Report No. 67-16
June 1967

AN APPLICATION OF LINEAR PROGRAMMING TO MINIMUM FUEL OPTIMAL CONTROL

C. M. Waespy

Department of Engineering
University of California
Los Angeles, California

FOREWORD

The research described in this report, "An Application of Linear Programming to Minimum Fuel Optimal Control," Number 67-16, by Charles M. Waespy, was carried out under the direction of C. T. Leondes, Principal Investigator, in the Department of Engineering, University of California, Los Angeles.

This project is sponsored in part by the National Aeronautics and Space Administration under Grant NsG-237-62 to the Institute of Geophysics and Planetary Physics of the University.

This report was the basis for a Doctor of Philosophy dissertation submitted by the author.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF SYMBOLS

$x(t)$	a vector function of time
$x_i(t)$	the i^{th} component of vector $x(t)$
$\lvert u(t) \rvert$	the absolute value of a vector $u(t)$
A	the matrix A with elements a_{ij}
A_j	the j^{th} column of matrix A
cx	the inner product of row vector c and column vector x
π	an m-dimensional row vector of dual or costate variables
π_i	the i^{th} component of π
E_n	n dimensional Euclidean Space
Ω	convex set of recoverable states
P, Q, U	column vectors used in Generalized Programming
$\Phi(\tau)$	transition matrix
$\Phi^{-1}(\tau)$	inverse of the transition matrix
$y(k)$	a vector function of the index k
$\lVert u \rVert_q$	the q norm of vector u
$x \epsilon E_n$	x is a member of the set E_n
$\underline{R}_T$	position vector of a target point
$\underline{R}_S$	position vector of a controlled satellite
R_T	magnitude of the vector $\underline{R}_T$
R_S	magnitude of the vector $\underline{R}_S$
$\underline{w}$	orbital rate vector
w	magnitude of orbital rate vector $\underline{w}$
$\left[u_1(t), u_2(t)\right]$	column vector $u(t)$
$[R]$	the matrix R with elements r_{ij}

LIST OF FIGURES

LIST OF COMPUTATION SUMMARIES

CHAPTER I
INTRODUCTION

1.0 The Optimal Control Problem

Consider $x(t)$ to be a vector in a region S of m-dimensional space, and let it describe the "state" or condition of a dynamic system at some point $t \epsilon [t_o, t_f]$. The change in this state vector is expressed by a system of differential equations

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

where f is an m-dimensional vector function, and $u(t)$ is an r-dimensional control vector. The control vector, $u(t)$, is available to a designer to influence the evolution of the state vector $x(t)$. The range of $u(t)$ is a subset, U, of r-dimensional control space.

There are two manifolds in S, S_o and S_f, which contain the initial condition $x(t_o) = x_o$, and the final state $x(t_f) = x_f$ of the system, respectively.

The Optimal Control Problem is that of finding a $u^* \epsilon U$ which transfers the state x from $x_o \epsilon S_o$ to $x_f \epsilon S_f$ in such a manner that the functional

$$J = \int_{t_o}^{t_f} f_{m+1}[x(\tau), u(\tau)]\, d\tau \tag{1.2}$$

takes on a least possible value.

During the last ten years, the mathematical structure underlying this problem has been honed to a fine edge by various workers, employing a variety of points of view. The Calculus of Variations, the Maximum Principle, and the functional equation approach of Dynamic Programming have risen to provide means by which necessary (and in a few cases, sufficient) conditions for the optimal control function $u^*(t)$ can be derived. These are fully covered and described in detail in the literature.[1,2] A similar statement, however, cannot be made concerning the application of these conditions to compute optimal control functions for actual engineering problems. Part of the reason for this is that each method of deriving necessary conditions for the optimal control function $u^*(t)$ leads to a subsidiary computational problem which has proven troublesome in engineering practice. This is true even when the differential equation system Equation (1.1) is linear, and when the cost functional Equation (1.2) is specifically tailored to alleviate mathematical difficulties.

The general objective of the work to follow is to demonstrate how a relatively efficient computational technique, the simplex method of Linear Programming, can be applied to at least one type of optimal control problem which has significant application to the Aerospace Engineering field. The case to be treated is the so called Minimum Fuel Problem wherein the scalar integrand of Equation (1.2) is defined as follows

$$f_{m+1} = \sum_{i=1}^{r} |u_i(t)| \qquad (1.3)$$

The integrand of Equation (1.2) may take various forms. Two companions to the problem at hand are

$$f_{m+1} = \sum_{i=1}^{r} u_i^2(t) \qquad (1.4)$$

and

$$f_{m+1} = 1 \qquad (1.5)$$

These are termed the Minimum Energy Problem and the Minimum Time Problem respectively.

1.1 Optimal Control and the Adjoint Vector

For the above named problems, and for a dynamic process modeled by a system of linear differential equations

$$\dot{x}(t) = A(t)\, x(t) + B\, u(t) \ ; \quad |u(t)| \leq M \qquad (1.6)$$

the optimal control function $u^*(t)$ may be discontinuous at a finite number of points in time. Furthermore the control law, as a function of time, depends on the function M signum $\left[\pi^*(t)B\right]$. Here $\pi^*(t)$ is the solution to the homogeneous differential equation system adjoint to Equation (1.6). An appropriate initial condition to the adjoint system must be found in order to obtain $\pi^*(t)$.

We can sketch the role of $\pi^*(t)$ for the minimum energy, time, and fuel problems as shown in Figure 1-1. The central problem of optimal control specification is to determine the relation between $\pi^*(t)$ and the given conditions of the problem. In general, this can be done only by iterative numerical computation. One by-product of the work to follow is a detailed insight into the nature and characteristics of $\pi^*(t)$ as used in various approaches to the Minimum Fuel Problem.

