1, \dots, t_N\}$ be the ordered set of switching times of v with $t_1 < t_2 < \dots < t_N$ and $t_r = t_{r_j}^j$

for at least one $j \in \{1, \dots, m_2\}$, and for some $r_j \in \{1, \dots, N_j\}$. Note that $N \leq N_1 + \dots + N_{m_2}$. With these notations, we shall reformulate the problem in Section 4.1 as the following multiple-stage problem: The cost functional

$$J = g(t_{N+1}, x_{N+1}(t_{N+1})) + \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} L_r(t, x_r, u_r, v_r) dt \quad (4.2.1)$$

where t_{N+1} is the terminal time t_f , and x_r , u_r , v_r are the state vector and the control vectors for the r th stage respectively. The function L_r is defined as

$$L_r(t, x_r, u_r, v_r) = L(t, x, u, v), \quad t \in T_r, \quad r = 1, \dots, N+1 \quad (4.2.2)$$

where $T_r = [t_{r-1}, t_r]$.

The state equation for the r th stage is:

$$\dot{x}_r = f_r(t, x_r, u_r, v_r), \quad t \in T_r, \quad r = 1, \dots, N+1 \quad (4.2.3a)$$

$$x_r(t_{r-1}) = x_{r-1}(t_{r-1}) \quad r = 2, \dots, N+1 \quad (4.2.3b)$$

where

$$f_r(t, x_r, u_r, v_r) = f(t, x, u, v), \quad t \in T_r, \quad r = 1, \dots, N+1 \quad (4.2.4)$$

The isoperimetric constraints are

$$I^\gamma = 0 \quad \gamma = 1, \dots, p \quad (4.2.5a)$$

$$I^\gamma = \theta^\gamma(t_{N+1}, x_{N+1}(t_{N+1})) + Z^\gamma(\beta) + \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} M_r^\gamma(t, x_r, u_r, v_r) dt \quad (4.2.5b)$$

$$Z^\gamma = (\beta^\gamma)^2, \quad \gamma = 1, \dots, p'; \quad Z^\gamma = 0, \quad \gamma = p'+1, \dots, p \quad (4.2.5c)$$

and the stopping condition is

$$S = 0 \quad (4.2.6a)$$

$$S(t_{N+1}) = \theta^0(t_{N+1}, x_{N+1}(t_{N+1})) + \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} M_r^0(t, x_r, u_r, v_r) dt \quad (4.2.6b)$$

where

$$M_r^\gamma(t, x_r, u_r, v_r) = M^\gamma(t, x, u, v), \quad t \in T_r, \quad r = 1, \dots, N+1 \\ \gamma = 0, \dots, p \quad (4.2.7)$$

also assume $\dot{S} \neq 0$ for all $t \in [t_o, t_f]$. Note that v_r^j is constant in T_r for each j and r , and furthermore, v_r^j may be constant in more than one interval.

The problem is to determine the control variables $u(t)$, $v(t)$, the control parameter β , and the corresponding state $x(t)$ such that the cost functional J is minimized and the constraints are satisfied.

We shall divide this problem into two parts. One is to consider the problem with a fixed number of stages, and to obtain a computational algorithm for determining $u(t)$, β , and the switching sequence of $v(t)$. The other part is to

consider the problem of varying the number of stages, especially, to add new stages if it is desired.

4.3 The Variations of the Multiple-stage Problem with Fixed Number of Stages

The variations of quantities of interest can be obtained in a similar way as those obtained in Chapter III. We, therefore, will omit some of the details involved in obtaining the variations for the present problem.

Let Q be the quantity of interest,

$$Q = G(t_{N+1}, x_{N+1}(t_{N+1})) + Z(\beta) + \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} F_r(t, x_r, u_r, v_r) dt \quad (4.3.1)$$

where Q can be J , I , or S . The term $Z(\beta)$ is dropped if we designate J or S for Q .

Let x_r^* be a nominal solution arc of the system (4.2.3). Then

$$x_r^* : t, x_r^*(t), u_r^*(t), v_r^*(t), \beta^*, \quad t \in T_r, \quad r = 1, \dots, N+1 \quad (4.3.2)$$

is the explicit expression for the arc x_r^* . If we perturb the nominal x_r^* by perturbing u_r^* , v_r^* , β^* , $t_0, \dots, t_{N+1}$, and $x_0^*(t_0)$, $x_{N+1}^*(t_{N+1})$, and adjoin the differential constraints (4.1.3) to Q , the corresponding total variation of Q is

$$\begin{aligned}
dQ = & \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} \left[\lambda_r^T \left(\frac{\partial f_r}{\partial u_r} \right)^* + \left(\frac{\partial F_r}{\partial u_r} \right)^* \right] \delta u_r dt \\
& + [\lambda_1^T dx_1 - (\lambda_1^T \dot{x}_1 + F_1) dt]_{*, t_0} + [\dot{G} + F_{N+1}]_{*, t_f} dt_f + \left(\frac{\partial Z}{\partial \beta} \right)^* d\beta \\
& + \sum_{r=1}^N [(\lambda_r^T \dot{x}_r + F_r) - (\lambda_{r+1}^T \dot{x}_{r+1} + F_{r+1})]_{*, t_r} dt_r \quad (4.3.3)
\end{aligned}$$

where we have used the fact that the state $x_r(t)$ is continuous, and therefore, the total variations $dx_r(t_r)$ and $dx_{r+1}(t_r)$ are equal, correct to first order for all $r = 1, \dots, N$. Also δx_r satisfies the variational equation

$$\delta \dot{x}_r = \left(\frac{\partial f_r}{\partial x_r} \right)^* \delta x_r + \left(\frac{\partial f_r}{\partial u_r} \right)^* \delta u_r, \quad t \in T_r, \quad r = 1, \dots, N+1 \quad (4.3.4a)$$

and

$$\delta x_r(t_{r-1}) = dx_r(t_{r-1}) - \dot{x}_r(t_{r-1}) dt_{r-1}, \quad r = 1, \dots, N+1 \quad (4.3.4b)$$

and λ_r satisfies the adjoint equation and the boundary conditions

$$\dot{\lambda}_r^T = - \lambda_r^T \left(\frac{\partial f_r}{\partial x_r} \right)^* - \left(\frac{\partial F_r}{\partial x_r} \right)^*, \quad t \in T_r, \quad r = 1, \dots, N+1 \quad (4.3.5a)$$

$$\lambda_r^T(t_r) = \lambda_{r+1}^T(t_r) \quad r = 1, \dots, N \quad (4.3.5b)$$

$$\lambda_{N+1}^T(t_{N+1}) = \left(\frac{\partial G}{\partial x_{N+1}} \right)^*_{*, t_f} \quad (4.3.5c)$$

Note that the only changes of v_r considered here are the

changes of switching times t_{r-1} and t_r associated with v_r .

Now we shall use the stopping condition to eliminate t_f from dQ . Substitute S for Q in (4.3.3), and set λ_{Sr} for λ_r in (4.3.3) to indicate that λ_r is associated with S . Furthermore, by the same reason employed in Chapter III, setting $dS = 0$, and solving for dt_f , we have

$$\begin{aligned} dt_f = & - \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} \frac{1}{(\dot{S})_{*,t_f}} \left[\lambda_{Sr}^T \left(\frac{\partial f_r}{\partial u_r} \right)^* + \left(\frac{\partial M_r^0}{\partial u_r} \right)^* \right] \delta u_r dt \\ & - \frac{1}{(\dot{S})_{*,t_f}} \left[\lambda_{S1}^T dx_1 - (\lambda_{S1}^T \dot{x}_1 + M_1^0) dt \right]_{*,t_0} \\ & - \sum_{r=1}^N \frac{1}{(\dot{S})_{*,t_f}} \left[(\lambda_{Sr}^T \dot{x}_r + M_r^0) - (\lambda_{Sr+1}^T \dot{x}_{r+1} + M_{r+1}^0) \right]_{*,t_r} dt_r \end{aligned} \quad (4.3.6)$$

where λ_{Sr} satisfies the adjoint equation and boundary condition (4.3.5) with $G = \theta^0$ and $F_r = M_r^0$, and where

$$(\dot{S})_{*,t_f} = [\dot{\theta}^0 + M_{N+1}]_{*,t_f} \quad (4.3.7)$$

Substituting dt_f into dQ , we have

$$\begin{aligned} dQ = & \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} \left(\frac{\partial H_{QSr}}{\partial u_r} \right)^* \delta u_r dt + \sum_{r=1}^N [H_{QSr} - H_{QSr+1}]_{*,t_r} dt \\ & + \left(\frac{\partial Z}{\partial \beta} \right)^* d\beta + K_Q db \end{aligned} \quad (4.3.8)$$

where

$$H_{QSr} = \lambda_{QSr}^T f_r + F_r - \left(\frac{\dot{Q}}{\dot{S}} \right)_{*,t_f} M_r^0, \quad r = 1, \dots, N+1 \quad (4.3.9)$$

$$\lambda_{QSr} = \lambda_{Qr} - \lambda_{Sr} \left(\frac{\dot{Q}}{\dot{S}} \right)_{*,t_f}^T \quad r = 1, \dots, N+1 \quad (4.3.10)$$

$$\left(\frac{\dot{Q}}{\dot{S}} \right)_{*,t_f} = \left(\dot{G} + F_{N+1} \right)_{*,t_f} \quad (4.3.11)$$

$$K_Q = \left[- \left(H_{QS1} \right)_{*,t_0} \quad \lambda_{QS1}^T(t_0) \right] \quad (4.3.12)$$

$$db^T = \left[dt_0 \quad dx_1^T(t_0) \right] \quad (4.3.13)$$

and λ_{Qr} satisfies the adjoint equation and the boundary condition (4.3.5).

4.4 Successive Optimization Process for the Problem with Fixed Number of Stages

Let $U_r(t)$ ($r = 1, \dots, N+1$), V_r ($r = 1, \dots, N$), B and W be $(m_1 \times m_1)$, (1×1) , $(1+n) \times (1+n)$, and $(p' \times p')$ symmetric positive definite weighting matrices, respectively, selected in each iteration to provide suitable weighting and faster convergence. Let (dC) be the step-size in the control effort space. For the same reason as that in Chapter III, set

$$\begin{aligned} (dC)^2 = & \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} \delta u_r^T(t) U_r(t) \delta u_r(t) dt + \sum_{r=1}^N dt_r V_r dt_r \\ & + db^T B db + d\beta^T W d\beta \end{aligned} \quad (4.4.1)$$

If we substitute J and I for Q in (4.3.3), we shall obtain the expressions for dJ and dI respectively. Adjoin the dI and $(dC)^2$ constraints to dJ , the stationarity of the constrained dJ corresponding to variations in δu_r , dt_r , $d\beta$ and db yields the following optimal variations and the predicted optimal value of dJ :

$$\delta u_r(t) = U_r^{-1}(t) \left[\left(\frac{\partial H_{ISr}}{\partial u_r} \right)^T Q_{II}^{-1} dI - \left(\left(\frac{\partial H_{JSr}}{\partial u_r} \right)^T - \left(\frac{\partial H_{ISr}}{\partial u_r} \right)^T Q_{II}^{-1} P_{IJ} \right) \left(\frac{(dC)^2 - dI^T Q_{II}^{-1} dI}{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}} \right)^{\frac{1}{2}} \right]$$

$$t \in T_r, \quad r = 1, \dots, N+1$$

(4.4.2)

$$dt_r = V_r^{-1} \left[(H_{ISr}^T - H_{ISr+1}^T)_{*, t_r} Q_{II}^{-1} dI - ((H_{JSr} - H_{JSr+1})_{*, t_r} - (H_{ISr}^T - H_{ISr+1}^T)_{*, t_r} Q_{II}^{-1} P_{IJ}) \left(\frac{(dC)^2 - dI^T Q_{II}^{-1} dI}{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}} \right)^{\frac{1}{2}} \right]$$

$$r = 1, \dots, N \quad (4.4.3)$$

$$d\beta = W^{-1} \left(\frac{\partial Z}{\partial \beta} \right)^T_{*} Q_{II}^{-1} \left[dI + P_{IJ} \left(\frac{(dC)^2 - dI^T Q_{II}^{-1} dI}{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}} \right)^{\frac{1}{2}} \right] \quad (4.4.4)$$

$$db = B^{-1} \left[K_I^T Q_{II}^{-1} dI - (K_J^T - K_I^T Q_{II}^{-1} P_{IJ}) \left(\frac{(dC)^2 - dI^T Q_{II}^{-1} dI}{P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}} \right)^{\frac{1}{2}} \right]$$

(4.4.5)

$$dJ = P_{IJ}^T Q_{II}^{-1} dI - [(dc)^2 - dI^T Q_{II}^{-1} dI]^{\frac{1}{2}} [P_{JJ} - P_{IJ}^T Q_{II}^{-1} P_{IJ}]^{\frac{1}{2}} \quad (4.4.6)$$

Note that, except for dt_r , the expressions for the optimal $\delta u_r(t)$, $d\beta$, db and dJ are essentially the same as the corresponding ones obtained in Chapter III. Except in the expression for dt_r , we can drop the subscript r and let t run from t_0 to t_f instead of from t_{r-1} to t_r .

We have used the following notations:

$$U_{II} = \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} \left(\frac{\partial H_{ISr}}{\partial u_r} \right) * U_r^{-1} \left(\frac{\partial H_{ISr}}{\partial u_r} \right)^T dt \quad (4.4.7)$$

$$V_{II} = \sum_{r=1}^N (H_{ISr} - H_{ISr+1})_{*,t_r} V_r^{-1} (H_{ISr}^T - H_{ISr+1}^T)_{*,t_r} \quad (4.4.8)$$

and U_{IJ} , U_{JJ} , V_{IJ} , V_{JJ} are defined similarly. And W_{II} , B_{II} , B_{IJ} , B_{JJ} , Q_{II} , P_{IJ} , and P_{JJ} are precisely the same as we defined in Chapter III.

4.5 A Computational Procedure for the Problem with Fixed Number of Stages

For convenience, we shall name the following procedure the Computational Procedure A:

(1) Initialization: Guess and store the nominal

$t(N) = \{t_1, \dots, t_N\}$, $u_r^*(t)$, $v_r^*(t)$, $t \in T_r$, $r = 1, \dots, N+1$, β^* , and b^* .

- (ii) Forward integration: Integrate the state equation (4.2.3) from t_0 until $S = 0$ (i.e., until $t = t_f$), and store $x^*(t)$. Also integrate L , M , and M^0 , and add respectively to g , $\theta + Z$, and θ^0 at t_f to obtain J , I , and S .
- (iii) Except for the first iteration, check to see if the prediction errors are tolerable. Modify the value of (dC) . Stop the iterative process if $dC \leq dC_f$.
- (iv) Evaluate $(\dot{J}/\dot{S})_{*,t_f}$ and $(\dot{I}/\dot{S})_{*,t_f}$ at t_f .
- (v) Update the weighting matrices $U_r^{-1}(t)$, V_r^{-1} , B^{-1} , and W^{-1} .
- (vi) Backward integration: Backward integrate the adjoint equation (4.3.5) from t_f to t_0 obtain λ_{Jr} , λ_{Ir} , and λ_{Sr} , and simultaneously evaluate the integrals and sums to obtain U_{II} , U_{IJ} , U_{JJ} , V_{II} , V_{IJ} and V_{JJ} and store $(\partial H_{JSr}/\partial u_r)_{*,t}$, $(\partial H_{ISr}/\partial u_r)_{*,t}$, $(H_{JSr} - H_{JSr+1})_{*,t_r}$ and $(H_{ISr} - H_{ISr+1})_{*,t_r}$.
- (vii) Evaluate W_{II} , B_{II} , B_{IJ} , and B_{JJ} at t_0 to obtain Q_{II} , P_{IJ} and P_{JJ} . Invert Q_{II} .
- (viii) Choose suitable dI and (dC) .
- (ix) Compute the optimal $\delta u_r(t)$ ($r = 1, \dots, N+1$), dt_r ($r = 1, \dots, N$), $d\beta$, db and the predicted dJ .
- (x) Simultaneously with (ix) update $u_r(t)$, t_r , β , b , and

$$t_{\text{new}}^{(N)} = \left\{ t_{\text{new } 1}, \dots, t_{\text{new } N} \right\}$$

$$T_{\text{new } r} = [t_{\text{new } r-1}, t_{\text{new } r}] = [t_{r-1} + dt_{r-1}, t_r + dt_r]$$

Store both new and old values.

(xi) Go back to (ii).

Note that, although this procedure will not add any new stages to the system, it has the capability to decrease the number of stages by moving consecutive switching times to the same position.

4.6 Optimization Process for Adding a New Stage and the Corresponding Computational Procedure

So far we have only considered the problem with fixed number of stages. In this section, we shall consider the problem of adding new stages. There are two important questions we have to answer in this section: first, "Should we add a new stage?", and secondly "If the answer is yes, where shall we add?".

For simplicity, we shall consider the case of adding one stage at a time, and we shall not consider the isoperimetric constraints (4.2.5). The generalization of this simple case to a more general treatment is not difficult and shall not be considered here.

The unperturbed system (4.2.3) with $N + 1$ stages is called the nominal system. x^* is used to denote the nominal solution arc of (4.2.3), and $u_r(t)$ will be held unperturbed for all $r = 1, \dots, N + 1$.

Let a subarc (or stage) be added to the nominal arc at $t = t'$ with duration dt' , where

$$[t', t' + dt'] \subset (t_{r'-1}, t_{r'}) , \quad r' \in \{1, \dots, N+1\} \quad (4.6.1)$$

The perturbed system can be expressed as

$$\dot{x}'_r = f'_r \quad t \in T_r \quad r = 1, \dots, N+1 \quad (4.6.2a)$$

$$x_r(t_{r-1}) = x_{r-1}(t_{r-1}) \quad r = 2, \dots, N+1 \quad (4.6.2b)$$

where f'_r is defined as

$$f'_r = f_r(t, x_r, u_r, v_r) , \quad t \in T_r , \quad r = 1, \dots, r'-1 \quad (4.6.3a)$$

$$f'_r = f'_{r'} , \quad t \in T_{r'} \quad (4.6.3b)$$

$$f'_r = f_r(t, x'_r, u_r, v_r) , \quad t \in T_r , \quad r = r'+1, \dots, N+1 \quad (4.6.3c)$$

and $f'_{r'}$ is

$$f'_{r'} = f_{r'}(t, x_{r'}, u_{r'}, v_{r'}) \quad t \in [t_{r'-1}, t') \quad (4.6.4a)$$

$$f'_{r'} = f_{r'}(t, x_{r'}, u_{r'}, v'_{r'}) \quad t \in [t', t' + dt'] \quad (4.6.4b)$$

$$f'_{r'} = f_{r'}(t, x'_{r'}, u_{r'}, v_{r'}) \quad t \in (t' + dt', t_{r'}] \quad (4.6.4c)$$

Let Q be the quantity of interest defined in (4.3.1). As a first order effect, the partial variations of f_r and F_r with respect to the variation of v_r due to the addition of a

subarc can be considered as impulses occurring at t' with strengths

$$[f'_{r'} - f_{r'}]_{t'} dt' = [f_{r'}(t, x_{r'}, u_{r'}, v'_{r'}) - f_{r'}(t, x_{r'}, u_{r'}, v_{r'})]_{t'} dt' \quad (4.6.5)$$

and

$$[F'_{r'} - F_{r'}]_{t'} dt' = [F_{r'}(t, x_{r'}, u_{r'}, v'_{r'}) - F_{r'}(t, x_{r'}, u_{r'}, v_{r'})]_{t'} dt' \quad (4.6.6)$$

where $F'_{r'}$ is defined by replacing f by F in (4.6.4).

After adjoining the system (4.2.3) to Q , the total variation of Q is

$$\begin{aligned} dQ = & \sum_{r=1}^{N+1} \int_{t_{r-1}}^{t_r} \left[\left(\frac{\partial F}{\partial x_r} \right) \delta x_r + \lambda_r^T \left(\left(\frac{\partial f}{\partial x_r} \right) \delta x_r - \delta \dot{x}_r \right) \right] dt \\ & + \left[\left(\frac{\partial G}{\partial t} \right) dt + \left(\frac{\partial G}{\partial x_{N+1}} \right) dx_{N+1} \right]_{*, t_f} - [F_1 dt]_{*, t_0} \\ & + \sum_{r=1}^N [F_r - F_{r+1}]_{*, t_r} dt_r + [F_{N+1} dt]_{*, t_f} \\ & + [\lambda_{r'}^T (f'_{r'} - f_{r'}) + (F'_{r'} - F_{r'})]_{t'} dt' \end{aligned} \quad (4.6.7)$$

Integrating $\int_{t_{r-1}}^{t_r} \lambda_r^T \delta \dot{x}_r dt$ by parts and choosing λ to satisfy the adjoint equation and the terminal condition (4.3.5),

$$\begin{aligned}
dQ = & [\dot{G} + F_{N+1}]_{*,t_f} dt_f + [\lambda_1^T dx_1 - (\lambda_1^T \dot{x}_1 + F_1) dt]_{*,t_0} \\
& + \sum_{r=1}^N [(\lambda_r^T f_r + F_r) - (\lambda_{r+1}^T f_{r+1} + F_{r+1})]_{*,t_r} dt_r \\
& + [\lambda_{r'}^T (f_{r'}' - f_{r'}) + (F_{r'}' - F_{r'})]_{t'} dt' \quad (4.6.8)
\end{aligned}$$

If we substitute S and J for Q in (4.6.8), set dS to zero, solve for dt_f from $dS = 0$ and then substitute dt_f into dJ, we have

$$\begin{aligned}
dJ = & K_J db + \sum_{r=1}^N [H_{JSr} - H_{JSr+1}]_{*,t_r} dt_r \\
& + [H'_{JSr'} - H_{JSr'}]_{t'} dt' \quad (4.6.9)
\end{aligned}$$

where H_{JSr} , K_J and db are defined in (4.3.9), (4.3.12), and (4.3.13), respectively, with Q replaced by J. $H'_{JSr'}$ is defined as

$$H'_{JSr'} = \lambda_{JSr'}^T f_{r'}' + L_{r'}' - \left(\frac{\dot{J}}{\dot{S}} \right)_{*,t_f} M_{r'}^{O'} \quad (4.6.10)$$

where λ_{JSr} , $(\dot{J})_{*,t_f}$, $(\dot{S})_{*,t_f}$ are defined in (4.3.10), (4.3.11), and (4.3.7) respectively.

It is understood that $f_{r'}$, $L_{r'}$, and $M_{r'}^O$ at t' are evaluated on x^* , furthermore, $x_{r'}(t')$, $u_{r'}(t')$ in $f_{r'}'$, $L_{r'}'$, and $M_{r'}^{O'}$ are also on x^* , but $v_{r'}'(t')$ is different from $v_{r'}(t')$.

In the iterative procedure we shall give later, the process will be iterated a number of times to minimize dJ with optimal choices in δu_r , dt_r , $d\beta$, and db , and the number of stages will be held fixed (with possible decreases) during the iterations. Then the process will change its phase to determine whether and how to add a new stage.

In order to make the process simple and workable, we shall hold u_r , t_r , and b fixed in the process of adding a new stage. Setting dt_r and db to zero, (4.6.9) becomes

$$dJ = [H'_{JSr'} - H_{JSr'}]_{t'} dt' = \Delta H_{JSr'}(t') dt' \quad (4.6.11)$$

Choosing dt' positive, the minimization of dJ is equivalent to the minimization of $\Delta H_{JSr'}(t')$. Note that if v is not changed or if there is no stage to be added, then $\Delta H_{JSr'} = 0$. Therefore any new value for v which will minimize dJ must make $\Delta H_{JSr'}$ negative.

Let Δt be a selected step-size for the evaluations of $\Delta H_{JSr'}$ (Δt may be chosen to be the same as the integration step-size or greater), and choose dt' the same size as the integration step-size.

The following procedure is called the Computational Procedure B:

- (i) Start from $r' = r_1$ and $t' = t_0$ up to $r' = N+1$, and $t' \leq t_f - dt'$.
- (ii) (a) At t' assign a value to v , call it v' .

- (b) Evaluate $\Delta H_{JSr'}(t')$.
 - (c) Compare $\Delta H_{JSr'}(t')$ with different values of $v_{r'}'(t')$ for fixed t' . Store the minimum $\Delta \bar{H}_{JSr'}(t')$, and the corresponding $\bar{v}_{r'}(t')$.
 - (d) Compare the result of (c) with the Best-up-to-date, store the smaller and call it the Best-up-to-date, and denote it by $\Delta \bar{H}_{JS\bar{r}'}(\bar{t}')$. Also store the corresponding $\bar{v}_{\bar{r}'}(\bar{t}')$.
- (iii) (a) Increase t' by Δt , call it t' .
- (b) Repeat (ii) for the new t' .
- (iv) Repeat (iii) until t' reaches $t_f - dt'$.
- (v) (a) If $\Delta \bar{H}_{JS\bar{r}'}(\bar{t}')$ is zero, then make no change of v .
- (b) If $\Delta \bar{H}_{JS\bar{r}'}(\bar{t}')$ is negative, then add a new stage to the system after the r' th stage with
- $$T_{\bar{r}'+1} = [t_{\bar{r}'}, t_{\bar{r}'+1}] = [\bar{t}', \bar{t}' + dt']$$
- (c) Change the number N accordingly and again call it N , and reorder the switching sequence as
- $$t(N) = \{t_1, \dots, t_{r'}, t_{r'+1}, \dots, t_{N+1}\}$$
- (d) Relabel the system equation and other functions

4.7 A Computational Procedure for Algorithm II

The overall computational procedure for Algorithm II can be obtained by repeatedly executing Procedures A and B and skipping Block (i) of Procedure A:

- (i) Guess and store the nominal $t(N)$, $u_r^*(t)$, $v_r^*(t)$, β^* , and b^* .

- (ii) Execute Procedure A (skip (i) of A) for a given number of times.
- (iii) Execute Procedure B once.
- (iv) Go back to (ii).

V NUMERICAL EXAMPLES

Three examples are given in this chapter to illustrate the application of the computational technique: (1) Brachistochrone problem with inequality state constraint; (2) Orbit transfer of a solar sail ship; (3) Low thrust trajectory optimization problem. In each of the examples, alternative choices of dC and automatic schemes for updating dC and dI are discussed.

A FORTRAN program for Algorithm I is presented in APPENDIX A. The program is written in such a way that for each specific problem one has only to rewrite the subroutines corresponding to the given problem; however, the major part of the main program will remain the same.

The computer used for the preparation of the numerical data is a CDC-6500 digital computer.

5.1 Brachistochrone Problem with inequality State Constraint

The classical brachistochrone problem was first proposed by Johann Bernoulli in 1696. This problem can be stated as follows: Find the path of a heavy particle falling, under the influence of constant gravitational field, from a given initial point to a final point on a specified vertical line in minimum time. The initial point is not contained in the vertical line.

We shall modify the classical problem so that the path of the particle will be restricted in a given region. The motion of the particle can be described by the differential equations:

$$\dot{x}^1 = x^3 \cos u = f^1 \quad (5.1.1a)$$

$$\dot{x}^2 = x^3 \sin u = f^2 \quad (5.1.1b)$$

$$\dot{x}^3 = g \sin u = f^3 \quad (5.1.1c)$$

where x^1 is the horizontal displacement (feet), x^2 is the vertical displacement (feet, directed downward), x^3 is the velocity of the particle (feet/second), u is the path angle in radians, that is, the angle between the horizontal axis and the path at time t . We shall call $x = (x^1 \ x^2 \ x^3)^T$ the state variable (vector), and u the control variable. Here g is the gravitational acceleration (see Figure 4). The constraint is

$$\phi = x^2 - \frac{1}{2} x^1 - 1 \leq 0 \quad (5.1.2)$$

This problem has been studied by many authors, for instance, Dreyfus [DR-1], Denham [DE-2], Bryson et al. [BR-2], and McGill [MC-1]. In order to compare results with other authors, the following data are used and they were used in [DR-1]:

$$t_0 = 0$$

$$x^1(0) = x^2(0) = 0$$

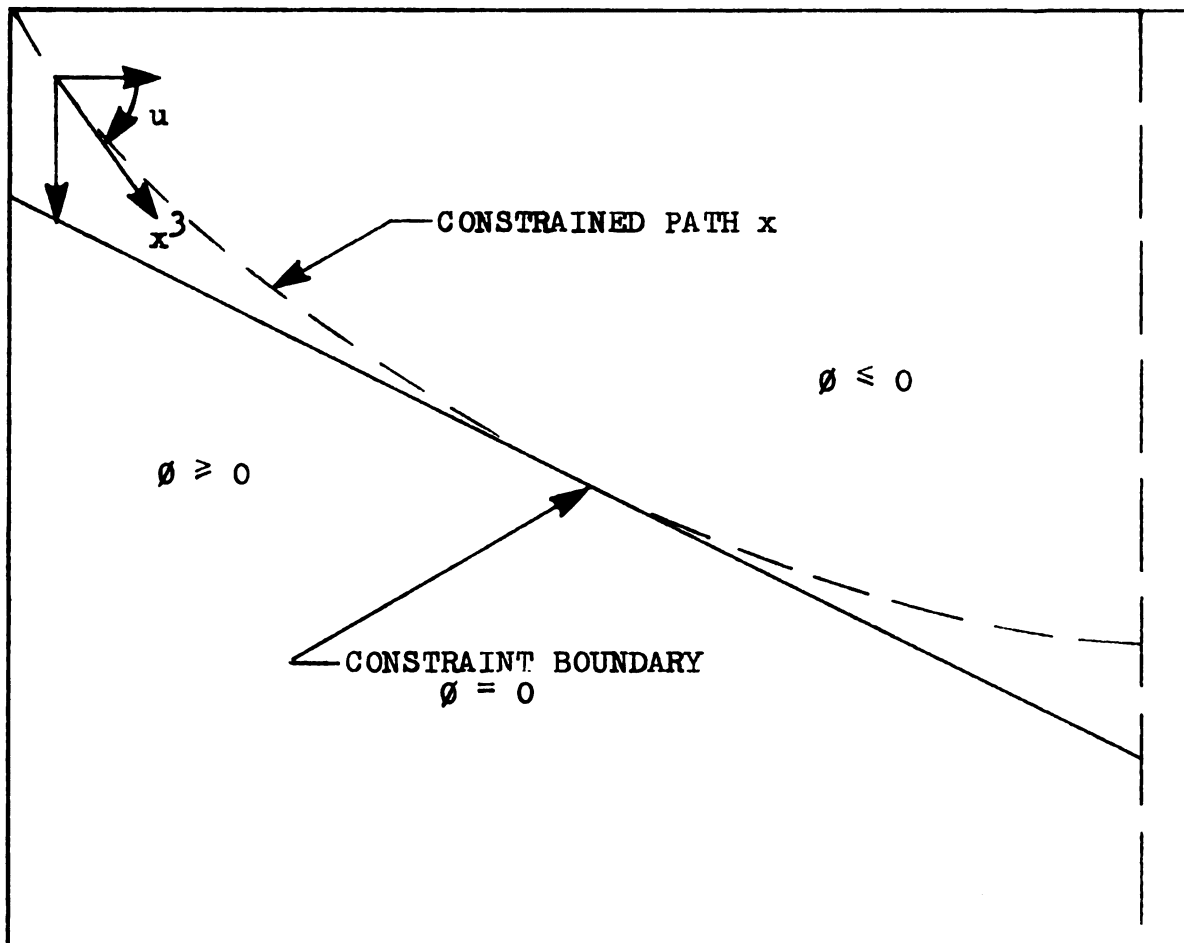


Figure 4 Brachistochrone problem

$$x^3(0) = 1 \quad \text{feet/second}$$

$$x^1(t_f) = 6 \quad \text{feet}$$

$$g = 32.2 \quad \text{feet/second/second}$$

The cost functional J is the terminal time

$$J = t_f \quad (5.1.3)$$

and the stopping functional is

$$S(t_f) = x^1(t_f) - 6 \quad (5.1.4)$$

where $S = 0$ is the stopping condition which defines the terminal time t_f . Instead of minimizing J directly, we shall minimize the penalized cost functional $\bar{J}$ defined by

$$\begin{aligned} \bar{J} &= J + \int_0^{t_f} wP(\varnothing)dt \\ &= t_f + \int_0^{t_f} wP(\varnothing)dt \end{aligned} \quad (5.1.5)$$

where w is a positive weighting coefficient, and $P(.)$ is the penalty function

$$P(\varnothing) = (\varnothing)^2 H(\varnothing) \quad (5.1.6)$$

$H(.)$ is defined by

$$H(\varnothing) = 1 \quad \text{if } \varnothing > 0 \quad (5.1.7a)$$

$$H(\varnothing) = 0 \quad \text{otherwise} \quad (5.1.7b)$$

In order to apply Algorithm I, we identify the following

$$g(t_f, x(t_f)) = t_f \quad (5.1.8)$$

$$L(t, x, u) = w(\emptyset)^2 H(\emptyset) \quad (5.1.9)$$

$$\theta^0(t_f, x(t_f)) = x^1(t_f) - 6 \quad (5.1.10)$$

and

$$M^0(t, x, u) = 0 \quad (5.1.11)$$

Applying the formulas derived in Chapter III formally, one obtains the following adjoint equations:

$$\dot{\lambda}_J^1 = w \emptyset H(\emptyset) \quad \lambda_J^1(t_f) = 0 \quad (5.1.12a)$$

$$\dot{\lambda}_J^2 = -2w \emptyset H(\emptyset) \quad \lambda_J^2(t_f) = 0 \quad (5.1.12b)$$

$$\dot{\lambda}_J^3 = -\lambda_J^1 \cos u - \lambda_J^2 \sin u \quad \lambda_J^3(t_f) = 0 \quad (5.1.12c)$$

$$\dot{\lambda}_S^1 = 0 \quad \lambda_S^1(t_f) = 1 \quad (5.1.13a)$$

$$\dot{\lambda}_S^2 = 0 \quad \lambda_S^2(t_f) = 0 \quad (5.1.13b)$$

$$\dot{\lambda}_S^3 = -\lambda_S^1 \cos u - \lambda_S^2 \sin u \quad \lambda_S^3(t_f) = 0 \quad (5.1.13c)$$

and

$$\lambda_{JS} = \lambda_J - \lambda_S \left(\frac{\dot{g} + L}{\dot{s}} \right)_{*, t_f} \quad (5.1.14)$$

$$H_{JS} = \lambda_{JS}^T f + L \quad (5.1.15)$$

where

$$(\dot{g})_{*,t_f} = 1 \quad (5.1.16)$$

$$(\dot{s})_{*,t_f} = (x^3 \cos u)_{*,t_f} \quad (5.1.17)$$

$$r = (r^1 \ r^2 \ r^3)^T \quad (5.1.18)$$

The constraint on the effort space, the predicted optimal improvement in $\bar{J}$, and the optimal variation in u are, respectively

$$(dC)^2 = \int_0^{t_f} U(t) (\delta u(t))^2 dt \quad (5.1.19)$$

$$d\bar{J} = - (dC)(U_{JJ})^{\frac{1}{2}} \quad (5.1.20)$$

$$\delta u(t) = - U^{-1}(t) \left(\frac{\partial H_{JS}}{\partial u} \right) (dC)(U_{JJ})^{-\frac{1}{2}} \quad (5.1.21)$$

where

$$U_{JJ} = \int_0^{t_f} U^{-1}(t) \left(\frac{\partial H_{JS}}{\partial u} \right)^2 dt \quad (5.1.22)$$

Before one can obtain a numerical solution of the problem, one has to face a number of difficult problems, such as, how to choose the iterative step-size dC , the weighting matrix (scalar here) U , the weighting factor w for the penalty function, and the integration step-size. There are no straight forward answers to these questions; they can only be found by guess and experiment. Since dC

is a measure of control effort, it has direct control on the progress of the iterative process as well as the linearization error. Therefore, the value of dC must be updated according to iteration number and error test. It is possible in some iterations that excessive linearization errors are observed. Then one may have to go back to the previous iteration, and therefore, both new and old control functions have to be stored to provide suitable back-up capability. The sample program presented in APPENDIX A is fully automatic; there is no operator interface required.

In obtaining the following numerical results, the same integration step-size (.001 second), and weighting factor for penalty function (100) were used along with different schemes in updating dC and the alternative choices of the weighting matrix U . An initial guess for the control function $u(t) = \pi/6$ (30°), $t \in [0,1]$ was used throughout the computations.

Case I: (1) $dC = .1$; (2) $U^{-1} = 1$; (3) In each iteration, a linearization test was performed. If the prediction error (due to linear approximation) exceeded the prescribed tolerance, dC was then reduced to 70%. (4) Subroutine SELECT was called by the main program once in each iteration, and dC was then reduced to 70% if $1 \leq N \leq 10$, and no effect if $N > 10$.

Figure 6 shows the terminal time t_f as a function of iteration number N . In Figure 6, $N = 0$ is the initial

iteration, ○ indicates that there was no violation of the constraint boundary, and ● indicates that the state constraint was violated in that iteration. After 26 iterations (including the initial iteration), the terminal time dropped from .86767 second at $N = 0$ to .74589 second at $N = 25$. In Figure 5, several of the descent paths corresponding to $N = 0, 1, 2, 25$ were plotted. The average computer time was 4.389 seconds/iteration.

Case II: (1) $dC = .2$; (2) $U^{-1} = a + (b - a) \left(\frac{t - .5t_2}{.5t_2} \right)^2$, $t \in [0, 1]$, $t_2 = 1$, $a = 2/3$, $b = 3/2$; (3) Same as that in I(3) except 95% instead of 70%; (4) Same as that in I(4).

With these changes in dC , U^{-1} , and the updating process, the terminal time decreased more rapidly than that of Case I. As shown in Figure 7, t_f decreased from .86767 second at $N = 0$ to .74478 second at $N = 24$. The average computer time was 4.423 seconds/iteration.

Case III: (1) $dC = .2$; $U^{-1} = \frac{1}{at+b}$, $t \in [0, 1]$, $a = -1.5$, $b = 2$; (3) Same as that in I(3) except 80% instead of 70%; (4) In subroutine SELECT, dC was reduced to 90% if $1 \leq N \leq 10$, 85% if $N > 25$, and no effect otherwise.

After 25 iterations with 4.399 seconds/iteration average execution time, the terminal time dropped from .86767 second to .74289 second.

Case IV: (1) $dC = .2$; (2) $U^{-1} = 1$; (3) Same as that in III(3); (4) Same as that in III(4).