1.2 Mathematical Programming

The Optimal Control Problem sketched in Figure 1-1 is a branch of a more general class, the Optimization Problem. In this problem it is desired to minimize or maximize a function of several variables with the variables satisfying given constraints. In the course of applying mathematical optimization methods to the allocation of scarce resources for economic and industrial processes, another branch of the Optimization Problem has developed. This branch contains problems of Mathematical Programming. The scarce resources in this case are considered to be "programmed."

This branch is characterized by the existence of a large number of feasible solutions — those which meet basic constraint equations. From these we seek to select the one or many which minimize or maximize a given objective function. A special case of Mathematical Programming is the Linear Programming (LP) Problem. The LP problem is characterized by a large number of variables, satisfying linear algebraic constraints, and having a linear objective function.

Linear Programming has been applied with outstanding success in the economic control of large petroleum refineries and other process industries. It is an optimization technique which can handle tens of thousands of problem variables, constrained under both equalities and inequalities, with comparative ease. Current applications of Linear Programming have been almost entirely to problems of "static" optimization, wherein a plant functions at a relatively fixed operating point (determined by the program), and time changes to variables affecting the cost function are relatively infrequent. In a refinery, for example, the disturbance may be the weekly arrival of a tanker from an oil field which has a different quality of crude oil than that usually received, necessitating a change in distillation settings to optimize the profit.

The successful and practical applications of Linear Programming to problems of "static" optimization are widespread and well-known. Not so well explored is its potential usefulness to what may be called "dynamic" optimization, wherein constraints may arise from differential or discrete time state equations of the type encountered in the optimal control field. Control problems are usually analyzed from a continuous function point of view, and the full power of functional analysis is brought to bear. There are exceptions of course, notably in the areas of sampled data and dynamic programming. But for the most part the point of view may be different from that of the algebraic approach of the linear programmer. One advantage of the latter approach is that the problem formulation, if achieved, is more directly applicable to digital computer solution than that usually obtained from the analytic function viewpoint. Another advantage stems from the fact that powerful LP codes capable of solving large scale problems already exist. One company proprietary code is reported to have solved a problem in Linear Programming involving 10^6 variables in over 20,000 constraint equations. The solution employed the decomposition principle developed by Wolfe and Dantzig; solution time was estimated to be 45 minutes.

Existing LP codes are flexible and efficient. In many cases, they will permit a parametric study of the solution to be made with only a small increase in computer time. Newer codes will exploit this parametric capability to perform stochastic programming.

In view of the computational demands of optimal control theory, and the computational power of LP methods, it appears worth-while to investigate the possible comparisons and interconnections which can be made between the two. This document will report the results of one such investigation. Others will be reported by reference.

1.3 Scope of the Investigation

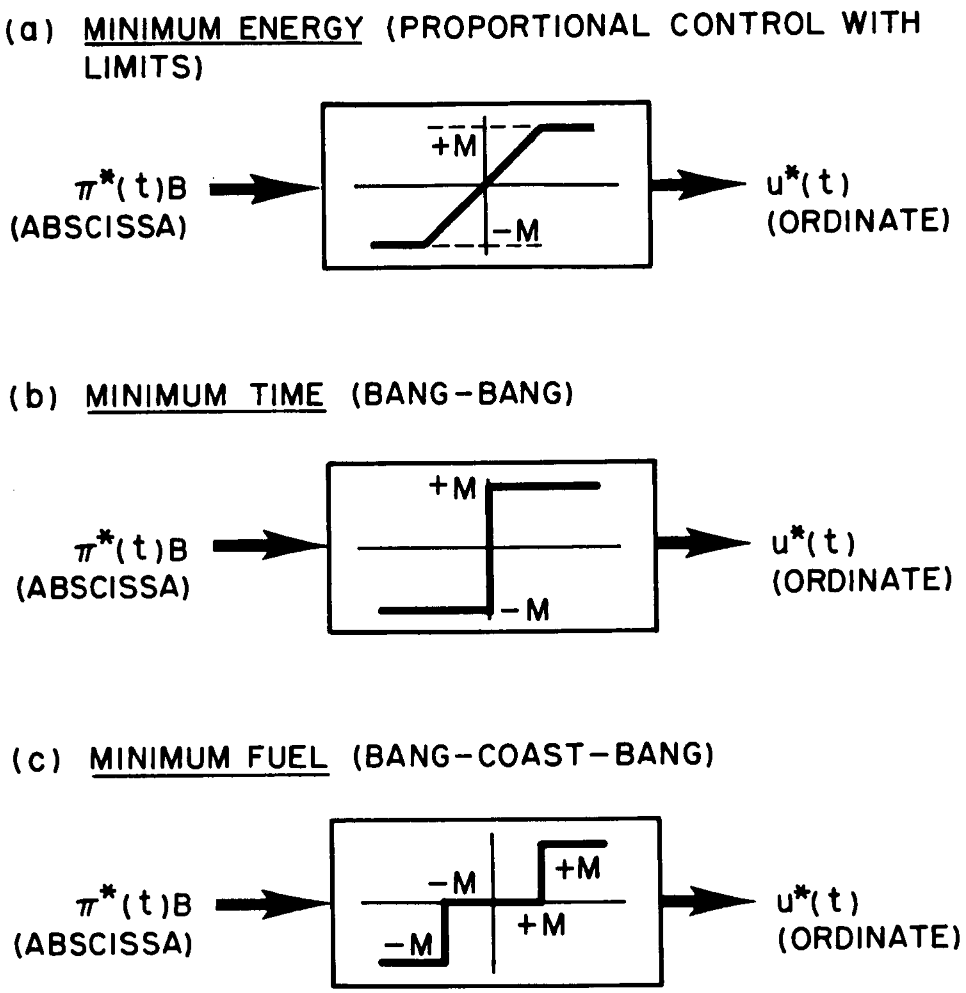
The body of the work to follow is composed of three major parts:

(a) Chapter II: This chapter introduces the elements of the Linear Programming computation method. The simplex algorithm, duality, decomposition, and generalized programming are described from a heuristic point of view. The treatment of the generalized programming concept is presented in a form which will allow future extension to nonlinear problems.