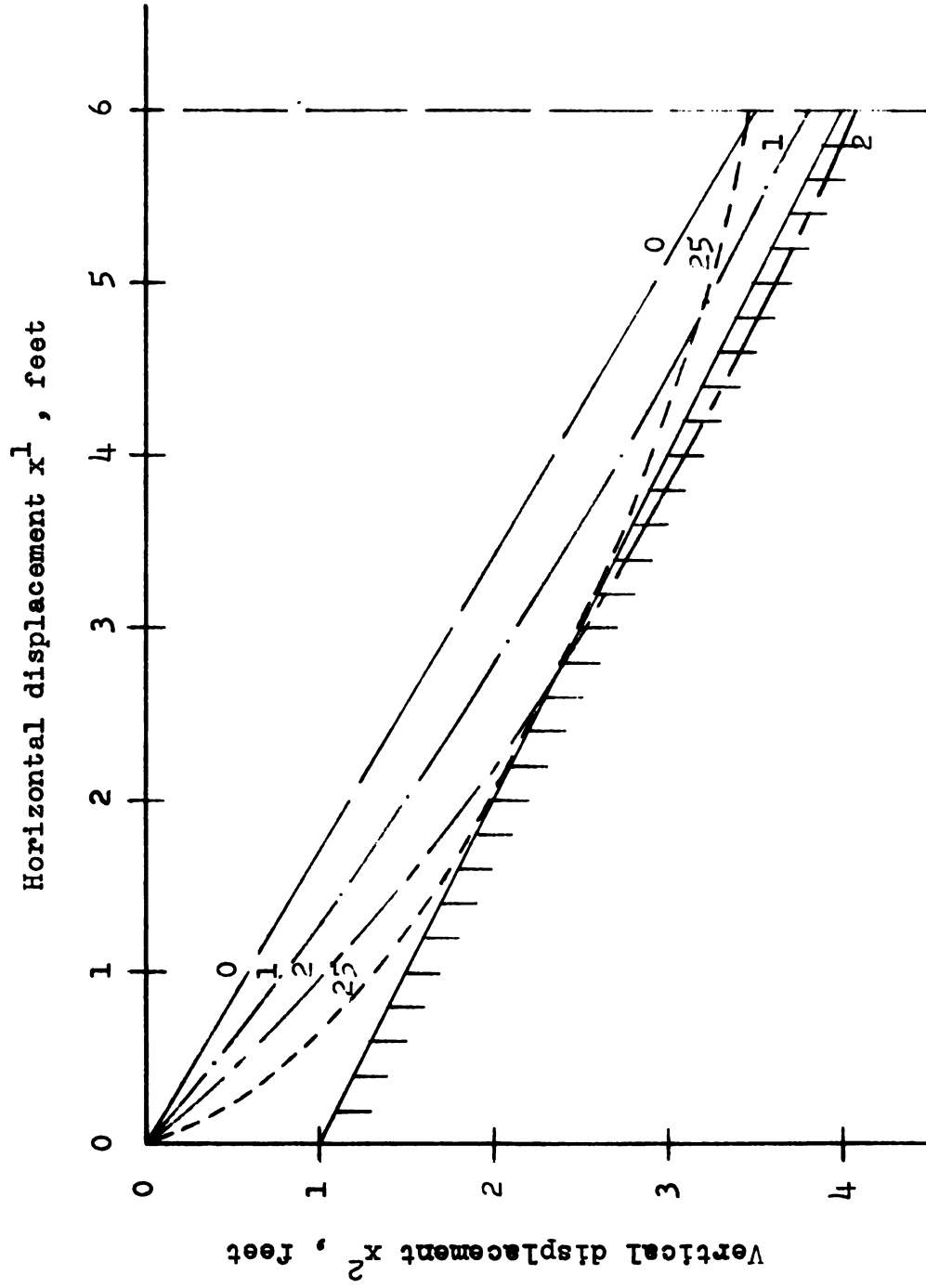
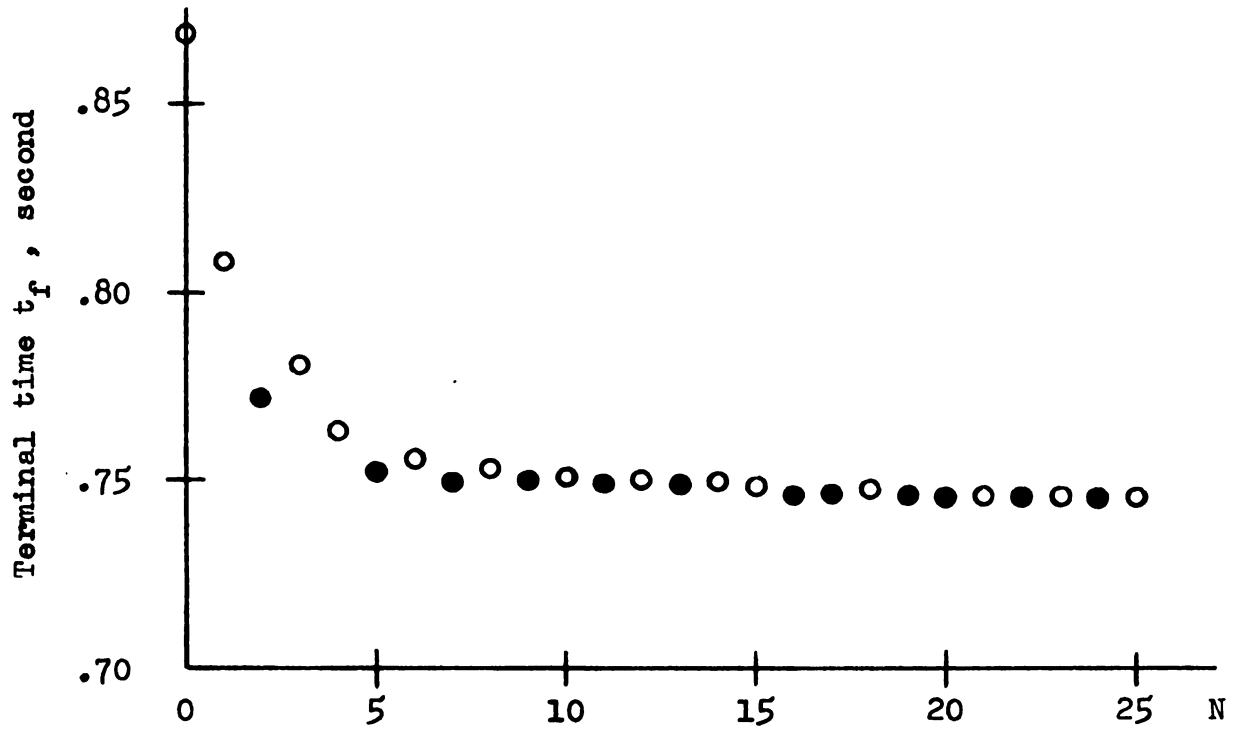
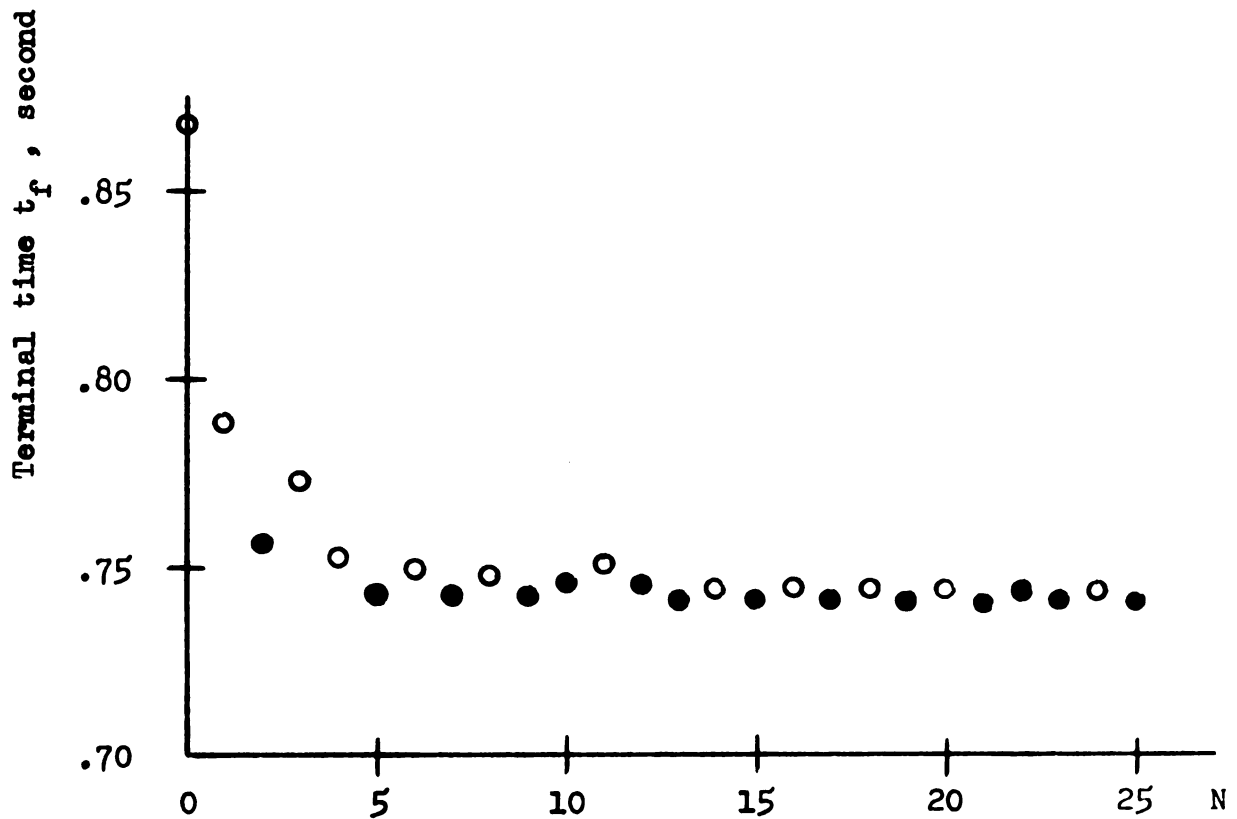


Figure 5 Paths of brachistochrone problem - Case I

Figure 6 t_f vs. N , brachistochrone - Case IFigure 7 t_f vs. N , brachistochrone - Case II

With this combination the terminal time dropped to .74221 second after 26 iterations ($N = 25$), .74207 after 37 iterations, and .74199 second after 43 iterations ($N = 42$) with an average computer time 4.425 second/iteration (or 3.171 minutes for 43 iterations on CDC-6500 computer). This result agrees with that obtained by Dreyfus [DR-1], in his work, after 31 iterations, the terminal time was .7422 second, and after a total of 50 iterations and 4 corner modifications the terminal time dropped to .7420 second with 10 minutes of computer time (IBM-7090 computer).

Figure 8 shows the terminal time t_f as a function of iteration number N , $N = 0, 1, \dots, 25$, for Case IV. Several control functions obtained during the iterative process corresponding to the iteration numbers 0, 1, 3, 25, 42 are plotted in Figure 9, the corresponding paths of the particle are presented in Figure 10. Here $N = 0$ represents the initial iteration, $N = 1, 3, 25$ correspond to the intermediate solutions, and $N = 42$ corresponds to the computed optimal solution.

From the results presented in this section, we can draw the following conclusions:

- (1) The values of dC and U and the updating scheme have direct influence on the speed of convergence of the optimization process. As one can see from the results, Case IV is superior to Case III, Case III is superior to Case II, and Case II is superior to Case I.

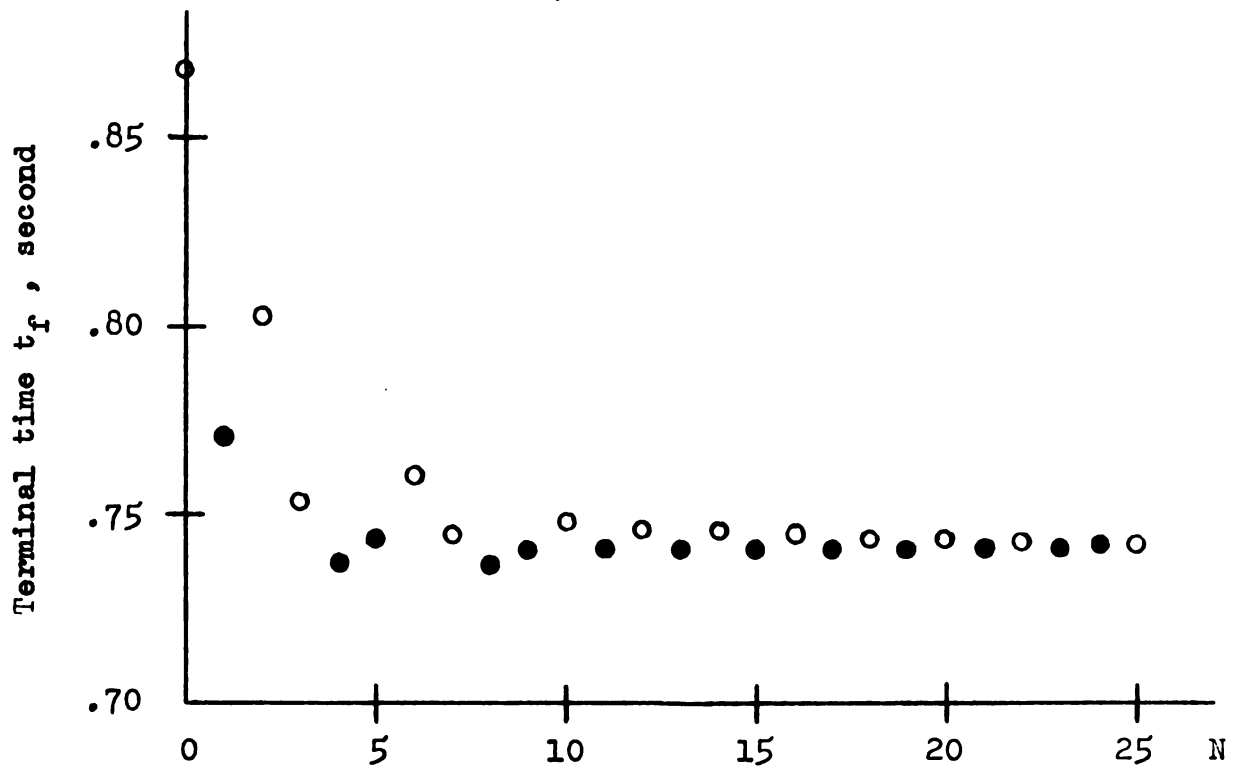


Figure 8 t_f vs. N , brachistochrone - Case IV

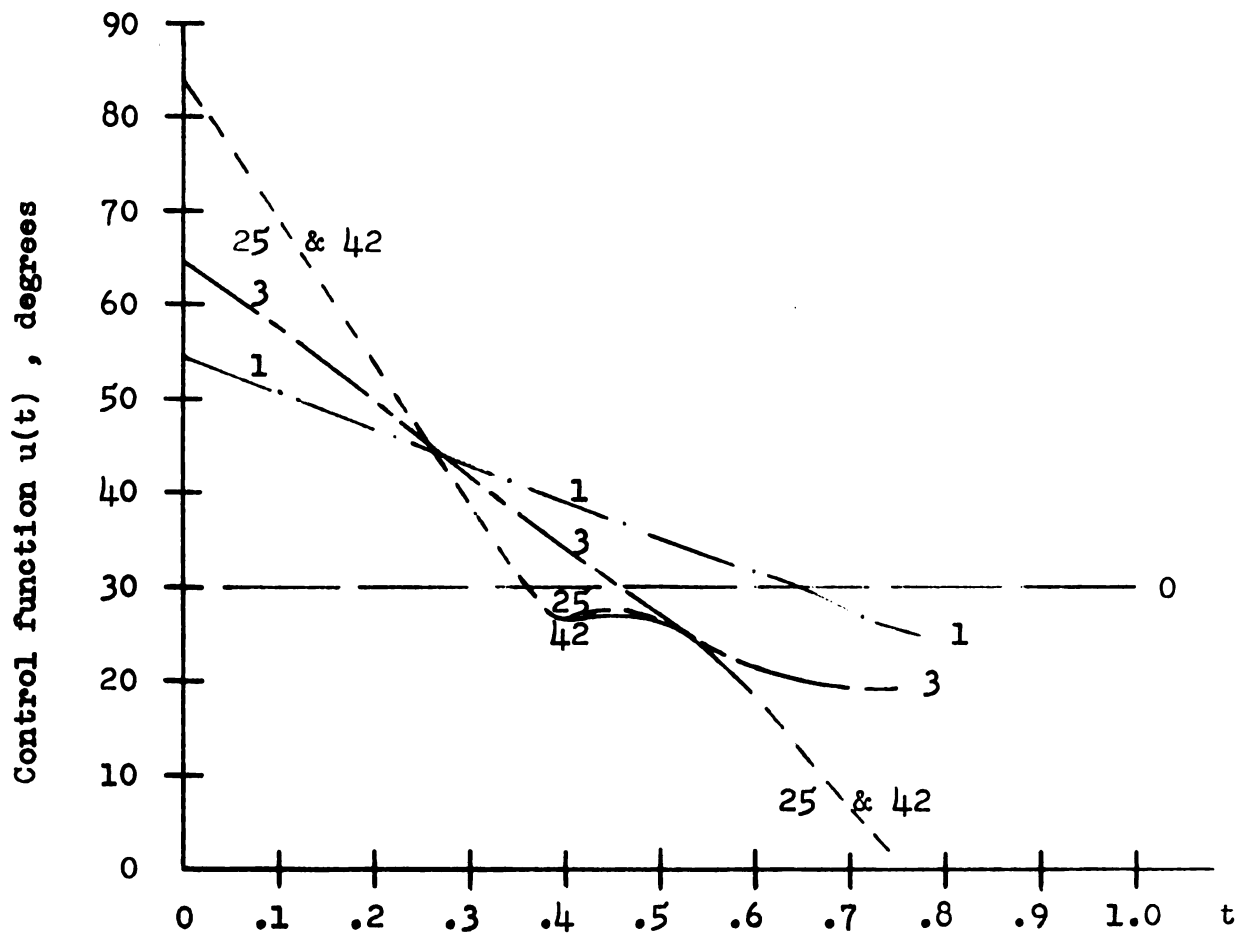


Figure 9 Control functions, brachistochrone - Case IV

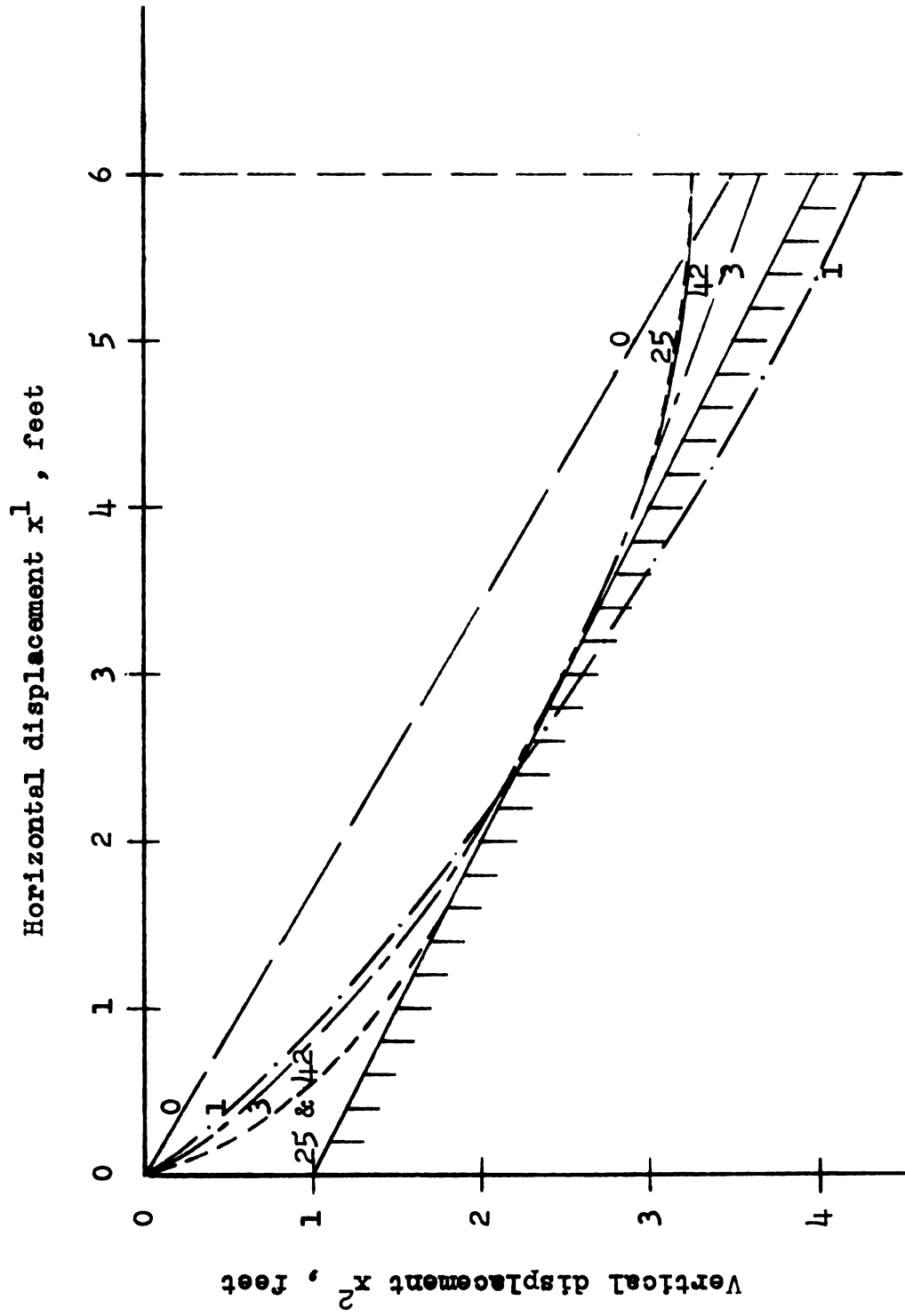


Figure 10 Paths of brachistochrone problem - Case IV

- (2) It seems not advisable to start with a large dC in the experimental stage of a given optimization problem. In general, a small dC may cause a slow convergence; however, too large a value of dC may create computational difficulty, such as poor predictability. Smaller dC will usually be associated with smoother t_f vs. N plots (compare Figure 6 and Figure 7).
- (3) It is inaccurate to say that the present iteration is inferior to the preceeding one by an argument simply based on the measurement $J_{\text{new}} - J_{\text{old}} > 0$. For instance, in Figure 6, $t_f(6) > t_f(5)$, however, the 6th iteration is not inferior to the 5th, since during the 5th iteration, the state constraint was violated and that was not the case for the 6th iteration. Furthermore, $t_f(17) > t_f(16)$, in Figure 6, the 17th iteration is again not inferior to the 16th although in both of these iterations the constraint was violated. This is because the degree of constraint violation incurred in the 17th iteration is less than that in the 16th (the accumulation of the weighted penalty for the 16th iteration was .02450, and it was .00021 in the 17th iteration).

5.2 Orbit Transfer of a Solar Sail Ship

A solar sail ship develops its propulsion energy from solar radiation. The radiation pressure is inversely proportional to the square of the distance between the sun and the sail. By controlling the sail angle, the ship can

be driven to any destination in the solar system.

Assume that the orbits of the planets in the solar system are circular and coplanar. Furthermore, only the gravitational force due to the sun is considered. the equations of motion of the ship are:

$$\dot{v}_r = \frac{(v_\phi)^2}{r} + a_o \left(\frac{r_o}{r}\right)^2 \cos^3 u - g_o \left(\frac{r_o}{r}\right)^2 \quad (5.2.1a)$$

$$\dot{v}_\phi = -2 \frac{v_r v_\phi}{r} + a_o \left(\frac{r_o}{r}\right)^2 \sin u \cos^2 u \quad (5.2.1b)$$

$$\dot{r} = v_r \quad (5.2.1c)$$

where u is the sail angle; v_r and v_ϕ are the radial and circumferential velocities of the ship, respectively; r_o , r , and r_p are the initial orbital, instantaneous, and final orbital radii respectively; g_o is the solar gravitational acceleration at the initial orbit; a_o is the acceleration of the ship due to solar radiation pressure exerted normally on the sail at r_o ; v_p is the orbital speed of the final orbit (see Figure 11).

We want to find the optimal sail angle and the associated trajectory of the ship such that the ship is driven from a given initial orbit to a specified final orbit in shortest time.

Tsu [TS-1] studied the problem and found an approximate analytic solution by setting $\dot{v}_r = 0$ with constant sail angle θ . The resulting trajectory of the ship was a

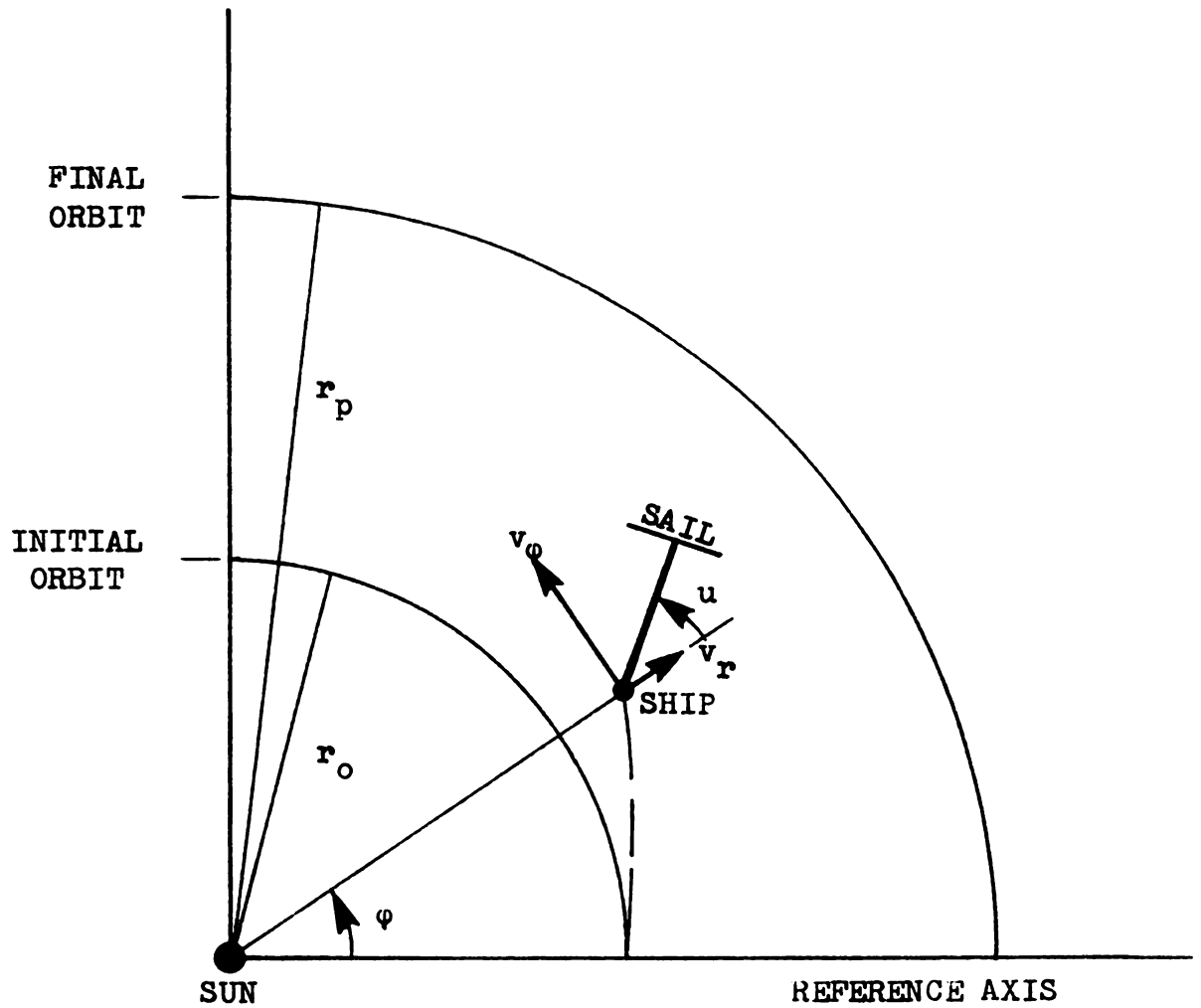


Figure 11 Orbit transfer of a solar sail ship

logarithmic spiral. London [LO-1] obtained an exact solution of the problem by assuming a logarithmic spiral trajectory and constant sail angle. Kelley [KE-1] obtained numerical solutions for variable sail angle for a transfer from Earth orbit to Mars orbit. We shall, in particular, consider the case of transferring from Earth orbit to Venus orbit with matching terminal orbital speed.

Let v_o be the initial orbital speed, and v_p be the terminal orbital speed. One readily finds that

$$v_p = v_o \sqrt{r_o/r_p} \quad (5.2.2)$$

The following data were used for the Earth to Venus transfer problem:

$$g_o = .592 \times 10^{-2} \text{ m/sec/sec (at Earth orbit)}$$

$$a_o = .2 \times 10^{-2} \text{ m/sec/sec (at Earth orbit)}$$

$$r_o = 149.6 \times 10^9 \text{ m}$$

$$r_p = 108.2 \times 10^9 \text{ m}$$

$$v_o = 29.76 \times 10^3 \text{ m/sec}$$

$$v_p = 35 \times 10^3 \text{ m/sec}$$

For computational purposes, the system is normalized by setting $x^1 = 10^{-3}v_r$, $x^2 = 10^{-3}v_\phi$ and $x^3 = 10^{-9}r$. Note that x^1 and x^2

are in km/sec, and x^3 is in GM (149.6 GM = 1 astronomical unit). With this notation, and the given data, (5.2.1) becomes

$$\begin{aligned}\dot{x}^1 &= 10^{-6} \frac{(x^2)^2}{x^3} + .04476(x^3)^{-2} \cos^3 u - .13249(x^3)^{-2} \\ &= f^1\end{aligned}\quad (5.2.3a)$$

$$\begin{aligned}\dot{x}^2 &= -2 \times 10^{-6} \frac{x^1 x^2}{x^3} + .04476(x^3)^{-2} \sin u \cos^2 u \\ &= f^2\end{aligned}\quad (5.2.3b)$$

$$\begin{aligned}\dot{x}^3 &= 10^{-6} x^1 \\ &= f^3\end{aligned}\quad (5.2.3c)$$

where $x = (x^1 \ x^2 \ x^3)^T$ is the state vector, and u is the control variable. The cost functional is defined as

$$J = w_1 t_f \quad (5.2.4)$$

where $w_1 = 1.1574 \times 10^{-5}$ is a normalization factor which converts seconds into days, that is, J represents the terminal time in days rather than seconds. The initial states are $x^1(0) = 0$, $x^2 = 29.76$ and $x^3(0) = 149.6$. The terminal constraints are

$$I^1 = x^2(t_f) - 35 = 0 \quad (5.2.5a)$$

$$I^2 = x^3(t_f) - 108.2 = 0 \quad (5.2.5b)$$

and the stopping function was chosen to be

$$S(t_f) = x^1(t_f) \quad (5.2.6)$$

$S = 0$ defines t_f if the ship is inside the Earth orbit.

The adjoint equations and their terminal conditions are

$$\dot{\lambda}_J^T = - \lambda_J^T \left(\frac{\partial f}{\partial x} \right)_* \quad \lambda_J^T(t_f) = [0 \ 0 \ 0] \quad (5.2.7)$$

$$\dot{\lambda}_I^T = - \lambda_I^T \left(\frac{\partial f}{\partial x} \right)_* \quad \lambda_I^T(t_f) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.2.8)$$

$$\dot{\lambda}_S^T = - \lambda_S^T \left(\frac{\partial f}{\partial x} \right)_* \quad \lambda_S^T(t_f) = [1 \ 0 \ 0] \quad (5.2.9)$$

where $f = (r^1 \ r^2 \ r^3)^T$, and $\left(\frac{\partial f}{\partial x} \right)_*$ is the $n \times n$ -matrix of partial derivatives which can be obtained by differentiating (5.2.3). The following were used in formulating the combined multipliers λ_{JS} and λ_{IS} :

$$(\dot{s})_{*,t_f} = (r^1)_{*,t_f} \quad (5.2.10)$$

$$(\dot{g})_{*,t_f} = w_1 \quad (5.2.11)$$

$$(\dot{\theta}^1)_{*,t_f} = (r^2)_{*,t_f} \quad (5.2.12a)$$

$$(\dot{\theta}^2)_{*,t_f} = (r^3)_{*,t_f} \quad (5.2.12b)$$

In the following computations, a dC of 200 was used, the integration step-size was 21600 seconds or .25 day. The initial guess for the control variable was $-\pi/4$.

Case I: (1) $U^{-1} = b - (b - a) \left(\frac{t - .5t_2}{.5t_2} \right)^2$, $t \in [0, 17280000]$

$a = .6666667$, $b = 1.5$; (2) In the subroutine SELCT, dC was reduced to 95%, and the error correction in I was

$$\begin{aligned} dI^1 &= -.05 I^1 && \text{if } |I^1| \leq .7 \\ &= -.2 I^1 && \text{otherwise} \\ dI^2 &= -.1 I^2 && \text{if } |I^2| \leq 1.082 \\ &= -.2 I^2 && \text{otherwise} \end{aligned}$$

each time SELCT was called.

Table 1 shows the results for some iterations:

Table 1 Numerical results, solar sail--Case I

<u>N</u>	<u>t_f(days)</u>	<u>dC</u>	<u>I^1</u>	<u>I^2</u>
0	190.257	--	.66432	-1.51442
1	183.708	200.00000	.87103	-1.38244
5	177.872	162.90125	.60193	- .30457
9	176.989	132.68409	.33148	.42920
20	178.796	75.47072	-.19432	.72614
35	180.004	34.96492	-.20613	.31793
50	180.228	16.19894	-.12551	.10774
70	180.094	5.80709	-.05074	.01969

From this table one observes that the terminal time t_f decreases to a minimum at $N = 9$, then increases to 180.228 at $N = 50$, and then decreases to 180.094 at $N = 70$. However, the terminal constraint errors are much smaller at $N = 70$ than those at $N = 9$.

Case II: (1) $U^{-1} = 1$; (2) In the main program, the value of dC was reduced to 90% if the prediction error exceeded the tolerance; (3) Same as that in I(2).

Table 2 shows some of the results obtained during the optimization process:

Table 2 Numerical results, solar sail--Case II

<u>N</u>	<u>t_f(days)</u>	<u>dC</u>	<u>I^1</u>	<u>I^2</u>
0	190.257	--	.66432	-1.51442
1	183.861	200.00000	.86967	-1.40991
5	178.270	162.90125	.59353	- .40070
8	177.695	139.66746	.47217	- .01108
20	178.574	75.47072	.12645	.22976
35	179.492	25.48943	.03152	.09307
50	179.746	3.33522	.01307	.02197
70	179.850	.14536	.00476	.00273

Comparing these results with those obtained in Case I, we conclude that Case II is more satisfactory than Case I, for, $N = 70$, $t_f = 179.850$ days in Case II, and $t_f = 180.094$ days in Case I, furthermore $I = (.00476 \quad .00273)^T$ in Case II, and $I = (-.05074 \quad .01969)^T$ in Case I. In both cases the average computer time was about 7.723 seconds/iteration.

Figure 12 shows the initial trajectory, the intermediate trajectory ($N = 8$), and the computed optimal trajectory ($N = 70$). Their corresponding sail angles (the control functions) were plotted in Figure 13 for Case II.