(b) Chapters III, IV and V: These chapters describe the application of the simplex method to the linear deterministic minimum fuel problem. To facilitate the description of a wide variety of associated concepts, only the minimum fuel problem is treated. However, this is not the only applicable and plausible application of LP to optimal control theory. For example, Whalen[3] suggests about sixteen other useful formulations. Torng[4] solves a two-state variable minimum fuel-minimum time discrete problem by LP. Sakawa and Hayashi[5] use LP in an iterative procedure yielding time optimal solutions.

The main objective of Chapters III, IV, and V is to demonstrate how the dual formulation of the LP problem unites some of the well known approaches for solving linear deterministic optimal control problems. These include Neustadt' s Geometric, the Maximum Principle, and the Functional Analysis approaches. Dantzig' s recently proposed application of Generalized Programming is also included. For the chosen problem, the optimal dual LP solution provides the initial condition to the state equation of the discrete maximum principle. The dual LP solution also provides the Lagrange multipliers necessary in the Functional Analysis approach to the optimal control problem. This result will permit the application of this technique to higher order systems previously not feasible because of computational difficulties.

(c) Chapter VI: This chapter applies Linear Programming methods to the terminal homing guidance problem for spacecraft. Typical rendezvous trajectories based on linearized (Wiltshire-Clohessey) equations of motion are calculated. Under the assumptions made, these minimize fuel consumption. The capabilities and limitations of the LP method, as well as solution characteristics to this problem, are demonstrated.



(a) MINIMUM ENERGY (PROPORTIONAL CONTROL WITH LIMITS)

(b) MINIMUM TIME (BANG-BANG)

(c) MINIMUM FUEL (BANG-COAST-BANG)

MINIMUM ENERGY, TIME, AND FUEL CONTROL LAWS

FIGURE 1-1

CHAPTER II

ELEMENTS OF LINEAR PROGRAMMING

2.0 Introduction

Linear Programming was conceived in 1947, although its potential, according to Dantzig, may have been recognized by Fourier in the period around 1823. It is an optimization technique characterized by (a) a relatively simple mathematical structure, (b) prodigious computing power and (c) the widest range of application to actual ("real world") engineering and industrial problems. The ordinary Linear Programming (LP) Problem has the following form:

(a) Given a linear objective function

$$f(x) = \sum_{i=1}^{n} c_i x_i \tag{2.1}$$

and the m constraint equations

$$Ax \; T \; b$$

where A is an $(m \times n)$ constant matrix, and T represents either $=$, $\leqq$, or $\geqq$.

(b) Find that x which maximizes or minimizes $f(x)$.

Much of what can be said of "Linear Programming and Extensions" is found in Dantzig' s book under that title.[6] This chapter selects, and merely sketches some of those elements of Dantzig' s theory which are applicable to the work which follows. Although the concepts of linear programming are quite simple on the surface, there exists in the theory and its applications an elegant mathematical subtlety which is appreciated only by studying the subject from many points of view, both theoretical and practical. Dantzig' s book is an excellent compendium which can lead to this appreciation.

2.1 Systems of Linear Equations

Consider the matrix equation $Ax = b$. Any vector x which satisfies this equation is called a solution of the system. Let us consider an inhomogeneous set of m equations in n unknowns, $m \leq n$. Such a system may: (1) be inconsistent, that is, have no solutions or (2) be consistent, that is, have exactly one or an infinite number of solutions.

To investigate the nature of, and to obtain solutions to, a system of m equations in n unknowns, elementary operations, pivot operations, and Gauss Jordan elimination are frequently employed. Consider the equation set

$$\begin{array}{llll}
a_{11} x_1 + a_{12} x_2 + \ldots\ldots\ldots\ldots\ldots & + a_{1n} x_n = b_1 & & \\
a_{21} x_1 + a_{22} x_2 + \ldots\ldots\ldots\ldots\ldots & + a_{2n} x_n = b_2 & & \\
\vdots & \vdots & & (2.2) \\
a_{m1} x_1 + a_{m2} x_2 + \ldots\ldots\ldots\ldots\ldots & + a_{mn} x_n = b_m & &
\end{array}$$

Two elementary operations may be performed on this system: (1) multiply an equation by a non-zero constant and (2) add a multiple of one equation to any other. These row operations have the property that the original system of equations and the transformed system have the same solution set; i.e., they are equivalent. Note that elementary column operations do not have this property. Elementary operations on columns do not result in an equivalent system, but rather in a change of variables.

A pivot operation is a sequence of elementary row operations executed as follows:

(a) Select a variable of interest, x_s, such that its coefficient in the r'th equation, a_{rs}, does not equal zero.

(b) Divide the r' th equation by a_{rs} (elementary row operation # 1). This yields a coefficient of 1 for x_s in the r' th equation.

(c) Subtract a_{is} times the new r' th equation from each i' th equations, $i = 1, 2, \ldots, r-1, r+1, \ldots, m$ (elementary row operation #2). This yields a coefficient of 0 for x_s in all equations except the r' th (where it is 1).