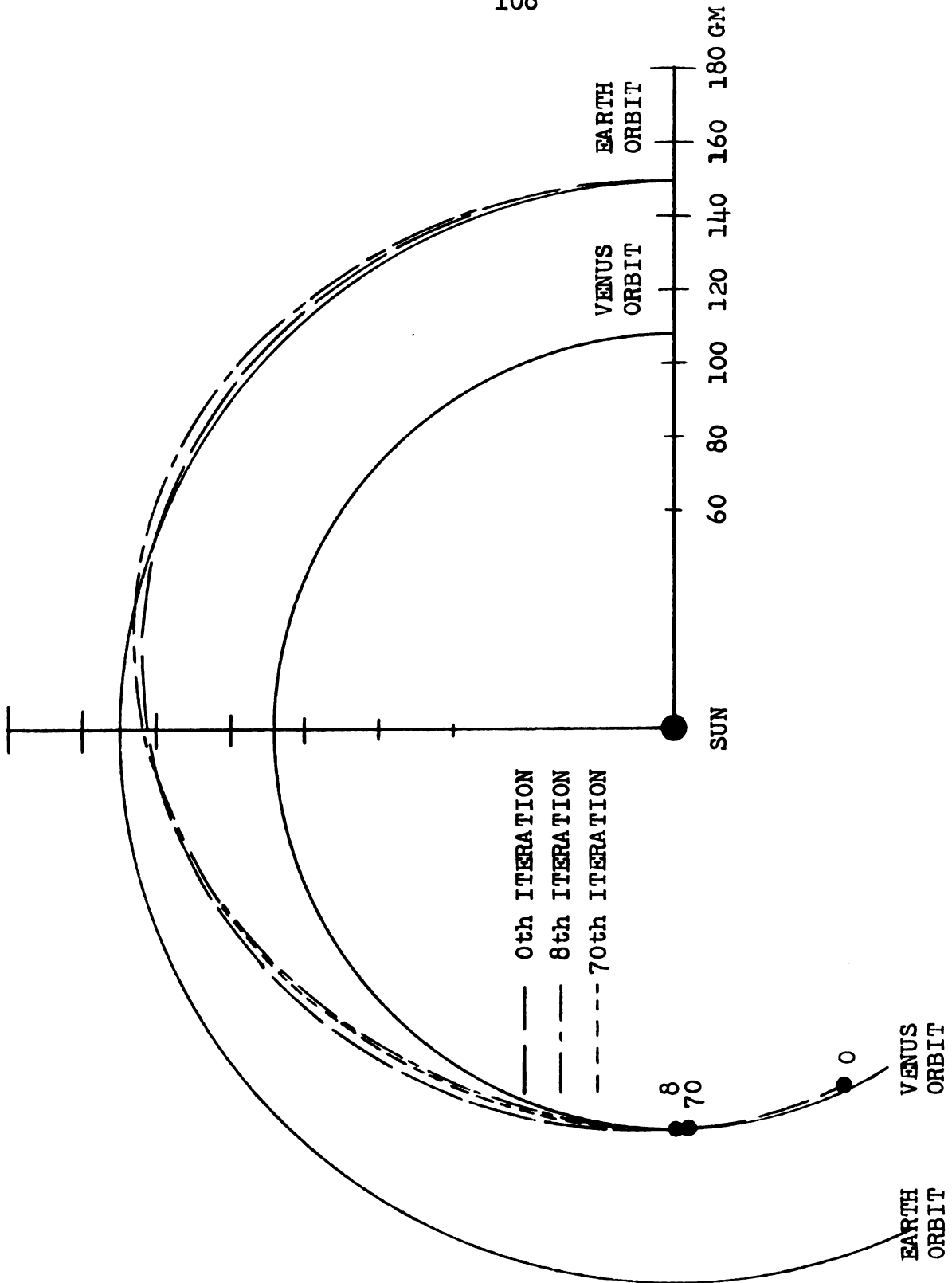


Figure 12 Trajectories of solar sail ship - Case II

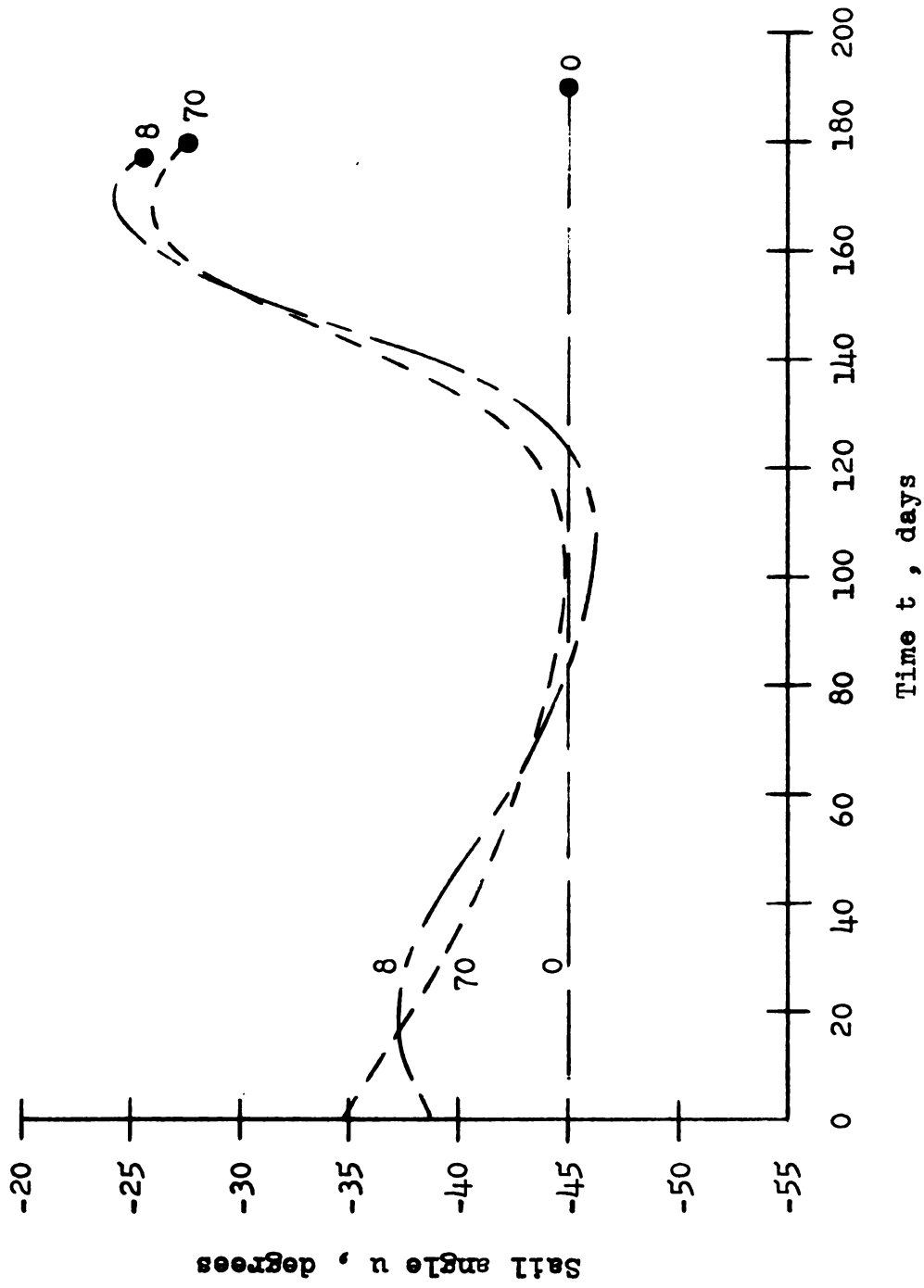


Figure 13 Sail angle of solar sail problem - Case II

5.3 Low Thrust Trajectory Optimization Problem

We shall consider the orbit transfer of a low thrust rocket. As in the solar sail problem, circular and coplanar orbits are assumed, and all but the sun's gravitational field are neglected. Only Earth to Venus orbit transfer will be studied numerically. Figure 14 shows the notation and the configuration of the problem.

The motion of the rocket can be described by the following set of differential equations:

$$\dot{v}_r = \frac{(v_\phi)^2}{r} + \frac{T}{m} \sin u - g_0 \left(\frac{r_0}{r}\right)^2 \quad v_r(0) = 0 \quad (5.3.1a)$$

$$\dot{v}_\phi = -2 \frac{v_r v_\phi}{r} + \frac{T}{m} \cos u \quad v_\phi(0) = v_0 \quad (5.3.1b)$$

$$\dot{r} = v_r \quad r(0) = r_0 \quad (5.3.1c)$$

where $r_0 = 149.6 \times 10^9$ m (average Earth orbital radius);

$v_0 = 29.76 \times 10^3$ m/sec (average Earth orbital speed);

$T = .5649336$ newton (constant thrust); g_0 is the same as that given in the last section. The mass of the rocket and fuel is

$$m = m_0 + \dot{m}t \quad (5.3.2)$$

where

$m_0 = 679.602$ kg (initial mass);

$\dot{m} = -1.012108 \times 10^{-5}$ kg/sec (fuel flow).

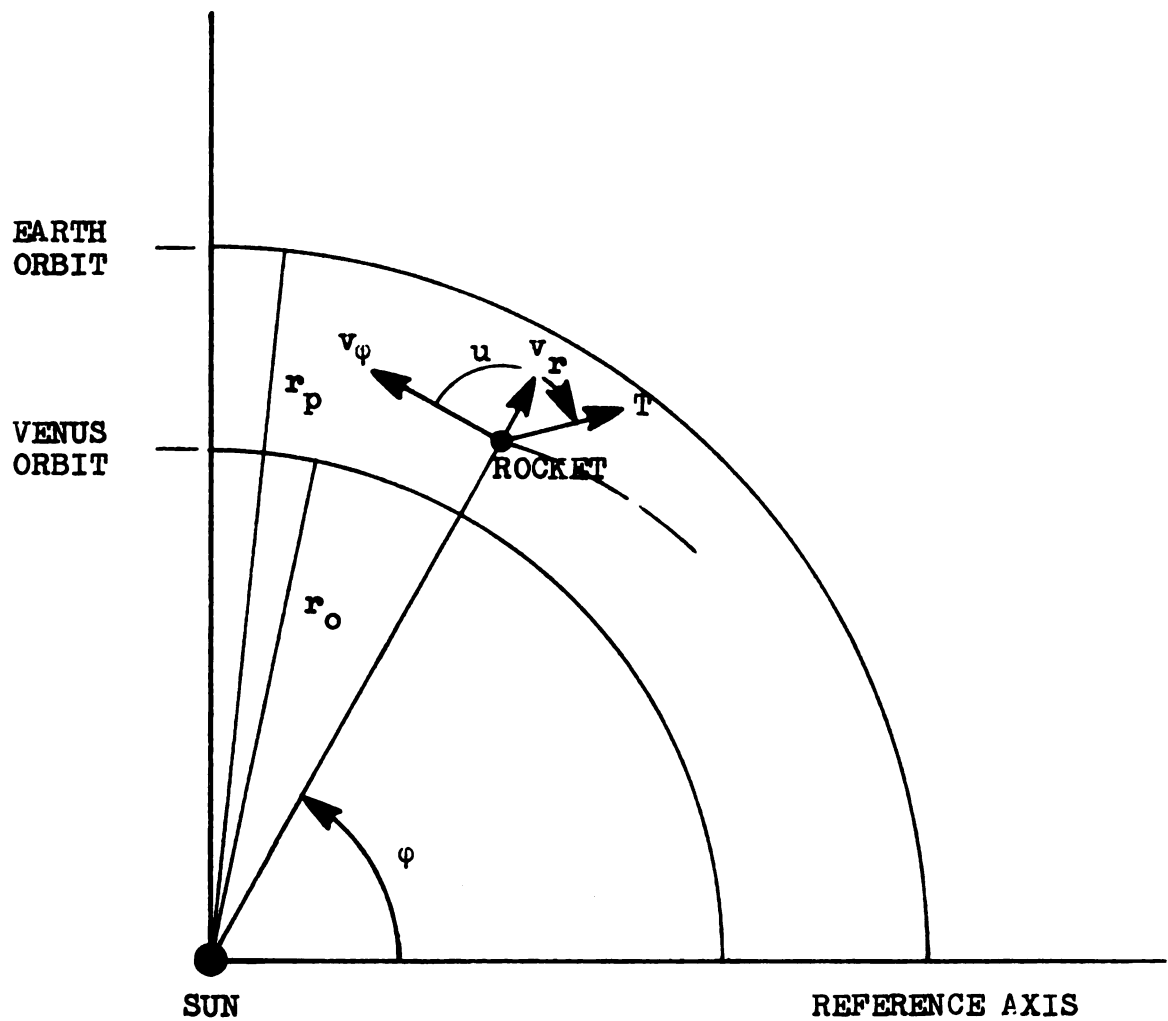


Figure 14 Orbit transfer of a low thrust rocket

Let v_p be the final orbital speed and r_p be the final orbital radius, then, for Earth-Venus transfer, $v_p = 35 \times 10^3$ m/sec and $r_p = 108.2 \times 10^9$ m.

The problem is to determine the optimal trajectory and the optimal thrust angle of the rocket such that the rocket is transferred from Earth orbit to Venus orbit with matching final orbital speed in shortest time. Similar problems have been considered by Lindorfer and Moyer [LI-1], Kelley [KE-2], Kelley et al. [KE-3], Moyer and Pinkham [MO-1], Kenneth and McGill [KEN-1], McReynolds [MCR-1], Sage [SA-1] and others.

If we use the same normalized variables as we did in the last section, and substitute the given data into (5.3.1), we obtain the normalized equations of motion:

$$\begin{aligned} \dot{x}^1 &= 10^{-6} \frac{(x^2)^2}{x^3} + \frac{8.31 \times 10^{-7}}{1-1.49 \times 10^{-8}t} \sin u - .13249(x^3)^{-2} \\ &= f^1 \end{aligned} \quad (5.3.3a)$$

$$\begin{aligned} \dot{x}^2 &= -2 \times 10^{-6} \frac{x^1 x^2}{x^3} + \frac{8.31 \times 10^{-7}}{1-1.49 \times 10^{-8}t} \cos u \\ &= f^2 \end{aligned} \quad (5.3.3b)$$

$$\begin{aligned} \dot{x}^3 &= 10^{-6} x^1 \\ &= f^3 \end{aligned} \quad (5.3.3c)$$

$$x(0) = (0 \quad 29.76 \quad 149.6)^T$$

where $x = (x^1 \ x^2 \ x^3)^T$ is the state variable (vector), and u is the control variable. The cost functional is

$$J = w_1 t_f \quad (5.3.4)$$

where $w_1 = 1.1574 \times 10^{-5}$. The terminal constraints are

$$I^1 = x^2(t_f) - 35 = \theta^1 = 0 \quad (5.3.5a)$$

$$I^2 = x^3(t_f) - 108.2 = \theta^2 = 0 \quad (5.3.5b)$$

the stopping functional is

$$S(t_f) = x^1(t_f) = \theta^0 \quad (5.3.6)$$

where $S = 0$ defines the terminal time t_f if the rocket is inside the Earth orbit.

The adjoint equations and their respective terminal conditions are:

$$\dot{\lambda}_J^T = - \lambda_J^T \left(\frac{\partial f}{\partial x} \right)_* \quad \lambda_J^T(t_f) = [0 \ 0 \ 0] \quad (5.3.7)$$

$$\dot{\lambda}_I^T = - \lambda_I^T \left(\frac{\partial f}{\partial x} \right)_* \quad \lambda_I^T(t_f) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.3.8)$$

$$\dot{\lambda}_S^T = - \lambda_S^T \left(\frac{\partial f}{\partial x} \right)_* \quad \lambda_S^T(t_f) = [1 \ 0 \ 0] \quad (5.3.9)$$

where $f = (f^1 \ f^2 \ f^3)^T$ and $\left(\frac{\partial f}{\partial x} \right)_*$ is the partial derivative matrix evaluated along the trajectory of the rocket. The quantities $(\dot{S})_{*,t_f}$, $(\dot{g})_{*,t_f}$, $(\dot{\theta}^1)_{*,t_f}$, and $(\dot{\theta}^2)_{*,t_f}$,

are the same as that given in the last section.

In the previous two examples, we used different values of dC , and different updating schemes to obtain different speeds of convergence. In this example we studied two cases with everything identical except different initial guesses of the control function u . For both cases, dC was reduced to 70% if the prediction error exceeded 2, and the present iteration would then be discarded, the program would regenerate the trajectory of the immediate preceding iteration, and then continued for further improvement. In the subroutine SELECT, dC was reduced to 97.5% for each $1 \leq N_{\text{eff}} \leq 15$, and no effect otherwise. Where N_{eff} is the effective iteration number which differs from N by 2 times the number of iterations discarded preceding the N_{eff} th effective iteration. The same scheme for choosing dI was used as that in I(3) of Section 5.2.

Case I: $u(t) = 2\pi/3$, $t \in [0, 17280000]$.

Table 3 shows the numerical results for this case. One observes, from Table 3, that the terminal time t_f increased from 211.732 days to a maximum of 248.993 days (at $N = N_{\text{eff}} = 18$), however the constraint I^2 was driven from 9.65560 to a much more satisfactory value, 1.12011. The terminal time then decreased to 202.370 days at $N = 57$ or $N_{\text{eff}} = 53$ with $I^1 = -.05333$ and $I^2 = .03045$.

Table 3 Numerical results, low thrust--Case I

<u>N</u>	<u>N_{eff}</u>	<u>t_f(days)</u>	<u>dC</u>	<u>I¹</u>	<u>I²</u>
0	0	211.732	--	-.07349	9.65560
1	1	212.806	100.00000	-.06891	9.11231
3	3	215.677	95.00000	-.05462	8.05579
5	5	219.369	90.25000	-.03620	7.03335
7	7	223.707	85.73750	-.01318	6.03741
9	9	228.542	81.45062	-.01445	5.06257
18	18	248.993	69.83373	.11914	1.12011
25	25	237.318	69.83373	.09429	.48234
30	30	228.940	69.83373	.09199	.28691
40	40	214.147	69.83373	.08697	.10241
50	50	203.369	69.83373	.03474	.04004
57	53	202.370	34.21853	-.05333	.03045

Several trajectories are plotted in Figure 15, the corresponding thrust angles are plotted in Figure 16. Here $N_{\text{eff}} = 0$ corresponds to the initial iteration, and $N_{\text{eff}} = 53$ corresponds to the computed optimal trajectory.

Case II: $u(t) = 5\pi/6$, $t \in [0, 17280000]$.

The numerical results for some iterations are given in Table 4.

Figure 17 and Figure 18 show some of the trajectories and the corresponding thrust angle of the rocket respectively.

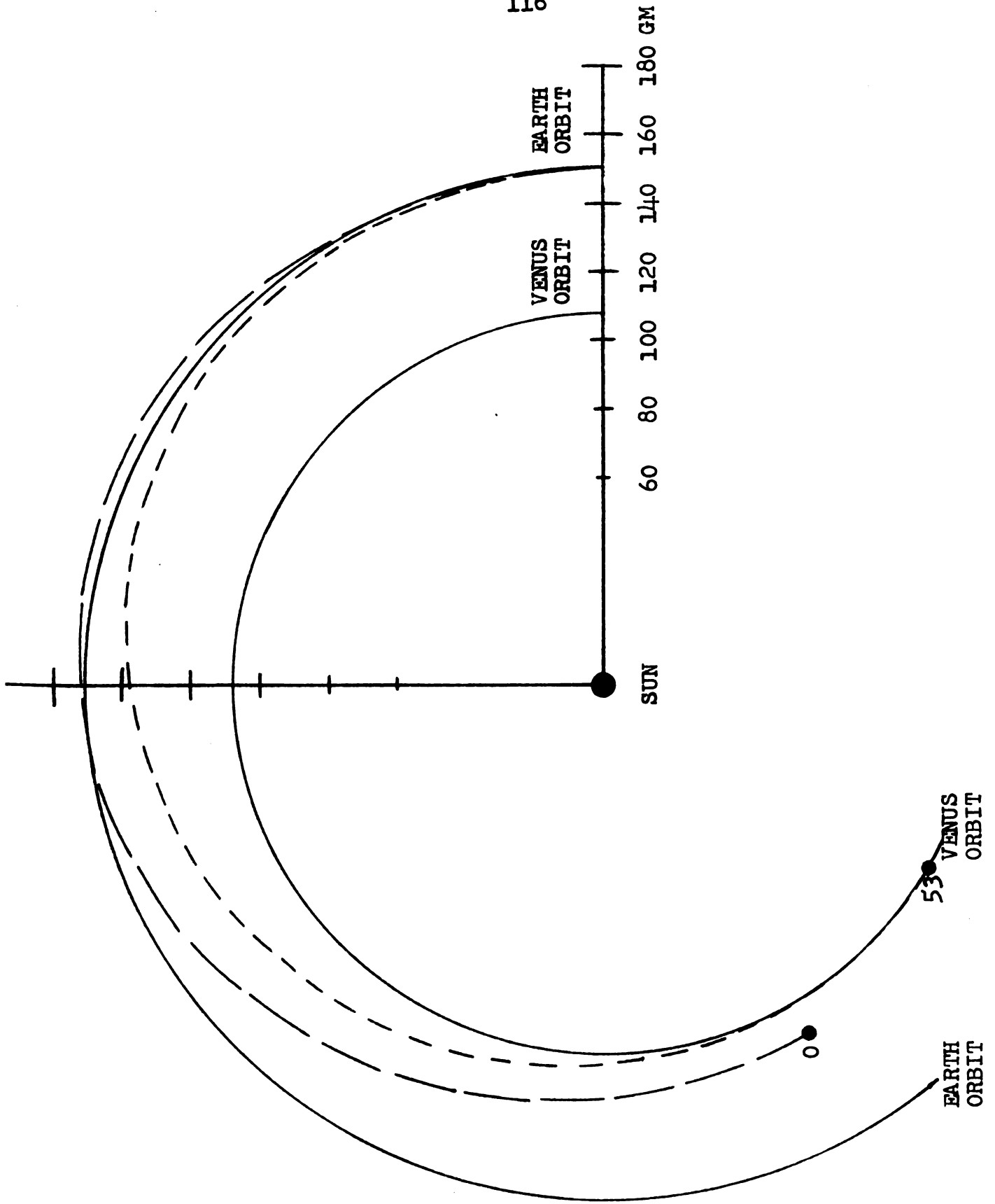


Figure 15 Trajectories of low thrust rocket - Case I

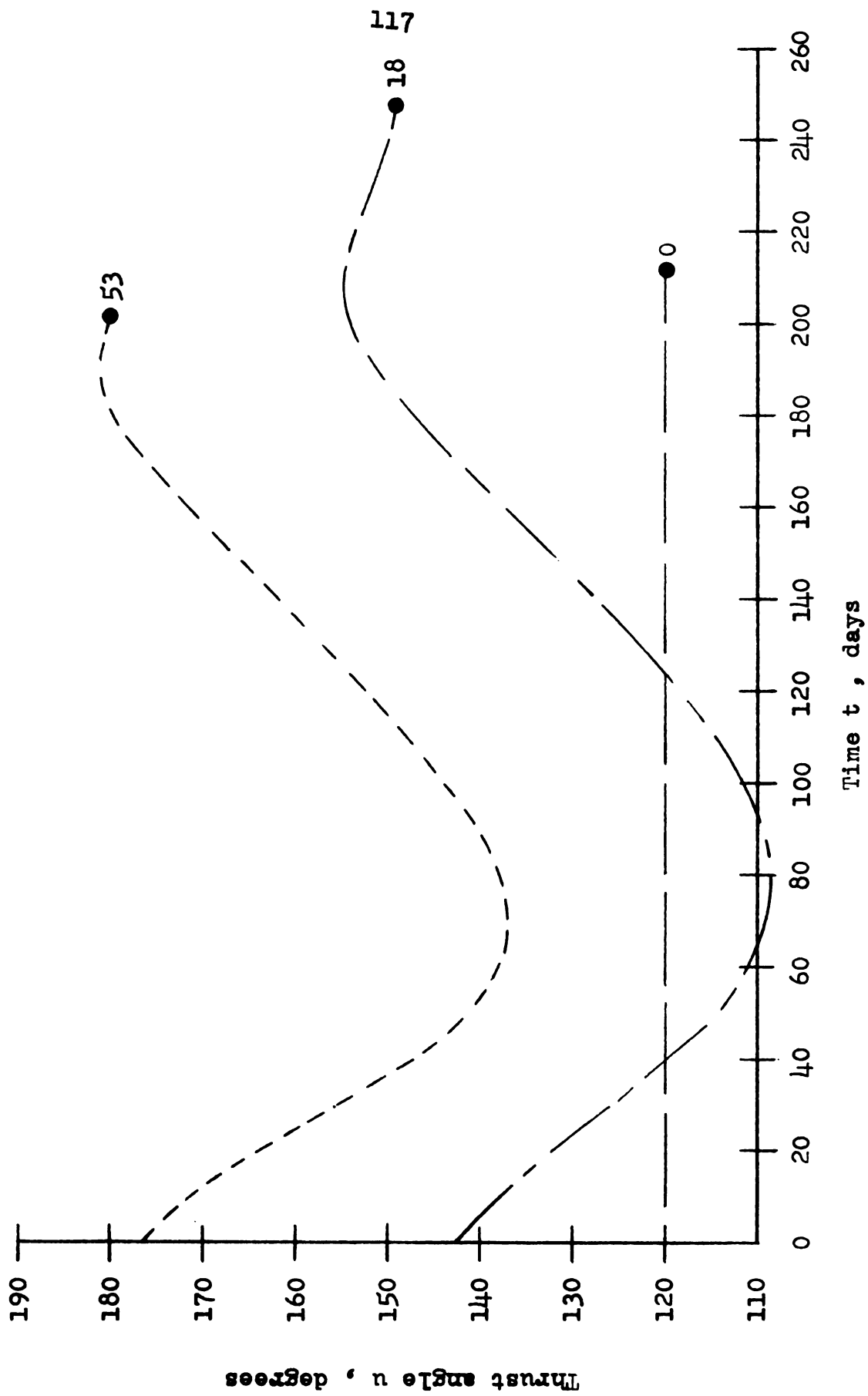


Figure 16 Thrust angle of low thrust rocket - Case I

Table 4 Numerical results, low thrust--Case II

<u>N</u>	<u>N_{eff}</u>	<u>t_f(days)</u>	<u>dC</u>	<u>I¹</u>	<u>I²</u>
0	0	168.783	--	2.60378	.79475
1	1	170.559	100.00000	2.46572	.75794
3	3	174.602	95.00000	2.18244	.67727
5	5	179.364	90.25000	1.87816	.58379
7	7	185.083	85.73750	1.52584	.47010
9	9	192.492	81.45062	1.05455	.31809
11	11	196.054	77.37809	.64266	.23335
15	15	194.235	69.83373	.52046	.15884
20	20	192.706	69.83373	.39178	.09931
27	27	191.768	69.83373	.26846	.05286
30	28	191.785	48.88361	.25923	.04783
42	40	192.754	48.88361	.16438	.02926
52	50	193.457	48.88361	.08437	.03647