If Equations (2.2) constitute the original system, the transformed system will be the following:

$$\begin{array}{lllll}
\bar{a}_{11}\, x_1 + \ldots\ldots & + 0\, x_s + \ldots\ldots\ldots & + \bar{a}_{1n}\, x_n = \bar{b}_1 & & \\
\vdots & \vdots & \vdots & & \\
\bar{a}_{r1}\, x_1 + \ldots\ldots & + 1\, x_s + \ldots\ldots\ldots & + \bar{a}_{rn}\, x_n = \bar{b}_r & & (2.3) \\
\vdots & \vdots & \vdots & & \\
\bar{a}_{m1}\, x_1 + \ldots\ldots & + 0\, x_s + \ldots\ldots\ldots & \bar{a}_{mn}\, x_n = \bar{b}_m & &
\end{array}$$

The new coefficients, $\bar{a}_{ij}$, are related to the old, a_{ij}, by

$$\bar{a}_{ij} = a_{ij} - a_{is}\,\frac{a_{rj}}{a_{rs}} \qquad i \neq r$$

$$\bar{a}_{ij} = \frac{a_{ij}}{a_{rs}} \text{ or } \bar{a}_{rj} = \frac{a_{rj}}{a_{rs}} \qquad i = r \qquad (2.4)$$

Gauss Jordan Elimination consists of a sequence of pivot operations defined above to yield an equivalent system of the following form.

$$\left[\begin{array}{cccc|l}
x_1 & & & & +\bar{a}_{1,r+1}\,x_{r+1}+\ldots\ldots+\bar{a}_{1,n}\,x_n=\bar{b}_1\\
& \ddots & & & \\
& & x_2 & & +\bar{a}_{2,r+1}\,x_{r+1}+\ldots\ldots+\bar{a}_{2,n}\,x_n=\bar{b}_2\\
& & & \ddots & \\
& & & x_r & +\bar{a}_{r,r+1}\,x_{r+1}+\ldots\ldots+\bar{a}_{2,n}\,x_n=\bar{b}_r\\ \hline
& & & & 0\quad x_{r+1}+\ldots\ldots+0\quad x_n=\bar{b}_{r+1}\\
& & & & 0\quad x_{r+1}+\ldots\ldots+0\quad x_n=\bar{b}_m
\end{array}\right] \qquad (2.5)$$

Several properties of the original system can be distinguished with the aid of this transformed, but equivalent, set of equations:

(1) Inconsistent System: The original system is inconsistent if $\bar{b}_{r+1}, \bar{b}_{r+2}, \ldots, b_m$ do not equal zero. The proof is immediate since $0 = \bar{b}_{r+1} \neq 0$ is a contradiction.

(2) Consistent System: The original system is consistent if, and only if, $\bar{b}_{r+1} = \bar{b}_{r+2} = \ldots. = \bar{b}_m = 0$. A consistent system has at least one solution. In this case, we can determine one of possibly many solutions by setting all variables $x_{r+1}, x_{r+2}, \ldots, x_n$ equal to zero.

(2a) Consistent System With Exactly One Solution: There is exactly one solution to the original system if the $\bar{a}_{ij} = 0$ for all $i = 1, 2, \ldots, r$ and all $j = r+1, r+2, \ldots, n$. This solution is $x = (\bar{b}_1, \bar{b}_2, \ldots. \bar{b}_r)$.

(2b) Consistent System With Infinite Number of Solutions: If $\bar{a}_{ij} \neq 0$ for some i, j such that $i = 1, 2 \ldots, r$ and $j = r+1, r+2, \ldots, n$ then the original system has an infinite number of solutions corresponding to the infinite domain of the real variable x_j for $j = r + 1, r + 2, \ldots, n$.

(2c) Consistent System With Redundant Equations: If, as in System (2.5) above, there appear equations with all zero coefficients and the corresponding b_i, $i = r+1, r+2, \ldots, m$ equal zero (this is implied by the requirement for consistency), then the original system contains equations which are redundant. Redundant equations are those which are linearly dependent on the first r equations. If this is not the case, i.e., $r = m$, then the original system is of full rank.

Linear Programming concerns itself almost exclusively with systems possessing property (2b), i.e., consistent systems with an infinite number of solutions. Faced with an infinity of solutions, we find it useful to devise a way of arranging them in some preferred order. In linear programming, this is done by assigning a linear cost $z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$ to each solution.

We then try to find the minimum or maximum of this function, subject to the other linear algebraic constraint equations. For many linear programming applications, negative solutions make no sense; hence, the variables x_j are restricted in most algorithms to be non-negative. This is a minor inconvenience in optimal control applications where the control variables, corresponding to the x_j' s, must be allowed both a positive and negative range. However, the non-negativity restriction aids in the formulation and coding of solution algorithms; hence, it is retained. At any rate, it can easily be circumvented by a change of variables.

$$x_j = x_j^+ - x_j^- ; \; x_j^+ \geq 0, \; x_j^- \geq 0$$

The next section will indicate how the Gauss Jordan procedure, supplemented by linear cost function and non-negativity restrictions, is used to obtain LP solutions.

2.2 The Simplex Method

The approaches to explaining the simplex method are as numerous and as varied as its extensions and applications. Here we follow the Dantzig notation[6] to present a broad overview of what is done to solve an LP problem. Except for the added requirements imposed by the cost function and the non-negativity constraints, arithmetic operations in one form of the simplex method are the same as those of Gauss Jordan elimination for solving a set of linear algebraic equations. Originally the problem may be stated as in Section 2.0.

> "Find a vector x to minimize or maximize a function $f(x)$, such that $Ax \; T \; b$ is satisfied."

T may represent equality or inequalities in either direction.

The given problem is first reduced to the Standard Form:

$$\text{Minimize } z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$

$$\begin{aligned} \text{Subject to } a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n &= b_1 \\ a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n &= b_2 \\ a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n &= b_n \\ x_j \geqq 0 \quad j = 1, 2, \ldots, n \end{aligned} \tag{2.6}$$

The Standard Form may be obtained from a given problem by (a) use of slack variables when T is $\leqq$ or $\geqq$, (b) change of variables $x_j = x_j^+ - x_j^-$, $x_j^+ \geq 0$, $x_j^- \geq 0$ when x_j is to range over both positive and negative values, and (c) multiplication by - 1 if f(x) is to be maximized under the original problem statement.

The simplex method consists of two parts, Phase I and Phase II, both employing the same algorithm consisting of pivot operations described previously, and optimality tests to be described below. The purpose of Phase I is to transform the Standard Form into Canonical Form. Phase II then operates on the Canonical Form to obtain an optimal solution (if one exists).

The Phase I operation also embodies a test to determine whether or not a feasible solution to the original problem does in fact exist. The Phase I procedure is simple enough in principle, but involves too detailed an explanation. Hence, it will not be described here. Instead, the reader is referred to any good textbook on Linear Programming, in particular Chapter 5 of Reference 6.

We now assume that Phase I has been successfully applied to the Standard Form. This results in an equivalent Canonical Form.

The Canonical Form with respect to variables x_1, $x_2, \dots, x_m$, $(-z)$ is as follows:

$$\left[\begin{array}{cccc|l|l}
x_1 & & & & +\bar{a}_{1,m+1}\,x_{m+1}+\dots\dots+\bar{a}_{1n}\,x_n & =\bar{b}_1\\
 & x_2 & & & +\bar{a}_{2,m+1}\,x_{m+1}+\dots\dots+\bar{a}_{2n}\,x_n & =\bar{b}_2\\
 & & \ddots & & & \\
 & & & x_m & +\bar{a}_{m,m+1}\,x_{m+1}+\dots\dots+\bar{a}_{m,n}\,x_n & =\bar{b}_m\\ \hline
 & & & (-z) & +\bar{c}_{m+1}\,x_{m+1}+\dots\dots+\bar{c}_n\,x_n & =-\bar{z}_o
\end{array}\right] \qquad (2.7)$$

where $x_j \geq 0$, $j = 1, 2, \ldots, n$, and the bar over the constant terms designates changes in their value resulting from the arithmetic operation under discussion.

When each $\bar{b}_i$, $i = 1, 2, \ldots, m$ is greater than or equal to zero, we have a feasible canonical form. We desire to minimize z. Note also that Phase I has eliminated redundant equations.

The variables $x_1, x_2, \ldots, x_m$, $(-z)$ are called the basic variables of this particular Canonical Form; all other variables are termed non-basic. A solution n-vector

$$x = (x_1, x_2, \ldots, x_m \; 0\;0\;0\;0\;0\;0)$$

with no more than m non-zero components is a basic feasible solution. If the number of non-zero components of the n-vector is precisely m, we have a non-degenerate basic feasible solution. If the value of one or more of the basic variables x_j $j = 1, 2, \ldots, m$ is zero (that is, $\bar{b}_i = 0$, for some $i = 1, 2, \ldots, m$), we have a degenerate basic feasible. The values $\bar{c}_j$ for $j \geq m + 1$ are called relative costs because they are costs associated with possible increase in non-basic variables relative to a particular (unique) canonical form.

The Canonical Form $\Big($Equation (2.7)$\Big)$ is devised to expedite both the implementation of tests for optimality as well as the calculation of the optimal solution. Its usefulness follows from the below-listed observations:

(1) If $x_j = 0$ for $j = m+1, m+2, \ldots, n$ we have a basic feasible solution (b.f.s.). A b.f.s. is non-degenerate if $\bar{b}_i \neq 0$ for all $i = 1, 2, \ldots, m$.

(2) If $\bar{c}_j > 0$ for $j = m+1, m+2, \ldots, n$ the b.f.s. in (1) above is optimal, because increasing any x_j, $j = m+1, m+2, \ldots, n$, will increase the value of z. The b.f.s. is unique, and no other b.f.s. will give the same value for z. This follows from the uniqueness property of the <u>Canonical Form</u> (Reference 6 Theorem I page 80).

(3) If $\bar{c}_j < 0$ for some $j = s \geq m+1$, it is possible to decrease the value of z by increasing the value of x_s (to an upper limit to be specified later). We must, of course, compensate for this increase in x_s by adjusting the values of the existing basic variables in order to preserve equality in all equations.

(4) If $\bar{c}_j = 0$ for some $j = s \geq m + 1$, there will be no effect on the value of z from any allowable change in x_s. In this case, we can vary x_s and even make it basic without changing the value of z.

Observation (3) is of interest because it provides a means of improving the solution to obtain a lower value of z. Given $\bar{c}_s < 0$ we obtain a smaller z by increasing x_s as much as possible. However, we must retain feasibility. The effect of increasing x_s on feasibility is seen by setting all variables <u>except</u> the current basic variables <u>and</u> x_s equal to zero. Thus, from Equations (2.7)

$$x_i = \bar{b}_i - \bar{a}_{is} x_s \qquad i = 1, 2, \ldots, m$$

Feasibility is maintained if $\bar{b}_i - \bar{a}_{is} x_s \geq 0$ for all $i = 1, 2, \ldots, m$. Hence, we can increase x_s up to

$$x_s = \theta \triangleq \underset{i}{\text{Min}} \left\{ \frac{\bar{b}_i}{\bar{a}_{is}} : \bar{a}_{is} > 0 \right\} .$$

If $\bar{a}_{is} < 0$, then x_s can be increased without bound, to provide as small a value of z as is desired. In such cases, a minimum value of z does not exist.

Let us now consider the set of coefficient columns for which $\bar{c}_j < 0$ and for which there is at least one $\bar{a}_{ij} > 0$. If this set has more than one member, it is necessary to devise a column selection rule. Ideally, the column selected should afford the greatest reduction in z. To determine that column, a computation on each candidate would have to be made. Instead, the following a priori rule, successfully proven in practice, will be used.

Column Selection Rule: Choose the s'th column such that

$$\bar{c}_s = \operatorname*{Min}_j \bar{c}_j < 0 .$$

If, in the column s, we find more than one $\bar{a}_{is}$ such that $\bar{a}_{is} > 0$, then it is necessary to have on hand the following row selection rule.