Comparing Table 3 and Table 4, we conclude that to use $u = 5\pi/6$ as an initial guess for the thrust angle is more favorable than that of $2\pi/3$. It is true in both cases that longer travelling time (t_f) pays the price for better satisfaction of the terminal constraints.

In the computation of this example, the integration step-size was 21600 seconds (or .25 day), and the average computer time used was 8.788 seconds/iteration in Case I, and 7.786 seconds/iteration in Case II.

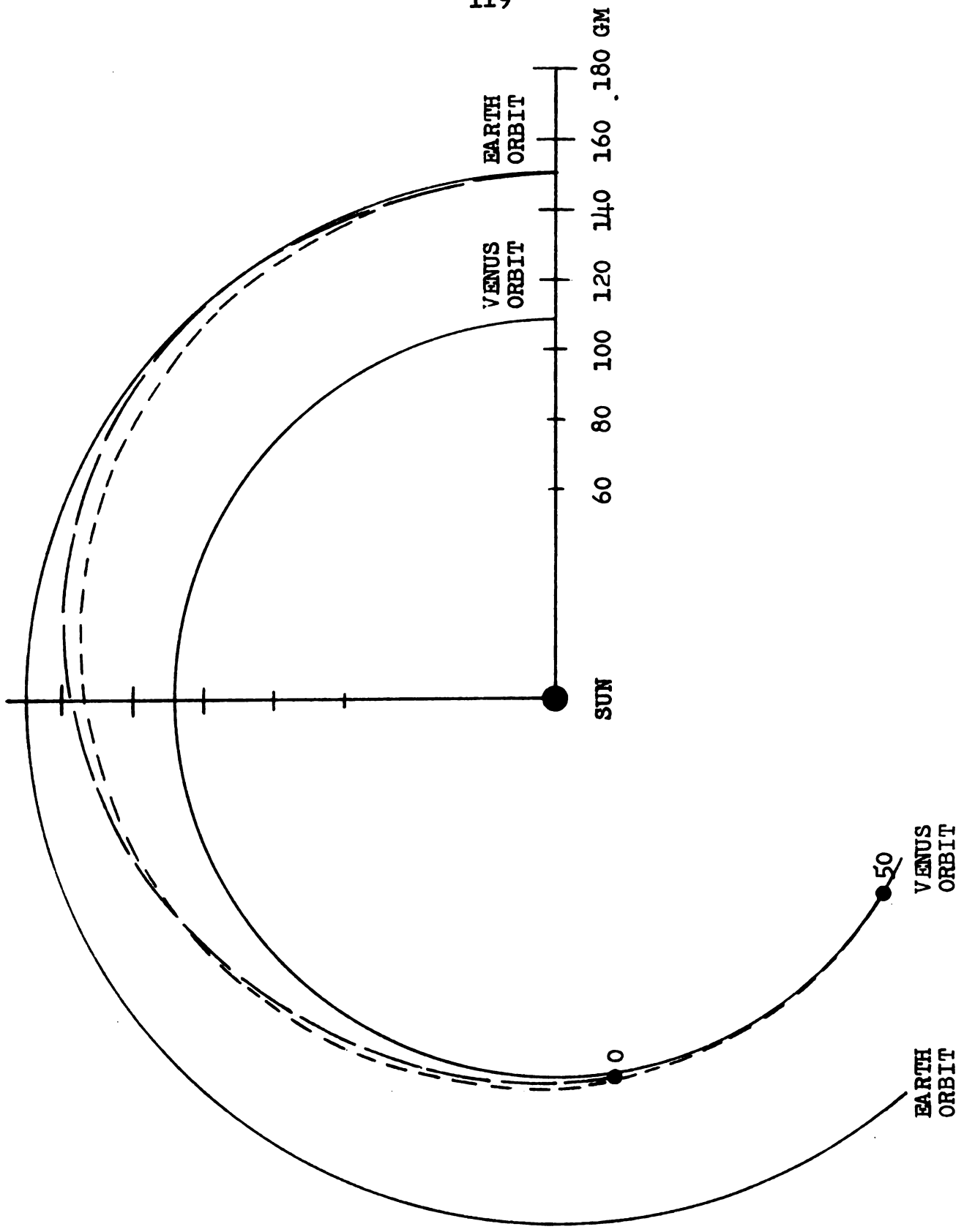


Figure 17 Trajectories of low thrust rocket - Case II

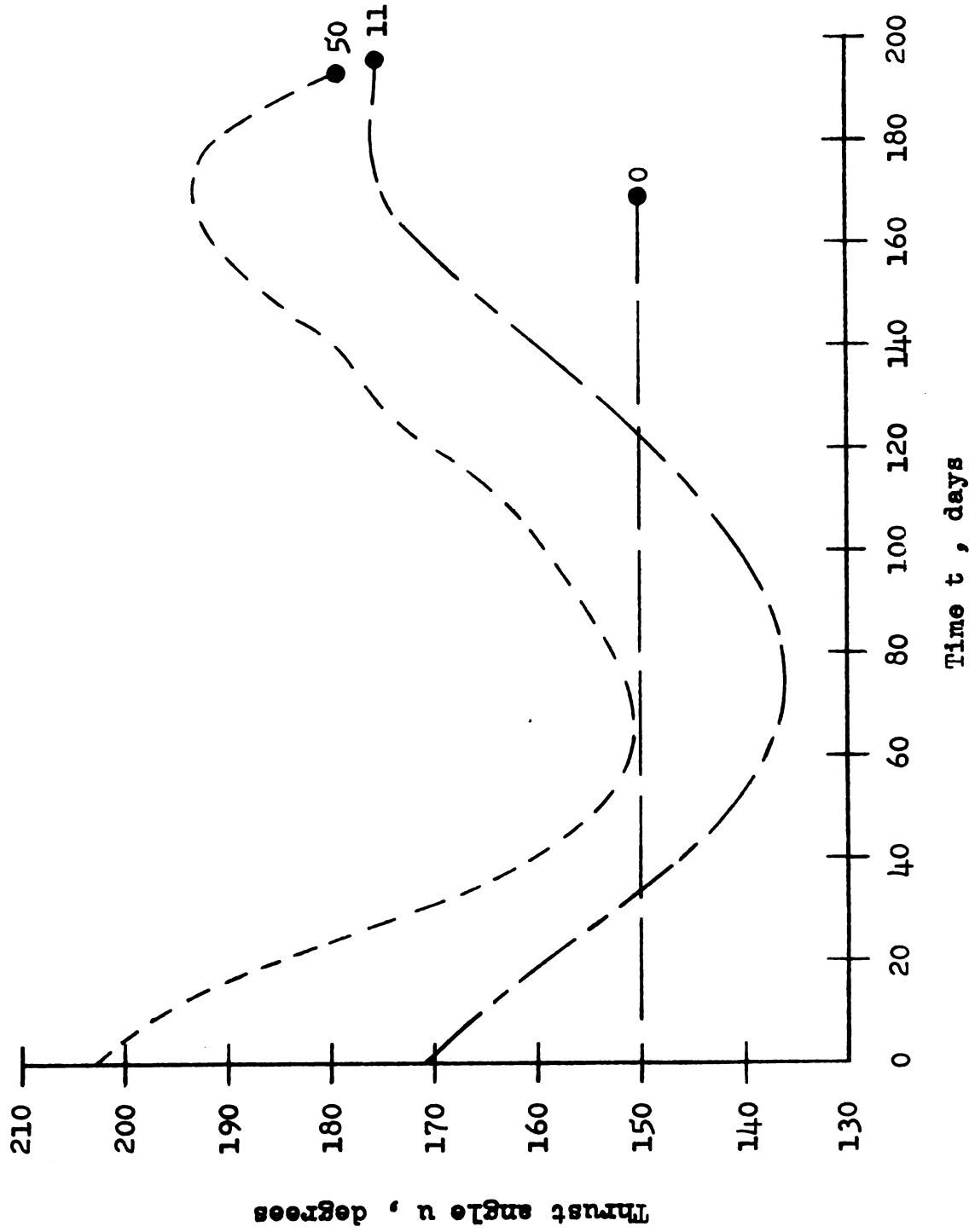


Figure 18 Thrust angle of low thrust rocket - Case II

VI SUMMARY AND CONCLUSION

This thesis considers the gradient computational technique for a class of optimal control problems, the control problem of Bolza, with various constraints. The major objective is to derive computational algorithms and their respective iterative procedures.

Some basic theorems and necessary and sufficient conditions of the variational calculus, from the simplest problem of the calculus of variations to a sophisticated control problem of Bolza, are presented in Chapter II. This introductory treatment of the variational theory serves as background material for the later development of the central objective of this thesis.

The control problem of Bolza is redefined in Chapter III as a computational version of the problem. Algorithm I is then derived in some detail. Iterative formulas are given in terms of the step-size dC in the control effort space, the error correction dI of the isoperimetric constraints I , the adjoint variables, and the stored nominal solution arc of the system obtained in the preceding iteration. As discussed in Chapter III, the present formulation of the optimal control problem is more general and is superior to those formulated earlier by other authors if computing time and memory requirement are considered.

An important subclass of the general control problem of Bolza is defined in Chapter IV. Bang-bang control problems, and problems with discrete control variables as well as continuous control variables belong to this category. Since it is so important in application, a special algorithm—Algorithm II is derived for this special class of problems, although the general algorithm—Algorithm I is also applicable. Iterative procedures for both algorithms are given in detail.

In Chapter V, three numerical examples are given to illustrate the application of the general computational algorithm. Alternative choices of the iterative parameters and their updating schemes are presented in detail. Solutions of these examples are plotted, tabulated and discussed. A sample program written in FORTRAN is given in APPENDIX A. The major part of the main program is machine independent, will not need to change for individual problems, however, the user has to write his own subroutines to fit the specific problem.

As a common character of all gradient techniques, the convergence of a problem is relatively insensitive to the initial guess of the problem. The convergence is fast in the starting iterations, and the speed of convergence slows down when the number of iterations increases.

Suitable choices of the iteration parameters (such as the iteration step-size, the weighting matrices, the error correction dI) and their updating schemes are the key factors

for obtaining a reasonable solution of a specific problem with a reasonable number of iterations. There seem to be no general rules for choosing these parameters besides reasonable guessing and conducting meaningful experiments. Relatively larger integration step-size can be used during the experimental stage of the problem so that computer time can be saved for obtaining final solution. One should not over emphasize the importance of the measurements of linearization errors. The suitability of linear approximation directly affects the degree of predictabilities in dJ and dI ; however, it has no effects on the accuracy of the solution arc of the system, and the latter can only be affected by the integration step-size. It is important to note that each time an iterative solution is rejected, it is necessary to regenerate the nominal solution obtained in the preceding iteration, hence one has to pay twice as much computer time needed for one iteration before any attempt can be made for further iteration. Therefore, one should not reject any solution unless it is important to do so.

Throughout this work, a digital computer to carry out the iterative process is assumed, however, the use of a well equipped hybrid computer is obviously possible. The problem of how convenient or inconvenient it will be and the problem of how much computation time can be saved against the use of a digital machine are left open.

The following extensions or further research work related to this study seem promising:

- (1) Formulate a multiple-stage control problem of Bolza and derive computational algorithms for the solution of the problem.
- (2) Search for new techniques for solving problems with inequality constraints.
- (3) Add further realism to the control problem of Bolza by incorporating stochastic effects in the system model.

REFERENCES

- [BE-1] Berkovitz, L. D., "Variational Methods in Problems of Control and Programming," J. Math. Anal. Appl., vol. 3, 1961.
- [BE-2] Berkovitz, L. D., "On Control Problems with Bounded State Variables," J. Math. Anal. Appl., vol. 5, 1962.
- [BL-1] Bliss, G. A., "The Evolution of Problems of the Calculus of Variations," American Mathematical Monthly, vol. 43, pp. 598-609, 1963.
- [BL-2] Bliss, G. A., "Lectures on the Calculus of Variations," The University of Chicago Press, 1946.
- [BL-3] Bliss, G. A., "The Problem of Bolza in the Calculus of Variations," Annals of Mathematics, XXXIII, pp. 261-274, 1932.
- [BLU-1] BLUM, E. K., "The Calculus of Variations, Functional Analysis, and Optimal Control Problems," in Topics in Optimization, ed. by G. Leitmann, 1967.
- [BR-1] Bryson, A. E. and Denham, W. F., "A Steepest-Ascent Method for Solving Optimum Programming Problems," J. of Applied Mechanics, pp. 247-251, June 1962.
- [BR-2] Bryson, A. E. Jr. and 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, pp. 2544-2550, Nov. 1963.
- [BRE-1] Breakwell, J. V. and Sperryer, J. L. and Bryson, A. E., "Optimization and Control of Nonlinear Systems Using the Second Variation," SIAM J. Control, Ser. A, vol. 1, No. 2, pp. 193-223, 1963.
- [CO-1] Coddington, E. A., Levinson, N., "Theory of Differential Equations," McGraw-Hill, 1955.
- [COU-1] Courant, R., "Calculus of Variations and Supplementary Notes and Exercises," 1945-1946, revised and amended by J. Moser, New York University Institute of Mathematical Sciences, New York, 1956-1957.

- [DE-1] 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, pp. 25-34, January 1964.
- [DE-2] Denham, W. F., "Steepest-Ascent Solution of Optimal Programming Problems," Raytheon Rept., BR-2393 (April 1963)
- [DR-1] Dreyfus, S. E., "Variational Problems with State Variable Inequality Constraints," Rand Corp. Paper P-2605 (July 1962)
- [ED-1] Edwards, R. N. and Brown, H., "Ion Rockets for Small Satellites," Preprint of ARS Controllable Satellites Conference, MIT, Cambridge, Mass., 1959.
- [EV-1] Eveleigh, Virgil, "Adaptive Control and Optimization Techniques," McGraw-Hill, 1967.
- [GA-1] Gamkrelidze, R. V., "Optimal Control Processes with Restricted Phase Coordinates," Izvest. Akad. Nauk S.S.S.R., Ser. Mat., vol. 24, 1960.
- [GE-1] Gelfand, I. M. and Fomin, S. V., "Calculus of Variations," Prentice-Hall, 1963.
- [GU-1] Guim, T., "On First Order Necessary Conditions for Variational and Optimal Control Problems," Ph.D. Dissertation, UCLA 1964.
- [HE-1] Hestenes, M. R., "A General Problem in the Calculus of Variations with Applications to Paths of Least Time," Rand Corporation RM-100, 1949; also Astia Document No, AD112382, 1950.
- [HE-2] Hestenes, M. R., "Variational Theory and Optimal Control Theory," in Computing Methods in Optimization Problems, edited by Balakrishnan, A. V. and Neustadt, L. W., 1964.
- [HE-3] Hestenes, M. R., "On Variational Theory and Optimal Control Theory," J. SIAM Control, Ser. A, vol. 3, No. 1, 1965.
- [HE-4] Hestenes, M. R., "Calculus of Variations and Optimal Control Theory," John Wiley & Sons, Inc. 1966.

- [HE-5] Hestenes, M. R., "Numerical Method of Obtaining Solutions of Fixed End Point Problems in the Calculus of Variations," Rand Corp. Rep. RM-102, 1949.
- [KA-1] Kalaba, R., "On Nonlinear Differential Equations, The maximum Operation, and Monotone Convergence," J. of Math. and Mech., vol. 8, pp. 519-574, 1959.
- [KE-1] Kelley, H. J., "Gradient Theory of Optimal Flight Paths," ARS Journal 30, pp. 947-954, Oct. 1960.
- [KE-2] Kelley, H. J., "Method of Gradients," G. Leitmann, ed., Optimization Techniques, Chapter 6, Academic Press, New York, 1962.
- [KE-3] Kelley, H. J., Kopp, R. E., and Moyer, H. G., "A Trajectory Optimization Technique Based Upon the Theory of the Second Variation," Presented at the AIAA Astrodynamics Conference, Yale University Connecticut, August 1963.
- [KEN-1] Kenneth, P. and McGill, R., "Two-Point Boundary-Value-Problem Techniques," Advances in Control Systems, vol. 3, edited by Leondes, Academic Press, New York, 1966.
- [LI-1] Lindorfer, W. and Moyer, H. G., "An Application of a Low-Thrust Trajectory Optimization Scheme to Planar Earth-Mars Transfer," ARS Journal, Feb. 1962.
- [LO-1] London, Howard S., "Some Exact Solutions of the Equations of Motion of A Solar Sail with Constant Sail Setting," ARS Journal, Feb. 1960.
- [LU-1] Lurie, A. I., "Thrust Programming in a Central Gravitational Field," G. Leitmann, ed., Topics in Optimization, Chapter 4, Academic Press, New York, 1967.
- [MC-1] McGill, Robert, "Optimal Control, Inequality State Constraints, and the Generalized Newton-Raphson Algorithm," J. SIAM, Ser. A. Control, vol. 3, 1965.
- [MC-2] McGill, R. and Kenneth, P., "Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator," AIAA J., vol. 2, pp. 1761-1766, 1964.
- [MCR-1] McReynolds, Stephen R., "The Successive Sweep Method and Dynamic Programming," J. MAA, vol. 19, No. 3, Sept. 1967.