Row Selection Rule: Select that row r for which

$$\theta = \operatorname*{Min}_i \frac{\bar{b}_i}{\bar{a}_{is}} \ : \ \bar{a}_{is} > 0$$

θ is always positive.

Next set $\theta = x_s$ and adjust the remaining basic variables to maintain equality. This is accomplished by a pivot operation* after which we obtain a new Canonical Form as follows:

*Note: The cost row is never used as a pivot row.

$$
\begin{array}{|l|l|l|}
\hline
x \dots \bar{a}_{ir} \; x_r \dots 0 & + \bar{a}_{1,m+1} \; x_{m+1} + \dots + 0 + \dots + \bar{a}_{1,n} \; x_n & = \bar{b}_1 \\
\vdots & \vdots & \vdots \\
0 \dots \bar{a}_{rr} \; x_r \dots 0 & + \bar{a}_{r,m+1} \; x_{m+1} + \dots + x_s + \dots + \bar{a}_{rn} \; x_n & = \bar{b}_r \\
\vdots & \vdots & \vdots \\
0 \dots \bar{a}_{mr} \; x_r \dots x_m & + \bar{a}_{m,m+1} \; x_{m+1} + \dots + 0 + \dots + \bar{a}_{m,n} \; x_n & = \bar{b}_m \\
\hline
0 \dots \bar{c}_r \; x_r \dots 0 & + \bar{c}_{m+1} \; x_{m+1} + \dots + 0 + \dots + \bar{c}_n \; x_n & = \bar{z} \\
\hline
\end{array}
$$

where

$$\bar{a}_{ij} = a_{ij} - \frac{a_{is} \, a_{ij}}{a_{rs}} \qquad i \neq r$$

$$\bar{a}_{rj} = \frac{a_{ij}}{a_{rs}} \qquad i = r$$

$$\bar{b}_i = b_i - \frac{a_{is} \, b_r}{a_{rs}} \qquad i \neq r$$

$$\bar{b}_r = \frac{b_r}{a_{rs}} = \theta \qquad i = r$$

$$\bar{c}_j = c_j - \frac{c_s \, a_{rj}}{a_{rs}}$$

$$\bar{c}_s = 0$$

$$\bar{z} = z + \frac{b_r \, c_s}{a_{rs}} < z \quad \text{(because } c_s < 0\text{)}$$

Note that $\bar{z} < z$ only if $b_r > 0$. If $b_r = 0$ (it cannot be less than zero by previously developed rules), we have a case of degeneracy. This situation will not occur as long as there are no ties in the row selection rule given above.

Barring degeneracy, the procedure described above is repeated until all $\bar{c}_j \geq 0$. At each repetition, a different and lower value of z is obtained, and a different b. f. s. is generated. For a system of m equations in n unknowns there are C^n_m basic feasible solutions possible. Since there are only a finite number of them and, during the procedure, none can be repeated (the value of z being different at each repetition), the number of iteration steps is finite. Hence, we will eventually find the optimal basic feasible solution.

Two facts are worthy of brief note here. The ill effects of degeneracy, while important in the theory, do not occur in practice unless the problem is carefully and artificially constructed to do so. The theoretical and computational problems which could conceivably arise can be readily handled, however. Secondly, the number C^n_m may be very large. Nevertheless, the simplex procedure homes-in on the optimal solution in much fewer iterations than might be expected from contemplation of this number. Further discussion of these two questions (among the many others which are also omitted from this exposition) can lead us too far afield. The reader is referred instead to Dantzig.[6] An excellent and more complete exposition of the simplex method and its many ramifications is contained there.

2.3 Matrix Operations

A matrix formulation of the arithmetic operations described in the preceding sections is useful in the remainder of this work.

Assume that the problem constraints are in Standard Form: $Ax = b$ where A is an $(m \times n)$ matrix with $m \leq n$ and vector x and b conform. We require $z = f(x) = cx$ to be minimized where c is an n-dimensional row vector of cost coefficients.

Let the rank of A be m, and select m linearily independent columns of A. Grouping these into a matrix B, we can write the Standard Form constraints as

$$\left[B \;\vdots\; R_{NB} \right] \left\{ \begin{array}{c} x_B \\ \hdashline x_R \end{array} \right\} = \left\{ b \right\} \tag{2.9}$$

Vector x_B contains the basic variables, and x_R the non-basic. The first m rows of the Canonical Form Equation (2.7) may therefore be expressed as

$$\left[I \;\vdots\; B^{-1} R_{NB} \right] \left\{ \begin{array}{c} x_B \\ \hdashline x_R \end{array} \right\} = \left\{ B^{-1} b \right\} \tag{2.10}$$

so that

$$B^{-1} R_{NB} = \begin{bmatrix} \bar{a}_{1,m+1} & \text{-------} & \bar{a}_{1,m} \\ & & \cdot \\ \bar{a}_{m,m+1} & \text{-------} \cdot & \bar{a}_{m,n} \end{bmatrix} \quad \text{and } B^{-1} b \quad \begin{Bmatrix} \bar{b}_1 \\ \\ \bar{b}_m \end{Bmatrix}$$

It remains now to determine the representation of $\bar{c}_j$ of Equation (2.7) in matrix terms. To begin, partition the cost functional $z = f(x) = cx$ so that

$$f(x) = c_B\, x_B + c_R\, x_R \tag{2.11}$$

where $c = \left[c_B \;\vdots\; c_R \right]$. Recall that the cost row $z = cx$ is never used as a source for a pivot in the simplex algorithm, but that every point operation had an effect on the transformation of the coefficients c_j into the $\bar{c}_j$ of Equation (2.7).

Let A_j be the j^th column of the original A matrix. Then

$$B^{-1} A_j = \begin{pmatrix} \bar{a}_{1j} \\ \bar{a}_{2j} \\ \bar{a}_{mj} \end{pmatrix}$$

where the $\bar{a}_{ij}$, $i = 1, 2, .., m$ are the same as those appearing in Equation (2.7). Define a scalar $\zeta_j = c_B B^{-1} A_j = \sum_{i=1}^{m} c_{Bi} \bar{a}_{ij}$.

Clearly ζ_j is the dot product of c_B and the Canonical Form's j^{th} column. The c_{Bi}'s, in turn, are the cost coefficients associated with whichever x_j's are basic at a given stage in the algorithm. In some L. P. texts, the column selection rule is based on the differences $(\zeta_j - c_j)$, $j = 1, 2, \ldots, n$. That is, if $(\zeta_j - c_j) > 0$, x_j is a candidate for the optimal basic solution. This fact is equivalent to the following statement

$$\bar{c}_j = c_j - \left[c_B B^{-1} \right] A_j = -\left(\zeta_j - c_j\right) \qquad (2.13)$$

where A_j is the j^{th} column vector of the original A matrix and $\bar{c}_j$ is the j^{th} coefficient of the cost row in Equation (2.7).

Before leaving this Section define the vector $c_B B^{-1} = \pi$. It can be shown that if x^*_B is the optimal basic solution with coefficients c^*_B, then

$$\pi^* = c^*_B B^{-1}$$

and π^* is the optimal solution to a related problem called the dual. The dual problem is described in the next section.

2.4 Duality

This section is intended to define only the essential properties of duality in Linear Programming. A complete exposition may be found in Reference 6, Chapter VI. For every LP problem in the unknown n-vector x (call it the primal problem), there exists another in the m-vector π (called the dual problem). Consider the following Standard Form as the primal problem.

(a) Find $x = (x_1, x_2, \ldots, x_n)$

(b) To minimize $z = \sum_{j=1}^{n} c_j x_j$ such that (2.14)

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m$$

and $x_j \geq 0$ for all j

Then the dual to this problem is written:

(a) Find $\pi = (\pi_1, \pi_2, \ldots, \pi_m)$

(b) To maximize $v = \sum_{i=1}^{m} \pi_i b_i$ such that (2.15)

$$\sum_{i=1}^{m} \pi_i a_{ij} \leq c_j$$

and π_i is unrestricted in sign

The relationship of the two problems can be represented by means of the following mnemonic called the "Tucker Diagram."

$$\begin{array}{c|cccc|c}
 & x_1 & x_2 & \cdots\cdots\cdots & x_n & \\
\hline
\pi_1 & a_{11} & a_{12} & \cdots\cdots\cdots & a_{1n} & b_1 \\
\pi_2 & a_{21} & a_{22} & \cdots\cdots\cdots & a_{2n} & b_2 \\
 & & & & & \\
\pi_m & a_{m1} & a_{m2} & \cdots\cdots\cdots & a_{mn} & b_m \\
\hline
 & c_1 & c_2 & \cdots\cdots\cdots & c_n &
\end{array}$$

The A matrix coefficients are presented as a detached array, so that the dot product of the x vector at the top outer row with every row of a_{ij}'s is related (in our case by equality) to the outer column of b_i's. The left outer column π, and the columns of a_{ij}'s are similarly related to the lower outer row of c_j's. From this diagram, the two problem statements may be readily visualized and stated as in Equations (2. 14 and 2. 15). The important relationship between the primal and dual problems is given by the following theorems proved in Reference 6.

Theorem 1: Duality Theorem

If feasible solutions to both the primal and dual exist, there exists an optimum solution to both systems and Min z = Max v.

Theorem 2: Unboundedness Theorem

(a) If a feasible solution to the primal system exists, but not to the dual, Min z approaches $-\infty$.

(b) If a feasible solution to the dual system exists, but not to the primal, Max v approaches $+\infty$.

Theorem 3: Optimality Conditions

If x is a feasible solution to the primal and π is a feasible solution to the dual, satisfying for $j = 1, 2, \ldots, n$,

$$\bar{c}_j = c_j - \sum_{i=1}^{m} \pi_i a_{ij} \geqq 0 \qquad (2.16)$$

then a necessary and sufficient condition for optimality of both solutions is

$$\bar{c}_j = 0 \quad \text{for} \quad x_j > 0 \qquad (2.17)$$

The following observations are applicable to Theorem 3:

(a) Equation (2.16) is identical to Equation (2.13). It may also be written as dual problem constraints

$$\sum_{i=1}^{m} \pi_i a_{ij} \leq c_j \qquad j = 1, 2, \ldots, m$$

(b) Theorem 3 may be combined into one statement

$$x_j \left[c_j - \sum_{i=1}^{m} \pi_i a_{ij} \right] = 0 \qquad (2.18)$$

(c) The Canonical Form Equation (2.7) can be used to infer the conditions of Theorem 3.

In the next Section, we will sketch the Kuhn-Tucker Conditions[7] for a linear cost function and linear equality constraints. They are identical with those of Theorem 3.

The π vector defined in this section will assume more significance in what follows. Its components, the variables π_i, $i = 1, 2, \ldots, m$ are known variously as prices, imputed values, and Lagrange multipliers. Interpretation as the adjoint or costate variables of optimal control theory will be developed in the chapters to follow. This interpretation may be introduced by an additional observation. Let z^* be the minimum cost. Then from Theorem 1,

$$z^* = \sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} \pi_i^* b_i = v^* \tag{2.19}$$

whereby

$$\frac{\partial z^*}{\partial b_i} = \pi_i^* \quad \text{for } i = 1, 2, \ldots, m \tag{2.20}$$

The b_i are the right-hand sides of the constraint equations; they determine the value of the constraint. Thus, the π_i^* represent the change in optimal value for a change in constraint.