- [MCS-1] McShane, E. J., "On Multipliers for Lagrange Problems," Amer. J. Math 61, 1939.
- [MO-1] Moyer, H. Gardner and Pinkham, Gordon, "Several Trajectory Optimization Techniques, Part II: Application," Computing Methods in Optimization Problems, edited by Balakrishnan and Neustadt, Academic Press, Inc., New York, 1964.
- [OK-1] Okamura, Kiyohisa, "Some Mathematical Theory of the Penalty Method for Solving Optimum Control Problems," J. SIAM Control, Ser. A, vol. 2, No. 3, 1965.
- [PO-1] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., "The Mathematical Theory of Optimal Processes," Interscience, New York, 1962.
- [SA-1] Sage, Andrew P., "Optimum Systems Control," Chapter 13, Prentice-Hall, Inc., New Jersey, 1968.
- [TS-1] Tsu, T. C., "Interplanetary Travel by Solar Sail," ARS Journal, June 1959.
- [VA-1] Valentine F. A., "The Problem of Lagrange with Differential Inequalities as Added Side Conditions," Contributions to the Calculus of Variations 1933-37, University of Chicago Press.
- [VAC-1] Vachino, R. F., "A Generalized Steepest Descent Algorithm for Multistage Optimization Processes," Ph.D. thesis, 1968, University of Michigan.
- [VAC-2] Vachino, R. F., "Steepest Descent with Inequality Constraints on the Control Variable," J. SIAM Control, vol. 4, No. 1, 1966.

APPENDIX A

A SAMPLE PROGRAM

```

FTN.
REDUCE.
LGO.
.
PROGRAM GCT (INPUT,OUTPUT,TAPE2=INPUT,TAPE3=OUTPUT)
C
C GRADIENT COMPUTATIONAL TECHNIQUE FOR THE CONTROL PROBLEM OF BOLZA
C FROM PH D THESIS. S J WANG. DEPT OF EE, MSU.
C
C
C SSA2      SSA2      SSA2      SSA2
C
EXTERNAL F, FO, ADJN, ADJNO
DIMENSION LSCRACH(2), MSCRACH(2)
DIMENSION B(19), BP(19), AUX(10,19), EB(19), EF(4), PRMT(5), WT1(2
1 ), TEMPV(2), DI(2), PII(2,2), PIJ(2), XO(4)
COMMON X(3), UNEW(1), UOLD(1), U(1,1), GISU(2,1), GJSU(1),
1 XINEW(2), XIOLD(2), XJNEW, XJOLD, GD, TD(2), TF, W1, W2, TO, DT,
2 NTF, N, NLIM, NITER, NEFF, NOLST, BIOLD
EQUIVALENCE (PII(1),B(13)), (PIJ(1),B(17)), (PJJ,B(19))
C
C UNUSED ARRAYS
C DIMENSION ANEW(1), ALPHA(1), AOLD(1), BNEW(1), BETA(1), BOLD(1),
1 W(1,1), A(1,1), TEMP(1,1), VI(1,1), VJ(1)
EQUIVALENCE (Z, ANEW(1), ALPHA(1), AOLD(1), BNEW(1), BETA(1), BOLD
1 (1), W(1), A(1), TEMP(1), VI(1), VJ(1))
C
C *****
C NSTAT=DIM OF STATE VARIABLE, NCONT=DIM OF CONTROL VARIABLE
C NFOR=NO OF FORWARD INTEGRATIONS, NCONS=DIM OF ISOPERIMETRIC CONSTRAINTS
C NPARA=DIM OF CONTROL PARAMETER, NPP=DIM OF INEQUALITY PART OF ISOP CONST
C NITER=NO OF ITERATIONS, NEFF=NO OF EFFECTIVE ITERATIONS
C NOLST=INDEX FOR 'NO PREVIOUS RECORDS', NLIM=MAX ALLOWED NO OF RECORDS
C NBACK=NO OF BACKWARD INTEGRATIONS, NTF=NO OF RECORDS IN THE FILE
C IN THIS EXAMPLE EACH RECORD CONTAINS 9 ITEMS, SEE SUBRS READ OR WRITE
C *****
C
C READ (2,1) NSTAT, NCONT, NFOR, NCONS, NPARA, NPP, NITER, NEFF,
1 NOLST
C WRITE (3,1) NSTAT,NCONT,NFOR,NCONS,NPARA,NPP,NITER,NEFF,NOLST
1 FORMAT (16I5)
C NBACK = NSTAT*(NCONS+2) + NCONS*(NCONS+NPARA+1) + NPARA + 1
C NLIM = 2000
C READ (2,2) (XO(1), I = 1, NFOR)
C WRITE (3,5) (XO(1), I = 1, NFOR)
C READ (2,2) (EF(1), I = 1, NFOR)

```

WW
WW
WW
WW
WW
WW
WW

```

WRITE (3,5) (EF(I), I = 1, NFOR)
READ (2,2) (EB(I), I = 1, NBACK)
WRITE (3,5) (EB(I), I = 1, NBACK)
READ (2,2) (WT(I), I = 1, NCONS)
WRITE (3,5) (WT(I), I = 1, NCONS)
READ (2,2) DC2, STOPC, TO, DT, EI, EJ, W1, W2
WRITE (3,5) DC2, STOPC, TO, DT, EI, EJ, W1, W2
2  FORMAT (8F10.5)
5  FORMAT (8F15.6)
31 READ (2,30) T, T2, U1, U2, UNEW, I
WRITE (3,35) T, T2, U1, U2, UNEW, I
30  FORMAT (5F10.5, I1)
35  FORMAT (5F15.6, I1)
N = (T - TO)/DT + 1.5
34 U(1,1) = 1./(U1*T + U2)
CALL WRITE
T = T + DT
N = N + 1
IF (T - T2) 34, 34, 32
32 IF (I) 31, 31, 33
33 NTF = N - 1
DC2 = DC2**2
STOPC = STOPC**2

```

C *****
C SUBROUTINE HPCG AND THIS PROGRAM, BETWEEN STATEMENTS 200 AND
C 999, ARE MACHINE INDEPENDANT. THE USER MUST CHANGE THE REST OF
C THIS PROGRAM AND SUBROUTINES ADJN, ADJN1, ADJNO, F, READ, SELECT,
C WRITE, AND WTMT FOR DIFFERENT COMPUTERS AND PROBLEMS. IF THE
C TRAJECTORY DATA IS STORED ON TAPE RATHER THAN ON DISK, THE CALLS
C TO READ AND WRITE IN BLOCK 900 MAY ALSO NEED TO BE CHANGED.

C *****
C INTEGRATE THE STATE EQUATIONS FORWARD. ALSO EVALUATE THE INTEGRALS FOR
C THE PROBLEM OF BOLZA.

WW
WW

```

200 NEFF = NEFF - 1
210 PRMT(1) = TO
PRMT(2) = TO + 1.E8*DT
PRMT(3) = DT
DO 201 I = 1, NFOR
B(I) = X0(I)
201 BP(I) = EF(I)

```

C IF (NCONS) 208, 208, 207

```

207 DO 202 I = 1, NCONS
202 XIOLD(I) = XJNEW(I)
208 XJOLD = XJNEW

C
C SUBROUTINE F (T, B, BP) CALCULATES BP AS A FUNCTION OF B AND T.
C SUBR FO (T, B, BP, IHLF, N, PRMT, DJPRD, DC2W) IS CALLED ONCE FOR EACH
C TIME INTERVAL. FO STORES X ON THE DISK, PRINTS ANY DESIRED
C TRAJECTORY INFORMATION, AND CALCULATES S, THE STOPPING CONDITION.
C WHEN S = 0, SET PRMT(5) NONZERO, AND CALCULATE XJNEW, XJNEW, GD,
C AND TD.
C CALL HPCG (PRMT, B, BP, NFOR, IHLF, F, FO, AUX, DJ, DC2)
C
C NITER = NITER + 1
C IF (NOLST) 400, 400, 303
C
C TEST FOR AGREEMENT OF DJ WITH XJNEW - XJOLD
C
303 IF (ABS((DJ - (XJNEW - XJOLD))/DJ) - EJ) 304, 1100, 1100
C
C TEST FOR AGREEMENT OF DI WITH XJNEW - XIOLD
C
304 IF (NCONS) 400, 400, 306
306 Z = 0.
C DJINT = 0.
C DO 305 I = 1, NCONS
C TEMPV(1) = WT(I)*ABS(DI(I))
C IF (DJINT - TEMPV(1)) 307, 308, 308
C DJINT = TEMPV(1)
307 DJINT = TEMPV(1)
308 TEMPV(1) = WT(I)*ABS( DI(I) - (XJNEW(I) - XIOLD(I)) )
C IF (Z - TEMPV(1)) 309, 305, 305
309 Z = TEMPV(1)
305 CONTINUE
C
C Z IS THE (WT1) WEIGHTED UNIFORM NORM OF DI - (XJNEW - XIOLD).
C IF (DJINT) 310, 400, 310
310 IF (Z/DJINT - EI) 400, 1200, 1200
400 NEFF = NEFF + 1
401 NOLST = 1
C
C UPDATE THE WEIGHTING MATRICES A, U, AND W. RETURN THE INVERSES IN
C A, U, AND W. (U IS STORED ON DISK. ALL THE MATRICES SHOULD BE
C POSITIVE DEFINITE.)
C CALL WTMAT (A, W)
C
C INTEGRATE THE THREE ADJOINT VECTORS BACKWARDS. ALSO CALCULATE THE

```

WW

WW

WW

WW

```

C      INTEGRALS UII, UIJ, UJJ, VI, AND VJ.
C
      DO 501 I = 1, NBACK
      501 BP(I) = EB(I)
C
C      PUT THE INITIAL CONDITIONS FOR THE ADJOINT IN B.
      CALL ADJINI (B)
C
      PRMT(1) = TO + (NTF-1)*DT
      PRMT(2) = TO
      PRMT(3) = - DT
C
C      SUBROUTINE ADJN (T, B, BP) CALCULATES BP AS A FUNCTION OF B AND T.
C      SUBR ADJNO (T, B, BP, IHLF, N, PRMT, DJPRD, DC2W) IS CALLED ONCE FOR EACH
C      TIME INTERVAL. ADJNO STORES GISU AND GJSU ON THE DISK AND PRINTS
C      ANY DESIRED TRAJECTORY INFORMATION.
C      CALL HPCG (PRMT, B, BP, NBACK, IHLF, ADJN, ADJNO, AUX, DJ, DC2)
C
C      CALCULATE PII, PIJ, PJJ
C
C      CALCULATE WII
C
      IF (NPP) 620, 620, 602
      602 DO 603 I = 1, NPP
      DO 603 J = 1, NPP
      603 PII(I,J) = PII(I,J) + BETA(I)*W(I,J)*BETA(J)*4.
C
C      COMPUTE AII, AIJ, AJJ
C
      620 IF (NPARA) 640, 640, 631
      631 IF (NCONS) 632, 632, 621
      621 DO 623 I = 1, NCONS
      DO 623 J = 1, NPARA
      Z = 0.
      DO 622 K = 1, NPARA
      622 Z = Z + VI(I,K)*A(K,J)
      623 TEMP(I,J) = Z
      DO 625 I = 1, NCONS
      DO 625 J = 1, NCONS
      Z = PII(I,J)
      DO 624 K = 1, NPARA
      624 Z = Z + TEMP(I,K)*VI(J,K)
      625 PII(I,J) = Z
      DO 627 I = 1, NCONS
      Z = PIJ(I)

```

```

DO 626 J = 1, NPARA
  626 Z = Z + TEMP(I,J)*VJ(J)
  627 PIJ(I) = Z
  632 DO 629 I = 1, NPARA
    Z = 0.
  DO 628 J = 1, NPARA
    628 Z = Z + VJ(J)*A(J,I)
  629 TEMPV(I) = Z
  DO 630 I = 1, NPARA
    630 PJJ = PJJ + TEMPV(I)*VJ(I)
  640 IF (NCONS - 1) 700, 642, 641
  641 CALL MINV (PII, NCONS, Z, LSCRACH, MSCRACH)
    GO TO 700
  642 PII(1,1) = 1./PII(1,1)
C
C   SELECT DC AND DI. RETURN DC2 = DC**2
  700 CALL SELCT (DC2, DI)
C
C   CALCULATE PREDICTED DJ
C
  IF (NCONS) 720, 720, 709
  720 RAD = SQRT(DC2/PJJ)
  DJ = - RAD*PJJ
  GO TO 830
  709 DO 711 I = 1, NCONS
    Z = 0.
    DO 710 J = 1, NCONS
      710 Z = Z + PII(I,J)*DI(J)
    711 TEMPV(I) = Z
    Z = 0.
    DJ = 0.
    DJ = DJ + PIJ(I)*TEMPV(I)
    DO 712 I = 1, NCONS
      712 Z = Z + TEMPV(I)*DI(I)
    RAD = DC2 - Z
    IF (RAD) 713, 800, 801
    713 Z = SQRT(DC2/Z)
    DO 714 I = 1, NCONS
      714 DI(I) = DI(I)*Z
C
C   COMPUTE DA, DBETA
C
  800 RAD = 0.
  GO TO 810
  801 DO 803 I = 1, NCONS

```

WW

WW

WW

WW

WW

WW


```

Z = 0.
DO 802 J = 1, NCONS
  802 Z = Z + PIJ(J)*PII(J,I)
  803 TEMPV(I) = Z
  Z = PJJ
DO 804 I = 1, NCONS
  804 Z = Z - TEMPV(I)*PIJ(I)
  RAD = RAD/Z
  RAD = SORT(RAD)
C
C   STORE INV(PII)*(DI+PIJ*RAD) IN PIJ
C
  810 DO 811 I = 1, NCONS
    811 TEMPV(I) = DI(I) + PIJ(I)*RAD
    DJ = DJ - RAD*Z
    DO 813 I = 1, NCONS
      Z = 0.
    DO 812 J = 1, NCONS
      812 Z = Z + PII(I,J)*TEMPV(J)
      813 PIJ(I) = Z
      IF (NPP) 830, 830, 820
C
C   COMPUTE DBETA
C
  BETA(I) AND BNEW(I) ARE EQUIVALENT. BE SURE TO USE EQUIVALENCE
  STATEMENT
C
  820 DO 821 I = 1, NPP
    821 TEMPV(I) = 2.*BETA(I)*PIJ(I)
    DO 823 I = 1, NPP
      Z = 0.
    DO 822 J = 1, NPP
      Z = Z + W(I,J)*TEMPV(J)
      BOLD(I) = BNEW(I)
      823 BNEW(I) = BNEW(I) + Z
C
C   COMPUTE DA
C
  830 IF (NPARA) 900, 900, 831
  831 DO 833 I = 1, NPARA
    Z = - VJ(I)*RAD
    IF (NCONS) 833, 833, 836
  836 DO 832 J = 1, NCONS
    832 Z = Z + VI(J,I)*PIJ(J)
  833 TEMPV(I) = Z

```

WW
WW

WW

WW
WW

WW

```

DO 835 I = 1, NPARA
  Z = 0.
DO 834 J = 1, NPARA
  834 Z = Z + A(I,J)*TEMPV(J)
  AOLD(I) = ANEW(I)
  835 ANEW(I) = ANEW(I) + Z
C
C   COMPUTE DU(T)
C
  900 N = 1
C
  INPUT X, UNEW, UOLD, U, GISU, AND GJSU FROM RECORD N.
  901 CALL READ
C
  DO 903 I = 1, NCONT
    Z = - GJSU(I)*RAD
    IF (NCONS) 903, 903, 908
  908 DO 902 J = 1, NCONS
    902 Z = Z + GISU(J,I)*PIJ(J)
    903 TEMPV(I) = Z
  DO 905 I = 1, NCONT
    Z = 0.
  DO 904 J = 1, NCONT
    904 Z = Z + U(I,J)*TEMPV(J)
    UOLD(I) = UNEW(I)
  905 UNEW(I) = UNEW(I) + Z
C
C   OUTPUT X, UNEW, UOLD, U, GISU, AND GJSU INTO RECORD N.
C   CALL WRITE
C
  N = N + 1
  999 IF (N - NTF) 901, 901, 210
C
  1003 DC2 = .81*DC2
  IF (DC2 - STOPC) 9999, 9999, 1092
  1092 GO TO 400
  1002 XJNEW = XJOLD
  IF (NCONS) 1006, 1006, 1004
  1004 DO 1005 I = 1, NCONS
    1005 XJNEW(I) = XJOLD(I)
  1006 IF (NPARA) 1028, 1028, 1007
  1007 DO 1008 I = 1, NPARA
    1008 ANEW(I) = AOLD(I)
  1028 IF (NPP) 1011, 1011, 1009
  1009 DO 1010 I = 1, NPP

```

WW

WW

WW

WW

WW

WW

WW


```

C      EVALUATE Y AND DERY AT PRMT(1), PRMT(1)+H, PRMT(1)+2H, PRMT(1)+3H
5      DO 10 N = 1, 3
        CALL FCT (X, Y, DERY)
        IF (N - 1) 11, 11, 12
11     CALL OUTP (X, Y, DERY, IHLF, NDIM, PRMT, DJW, DC2W)
12     IF (PRMT(5)) 4, 6, 4
6      DO 9 I = 1, NDIM
        AUX(N,I) = Y(I)
9      AUX(N+4,I) = DERY(I)

C      HEUN *****$*****
C      DO 101 I = 1, NDIM
101     Y(I) = AUX(N,I) + H*AUX(N+4,I)
        X = X + H
        CALL FCT (X, Y, DERY)
        DO 102 I = 1, NDIM
102     Y(I) = AUX(N,I) + .5*H*(AUX(N+4,I) + DERY(I))
10     CONTINUE

C      REFINE THE HEUN VALUES *****
C      21 N = 1
        CALL FCT (X, Y, DERY)
        X = PRMT(1)
        DO 22 I = 1, NDIM
        AUX(8,I) = DERY(I)
22     Y(I) = AUX(1,I) + H*(.375*AUX(5,I) + .7916667*AUX(6,I)
1      - .2083333*AUX(7,I) + .04166667*DERY(I))
23     X = X + H
        N = N + 1
        CALL FCT (X, Y, DERY)
        CALL OUTP (X, Y, DERY, IHLF, NDIM, PRMT, DJW, DC2W)
        IF (PRMT(5)) 4, 24, 4
24     IF (N - 4) 25, 204, 204
25     DO 26 I = 1, NDIM
        AUX(N,I) = Y(I)
26     AUX(N+4,I) = DERY(I)
        IF (N - 3) 27, 29, 204

C      DO 28 I = 1, NDIM
        DELT = AUX(6,I) + AUX(6,I)
        DELT = DELT + DELT
28     Y(I) = AUX(1,I) + .3333333*H*(AUX(5,I) + DELT + AUX(7,I))
        GO TO 23

C      DO 30 I = 1, NDIM

```

```

DELT = AUX(6,I) + AUX(7,I)
DELT = DELT + DELT + DELT
30 Y(I) = AUX(1,I) + .375*H*(AUX(5,I) + DELT + AUX(8,I))
GO TO 23

C
C START HAMMINGS MODIFIED PREDICTOR CORRECTOR *****
200 DO 203 N = 2, 8
DO 203 I = 1, NDIM
203 AUX(N-1,I) = AUX(N,I)

C
204 DO 205 I = 1, NDIM
AUX(4,I) = Y(I)
205 AUX(8,I) = DERY(I)
X = X + H
DO 207 I = 1, NDIM
DELT = AUX(1,I) + 1.333333*H*(AUX(8,I) + AUX(3,I) - AUX(7,I)
1 + AUX(6,I) + AUX(6,I))
Y(I) = DELT - .9256198*AUX(10,I)
207 AUX(10,I) = DELT

C
CALL FCT (X, Y, DERY)

C
DO 208 I = 1, NDIM
DELT = .125*(9.*AUX(4,I) - AUX(2,I) + 3.*H*(DERY(I) + AUX(8,I)
1 + AUX(8,I) - AUX(7,I)))
AUX(10,I) = AUX(10,I) - DELT
208 Y(I) = DELT + .07438017*AUX(10,I)

C
DELT = 0.
DO 209 I = 1, NDIM
209 DELT = DELT + AUX(9,I)*ABS(AUX(10,I))

C
PRMT(4) IS APPROXIMATELY THE GLOBAL ERROR
IF (PRMT(4) - DELT) 215, 210, 210
215 PRMT(4) = DELT

C
210 CALL FCT (X, Y, DERY)
CALL OUTP (X, Y, DERY, IHLF, NDIM, PRMT, DJW, DC2W)
IF (PRMT(5)) 212, 213, 212
212 RETURN
213 IF (H*(X-PRMT(2))) 214, 212, 212
214 IF (ABS(X-PRMT(2)) - .1*ABS(H)) 212, 200, 200
END

SUBROUTINE FO (T, B, BP, IHLF, M, PRMT, DJPRD, DC2W)
DIMENSION B(4), BP(4), PRMT(5)

```

WW

WW

```

DIMENSION IPLOT (41)
COMMON X(3), UNEW(1), UOLD(1), U(1,1), GISU(2,1), GJSU(1),
1 XINEW(2), XIOLD(2), XJNEW, XJOLD, GD, TD(2), TF, W1, W2, TO, DT,
2 NTF, N, NLIM, NITER, NEFF, NOLST, BIOLD
N = (T - TO)/DT + 1.5
IF (N-1) 1,1,2
1 PTIME = 0.
WRITE (3,25)
25 FORMAT (6H1 TIME, 16X, 15HSTATE VARIABLES, 21X, 5HORBIT, 9X,
1 7HCONTROL, 5X, 3H-60, 7X, 3H-30, 8X, 1H0, 8X, 3H+30, 7X, 3H+60, /
2 7H DAY HR, 5X, 7HRAD VEL, 3X, 7HCIR VEL, 3X, 6HRADIUS, 5X,
3 SHANGLE, 10X, 6HENERGY, 9X, 5HANGLE, 7X, 1H$, 9X, 1H$, 9X, 1H$,
4 9X, 1H$, 9X, 1H$, /)
IFF = 0
2 X(1) = B(1)
X(2) = B(2)
X(3) = B(3)
CALL WRITE
IF (T-PTIME) 4,3,3
3 PTIME = PTIME + 172 800.
IDAY = T*.000 011 574 07 + .020 833 33
IHOUR = (T - FLOAT(IDAY)*86 400.)*.000 277 777 77 + .5
H = .001*(B(1)**2 + B(2)**2) - 264.9811/B(3)
DEGB4 = B(4)*57.295 78
DEGU = UNEW(1)*57.295 78
I = DEGU*.33333333 + 21.5
IF (I) 21, 21, 22
I = 1
21 IF (42 - I) 23, 23, 24
22 I = 41
23 DO 14 J = 1, 41
24 IPLOT(J) = 1H
14 IPLOT(21) = 1H.
IPLOT(1) = 1H*
15 IF (I - 21) 15, 20, 16
I = I+1
DO 17 J = 1, 20
17 IPLOT(J) = 1H-
GO TO 20
16 I = I-1
DO 19 J = 22, 1
19 IPLOT(J) = 1H+
20 WRITE (3,5) IDAY, IHOUR, (B(1),1=1,3), DEGB4, H, DEGU, IPLOT
5 FORMAT (14, 13, 3X, 4F10.5, 2F15.5, 5X, 41A1)
4 IF (IFF) 11, 11, 12

```

WW

WW

WW

```

11 IF (B(1)) 13. 7. 7
13 IFF = 1
GO TO 7
12 IF (B(1)) 7. 9. 9
9 PRMT(5) = .001
NTF = N
H = .001*(B(1)**2 + B(2)**2) - 264.9811/B(3)
DEGB4 = B(4)*57.295 78
DEGU = UNEW(1)*57.295 78
TF = T - PRMT(3)*B(1)/(B(1)-BIOLD)
TFDAY=TF*.000 011 574 07 + .0005
DAYS = T*.000 011 574 07 + .0005
XJNEW = TF*W1 + (B(4) - BP(4)*(T-TF))*W2
XINEW(1) = B(2) - BP(2)*(T - TF) - 35.0
XINEW(2) = B(3) - BP(3)*(T - TF) - 108.2
GO = (W1 + W2*BP(4))/BP(1)
TD(1) = BP(2)/BP(1)
TD(2) = BP(3)/BP(1)
DC222=SQRT(DC2W)
WRITE(3,33) DAYS, (B(1),I=1,3), DEGB4, H, DEGU
33 FORMAT (1X, F7.3, 2X, 4F10.5, 2F15.5)
DEGB4 = DEGB4 - BP(4)*(T - TF)*57.295 78
WRITE (3,10) NITER, NEFF, TF, TFDAY, DEGB4, XJNEW, XINEW
10 FORMAT(3H-N=,I3.5H, NE=,I3.5H, TF=,F10.0,4HSEC=, F8.3,5H DAYS,
18H, THETA=, F10.4, 7H, JNEW=, F10.5,8H, I1NEW=, F10.5, 8H, I2NEW=,
2F10.5)
IF (NOLST) 7. 7. 77
TRUDJ=XJNEW-XJOLD
IF (DJPRD) 60. 55. 60
55 ERODJ = 9999.
GO TO 65
60 ERODJ=ABS((DJPRD-TRUDJ)/DJPRD)
IF (ERODJ - 9999.) 65. 65. 55
65 WRITE(3,78) NTF,DJPRD,TRUDJ,XJOLD,XIOLD(1),XIOLD(2),DC222,ERODJ
78 FORMAT (5H NTF=, I5. 7H, THE, 14H PREDICTED DJ=, F10.5,
113H, ACTUAL DJ =, F10.5, 7H, JOLD=,
2F10.5,8H, I1OLD=,F10.5,8H, I2OLD=,F10.5,/,
314H DC USED =,F10.5,30H, RELATIVE ERROR IN DJ =,F10.5)
BIOLD = B(1)
RETURN
END
SUBROUTINE ADJN1(B)
DIMENSION B(19)
DO 1 I=1,19
1 B(I) = 0.

```

```

B(3) = 1.
B(6) = 1.
B(10) = 1.
RETURN
END
SUBROUTINE F(T, B, BP)
  DIMENSION B(4), BP(4)
  COMMON X(3), UNEW(1), UOLD(1), U(1,1), GISU(2,1), GJSU(1),
1  XINEW(2), XIOLD(2), XJNEW, XJOLD, GD, TD(2), TF, W1, W2, TO, DT,
2  NTF, N, NLIM, NITER, NEFF, NOLST, BIOLD
  N = (T - TO)/DT + 1.5
  IF (N - NTF) 1, 1, 2
1  CALL READ
2  SINU = SIN(UNEW(1))
  COSU = ABS(COS(UNEW(1)))
C
C  B(1)--RADIAL VELOCITY, KM/S
C  B(2)--CIRCUMFERENTIAL VELOCITY, KM/S
C  B(3)--RADIUS, GM (149.6GM = 1 ASTRONOMICAL UNIT)
C  B(4)--ANGLE, RADIAN
C
  THRUST = .044 76*(COSU/B(3))**2
C      POLAR COORDINATE      THRUST      GRAVITY
  BP(1) = .000 001*B(2)**2/B(3) + THRUST*COSU - .132 49/B(3)**2
  BP(2) = -.000 002*B(1)*B(2)/B(3) + THRUST*SINU
  BP(3) = .000 001*B(1)
  BP(4) = .000 001*B(2)/B(3)
  RETURN
END
SUBROUTINE ADJN (T, B, BP)
  REAL LX
  DIMENSION B(19), BP(19)
  COMMON X(3), UNEW(1), UOLD(1), U(1,1), GISU(2,1), GJSU(1),
1  XINEW(2), XIOLD(2), XJNEW, XJOLD, GD, TD(2), TF, W1, W2, TO, DT,
2  NTF, N, NLIM, NITER, NEFF, NOLST, BIOLD
  EQUIVALENCE (FU1,FX12), (FU2,FX13)
  N = (T - TO)/DT + 1.5
  IF (N) 1, 1, 2
1  N = 1
2  CALL READ
  SINU = SIN(UNEW(1))
  COSU = ABS(COS(UNEW(1)))
  FX12 = .000 002*X(2)/X(3)
  FX13 = -.000 001*(X(2)/X(3))**2 + (.264 98 - .089 52*COSU**3)
1  /X(3)**3

```



```

C      FX21 = -FX12
C      FX22 = -.000 002*X(1)/X(3)
C      FX23 = .000 002*X(1)*X(2)/X(3)**2 - .089 52*SINU*COSU**2/X(3)**3
C
C      FX31 = .000 001
C
C      D(LAMBDA I) = - (LAMBDA I)*(PARTIAL F WRT X) - (PARTIAL M WRT X)
C      BP(1) = - B(3)*FX21 - B(5)*FX31
C      BP(2) = - B(4)*FX21 - B(6)*FX31
C      BP(3) = - B(1)*FX12 - B(3)*FX22
C      BP(4) = - B(2)*FX12 - B(4)*FX22
C      BP(5) = - B(1)*FX13 - B(3)*FX23
C      BP(6) = - B(2)*FX13 - B(4)*FX23
C
C      D(LAMBDA J) = - (LAMBDA J)*(PARTIAL F WRT X) - (PARTIAL L WRT X)
C      BP(7) = - B(8)*FX21 - B(9)*FX31
C      LX = W2/X(3)*.000 001
C      BP(8) = - B(7)*FX12 - B(8)*FX22
C      LX = - W2*X(2)/X(3)**2*.000 001
C      BP(9) = - B(7)*FX13 - B(8)*FX23
C
C      D(LAMBDA S) = - (LAMBDA S)*(PARTIAL F WRT X)
C      BP(10) = - B(11)*FX21 - B(12)*FX31
C      BP(11) = - B(10)*FX12 - B(11)*FX22
C      BP(12) = - B(10)*FX13 - B(11)*FX23
C
C      FU1 = -.134 28*SINU*(COSU/X(3))**2
C      FU2 = .044 76*(1. - 3.*SINU**2)*COSU/X(3)**2
C
C      (GISU) = ((LAMBDA I) - (TD)*(LAMBDA S))*(PARTIAL F WRT U)
C      GISU(1,1) = (B(1) - TD(1)*B(10))*FU1 + (B(3) - TD(1)*B(11))*FU2
C      GISU(2,1) = (B(2) - TD(2)*B(10))*FU1 + (B(4) - TD(2)*B(11))*FU2
C
C      (GJSU) = ((LAMBDA J) - (GD)*(LAMBDA S))*(PARTIAL F WRT U)
C      GJSU(1) = (B(7) - GD*B(10))*FU1 + (B(8) - GD*B(11))*FU2
C
C      D(PII) = - (GISU)*INV(U)*T(GISU)
C      BP(13) = - GISU(1,1)*U(1,1)*GISU(1,1)
C      BP(14) = - GISU(2,1)*U(1,1)*GISU(1,1)
C      BP(15) = BP(14)
C      BP(16) = - GISU(2,1)*U(1,1)*GISU(2,1)
C
C      D(PIJ) = - (GJSU)*INV(U)*T(GJSU)
C      BP(17) = - GJSU(1)*U(1,1)*GISU(1,1)

```



```

BP(18) = - GJSU(1)*U(1,1)*GISU(2,1)
C
D(PJJ) = - (GJSU)*INV(U)*T(GJSU)
C
BP(19) = - GJSU(1)*U(1,1)*GJSU(1)
C

```

```

RETURN
END

```

```

SUBROUTINE ADJNO (T,B,BP,HLF,M,PRMT, DJPRD, DC2W)

```

```

DIMENSION B(19),BP(19),PRMT(5)

```

```

CALL WRITE

```

```

RETURN

```

```

END

```

```

SUBROUTINE SELCT(DC2,DI)

```

```

DIMENSION DI(2)

```

```

COMMON X(3), UNEW(1), UOLD(1), U(1,1), GISU(2,1), GJSU(1),

```

```

1 XINEW(2), XIOLD(2), XJNEW, XJOLD, GD, TD(2), TF, W1, W2, TO, DT,

```

```

2 NTF, N, NLIM, NITER, NEFF, NOLST, BIOLD

```

```

IF(ABS(XINEW(1)) - .7000) 10, 10, 15

```

```

10 DI(1) = - XINEW(1)*.05

```

```

GO TO 20

```

```

15 DI(1) = - XINEW(1)*.2

```

```

20 IF(ABS(XINEW(2)) - 1.082) 25, 25, 30

```

```

25 DI(2) = - XINEW(2)*.1

```

```

GO TO 35

```

```

30 DI(2) = - XINEW(2)*.2

```

```

35 IF(NEFF-1) 4, 4, 3

```

```

3 DC2 = DC2*.9025

```

```

4 RETURN

```

```

END

```

```

SUBROUTINE WTMAT (A, W)

```

```

RETURN

```

```

END

```

```

SUBROUTINE READ

```

```

COMMON /1/ CBUF(9,2000)

```

```

COMMON X(3), UNEW(1), UOLD(1), U(1,1), GISU(2,1), GJSU(1),

```

```

1 XINEW(2), XIOLD(2), XJNEW, XJOLD, GD, TD(2), TF, W1, W2, TO, DT,

```

```

2 NTF, N, NLIM, NITER, NEFF, NOLST, BIOLD

```

```

IF(N) 2,2,1

```

```

1 IF (NLIM - N) 4, 3, 3

```

```

3 X(1) = CBUF(1,N)

```

```

X(2) = CBUF(2,N)

```

```

X(3) = CBUF(3,N)

```

```

UNEW(1) = CBUF(4,N)

```

```

UOLD(1) = CBUF(5,N)

```

```

U(1,1) = CBUF (6,N)

```

```

GJSU(1) = CBUF(7,N)
GISU(1,1) = CBUF(8,N)
GISU(2,1) = CBUF(9,N)
RETURN
2 WRITE (3,5) N
5 FORMAT (16H ILLEGAL RECORD,, 15, 11H, REQUESTED)
N = 1
GO TO 3
4 WRITE (3,5) N
N = NLIM
GO TO 3
END
SUBROUTINE WRITE
COMMON /1/ CBUF(9,2000)
COMMON X(3), UNEW(1), UOLD(1), U(1,1), GISU(2,1), GJSU(1),
1 XNEW(2), XIOLD(2), XJNEW, XJOLD, GD, TD(2), TF, W1, W2, TO, DT,
2 NTF, N, NLIM, NITER, NEFF, NOLST, BIOLD
IF(N) 2,2,1
1 IF (NLIM - N) 4, 3, 3
3 CBUF(1,N) = X(1)
CBUF(2,N) = X(2)
CBUF(3,N) = X(3)
CBUF(4,N) = UNEW(1)
CBUF(5,N) = UOLD(1)
CBUF(6,N) = U(1,1)
CBUF(7,N) = GJSU(1)
CBUF(8,N) = GISU(1,1)
CBUF(9,N) = GISU(2,1)
RETURN
2 WRITE (3,5) N
5 FORMAT (16H ILLEGAL RECORD,, 15, 11H, REQUESTED)
N = 1
GO TO 3
4 WRITE (3,5) N
N = NLIM
GO TO 3
END
SUBROUTINE MINV(A,N,D,L,M)
DIMENSION A(1),L(1),M(1)
SEARCH FOR LARGEST ELEMENT
D=1.0
NK=-N
DO 80 K=1,N
NK=NK+N
L(K)=K
MINV 1
MINV 2
MINV 3
MINV 4
MINV 5
MINV 6
MINV 7
MINV 8

```

```

M(K)=K
KK=NK+K
BIGA=A(KK)
DO 20 J=K,N
IZ=N*(J-1)
DO 20 I=K,N
IJ=IZ+I
10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
15 BIGA=A(IJ)
L(K)=I
M(K)=J
20 CONTINUE
      INTERCHANGE ROWS
      J=L(K)
      IF(J-K) 35,35,25
25 KI=K-N
DO 30 I=1,N
KI=KI+N
HOLD=-A(KI)
JI=KI-K+J
A(KI)=A(JI)
30 A(JI)=HOLD
      INTERCHANGE COLUMNS
35 I=M(K)
IF(I-K) 45,45,38
38 JP=N*(I-1)
DO 40 J=1,N
JK=NK+J
JI=JP+J
HOLD=-A(JK)
A(JK)=A(JI)
40 A(JI)=HOLD
      DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
      CONTAINED IN BIGA)
45 IF(BIGA) 48,46,48
46 D=0.0
RETURN
48 DO 55 I=1,N
IF(I-K) 50,55,50
50 IK=NK+I
A(IK)=A(IK)/(-BIGA)
55 CONTINUE
      REDUCE MATRIX
C DO 65 I=1,N
IK=NK+I

```

```

MINV 9
MINV 10
MINV 11
MINV 12
MINV 13
MINV 14
MINV 15
MINV 16
MINV 17
MINV 18
MINV 19
MINV 20
MINV 21
MINV 22
MINV 23
MINV 24
MINV 25
MINV 26
MINV 27
MINV 28
MINV 29
MINV 30
MINV 31
MINV 32
MINV 33
MINV 34
MINV 35
MINV 36
MINV 37
MINV 38
MINV 39
MINV 40
MINV 41
MINV 42
MINV 43
MINV 44
MINV 45
MINV 46
MINV 47
MINV 48
MINV 49
MINV 50
MINV 51
MINV 52
MINV 53

```

```

IJ=I-N
DO 65 J=1,N
  IJ=IJ+N
  IF(I-KD 60,65,60
60 IF(J-K) 62,65,62
62 KJ=IJ-I+K
  A(IJ)=A(IK)*A(KJ)+A(IJ)
65 CONTINUE
  C      DIVIDE ROW BY PIVOT
    KJ=K-N
    DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
70 A(KJ)=A(KJ)/BIGA
75 CONTINUE
  C      PRODUCT OF PIVOTS
    D=D*BIGA
  C      REPLACE PIVOT BY RECIPROCAL
    A(KK)=1.0/BIGA
80 CONTINUE
  C      FINAL ROW AND COLUMN INTERCHANGE
    K=N
100 K=(K-1)
    IF(K) 150,150,105
105 I=L(K)
    IF(I-K) 120,120,108
108 JQ=N*(K-1)
    JR=N*(I-1)
    DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
110 A(JI) =HOLD
120 J=M(K)
    IF(J-K) 100,100,125
125 KI=K-N
    DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
130 A(JI) =HOLD
    GO TO 100
150 RETURN

```

MINV 54
MINV 55
MINV 56
MINV 57
MINV 58
MINV 59
MINV 60
MINV 61
MINV 62
MINV 63
MINV 64
MINV 65
MINV 66
MINV 67
MINV 68
MINV 69
MINV 70
MINV 71
MINV 72
MINV 73
MINV 74
MINV 75
MINV 76
MINV 77
MINV 78
MINV 79
MINV 80
MINV 81
MINV 82
MINV 83
MINV 84
MINV 85
MINV 86
MINV 87
MINV 88
MINV 89
MINV 90
MINV 91
MINV 92
MINV 93
MINV 94
MINV 95
MINV 96
MINV 97
MINV 98

3	1	4	2	0	0	0	1	0		
		29C76		149.6	0.					
•										
•27700831	•08310249	•01662049	•62326869							
•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526
•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526
•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526	•0526
•5										
200.	•0001	0.		21600.	2.	2.				
•	17280000.	0.		1.	--.785498051					
•										

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 03178 0228