2.5 Kuhn-Tucker Conditions

The LP problem considered in the preceding sections of this chapter belongs to the class of constrained minimization problems which may be treated very generally using Kuhn-Tucker analysis.[7] To avoid going too far afield in illustrating this connection, we will generalize the previous LP problem Equation (2.6) only to the point of allowing f(x) to be nonlinear. It is worth considering at least this much of a generalization to expedite the analysis; and to link it with Kuhn-Tucker conditions derived elsewhere.[7]

If we could assume that the optimum x were an exterior point to the set of feasible solutions we would formulate the Lagrangian function

$$L(x,\pi) = f(x) + \sum_{i=1}^{m} \pi_i \left[b_i - r_i(x) \right] \tag{2.21}$$

where $r_i(x) = \sum_{j=1}^{n} a_{ij} x_j$, $i = 1, =, \ldots, m$. In this case classical necessary conditions for x^* to be the x which locally minimizes $f(x)$ are

$$\frac{\partial L}{\partial x_j} = 0 \quad \frac{\partial L}{\partial \pi} = 0 \tag{2.22}$$

In mathematical programming problems, however, the analysis must consider cases where the optimum x is a boundary point of the set of feasible solutions. This is always the case with linear cost functions. Thus, the analysis becomes more complicated. We are required to replace the simple set of necessary conditions given in Equation (2.22) with Kuhn-Tucker conditions. Fortunately, in the case of linear programming, the increased analytic complexity is more than offset by the resulting computational simplicity. In LP, computational efficiency is obtained by taking advantage of the prior knowledge that the optimal x will, not only be on the boundary, but will be an extreme point of the set of feasible solutions.

The Kuhn-Tucker optimality conditions for a nonlinear cost function with linear equality constraints are as follows:

> If a certain $x = x^*$ minimizes an objective function $f(x)$ subject to linear equality constraints, then there exists a vector π^* unrestricted in sign such that, for all values of j, $j = 1, 2, \ldots, n$

$$\text{(a)} \quad x_j^* = 0 \ \underline{\text{and}} \ \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \pi_i^* a_{ij} \geq 0 \tag{2.23}$$

<u>or</u>

$$\text{(b)} \quad x_j^* > 0 \ \underline{\text{and}} \ \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \pi_i^* a_{ij} = 0 \tag{2.24}$$

We may combine (a) and (b) to arrive at

$$x_j^* \left[\frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \pi_i^* a_{ij}\right] = 0 \qquad (2.25)$$

Next observe that, since in LP $f(x) = \sum_{j=1}^{n} c_j x_j$, $\frac{\partial f}{\partial x_j} = c_j$. Thus Equation (2.18) from the last section and Equation (2.25) above are equivalent conditions.

Let us now consider Equation (2.25) with $\frac{\partial f}{\partial x_j} = c_j$. Summing over j we obtain

$$\sum_{j=1}^{n} c_j x_j^* - \left[\sum_{i=1}^{m} \pi_i^* \sum_{j=1}^{n} x_j a_{ij}\right] = 0 \qquad (2.26)$$

Since $\sum_{j=1}^{n} a_{ij} x_j = b_i$ we obtain the Duality Theorem (Theorem 1 of Section 2.4).

$$\text{Min } f(x) = \sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} \pi_i^* b_i = \text{Max } v(\pi) \qquad (2.27)$$

An additional observation is in order before leaving this section. The m-vector π used in the statement of the Kuhn Tucker conditions, (a) and (b) above, is the same vector used in the Lagrangian function

$$L(x, \pi) = f(x) + \sum_{i=1}^{m} \pi_i \left[b_i - r_i(x)\right]$$

The point at which $f(x)$ takes on a local minimum is a saddle point of the Lagrangian $L(x, \pi)$. $L(x, \pi)$ is said to have a saddle point at $\left[x^*, \pi^*\right] \epsilon E_{m+n}$ which minimizes $f(x)$ if

$$L(x^*, \pi) \leqq L(x^*, \pi^*) \leq L(x, \pi^*) \qquad (2.28)$$

Our original minimization problem now has a "dual" aspect. Given π^* we must minimize $L(x, \pi^*)$ over acceptable values of x. On the other hand, to provide for constraints the dual problem of maximizing $L(x^* \pi)$ over acceptable values of π enters the picture.

2.6 Generalized Programming

In Section 2.2 were described methods for obtaining solutions to LP problems for which the columns of the $(m \times n)$ matrix A were constants and their number, n, predetermined. Section 2.2 demonstrated that a "Column Selection Rule" could be determined which, along with other rules and operations, would lead to an optimal solution in a finite number of steps. This Section deals with a programming problem formulation in which we wish to determine, not only the solution vector x, but also column vectors (analogous to the columns of the constant matrix, A) which, when added during the course of the algorithm, will yield a minimum $f(x)$.

In this new formulation, columns are allowed to be variable in both form and number. Such a programming problem bears the name "Generalized Programming." It was developed by Philip Wolfe in the course of his work with George Dantzig on decomposition algorithms. (Reference 6, Chapter 22).

Let us call the finite dimensional Euclidean space E_n, solution space. The solution vector, x^* of a programming problem is a member of E_n. Two functions may be defined

$$f: E_n \rightarrow E_1 \qquad \left(\text{e.g.,}\ f(x) = \sum_{i=1}^{n} c_i\, x_i \right)$$

$$r: E_n \rightarrow E_m \qquad \left(\text{e.g.,}\ r(x) = Ax = b \ \text{ and } \ x \geq 0 \right)$$

The m-dimensional space E_m is called the requirement space for reasons which will become apparent later. We combine the two mappings f and r into one defined as follows:

$$F: E_n \rightarrow E_{m+1}$$

where $F(x) = [r(x), f(x)]$. The space E_{m+1} will be termed the extended requirement space. It consists of E_m space extended to include an (m + 1)st coordinate representing the cost function f(x).

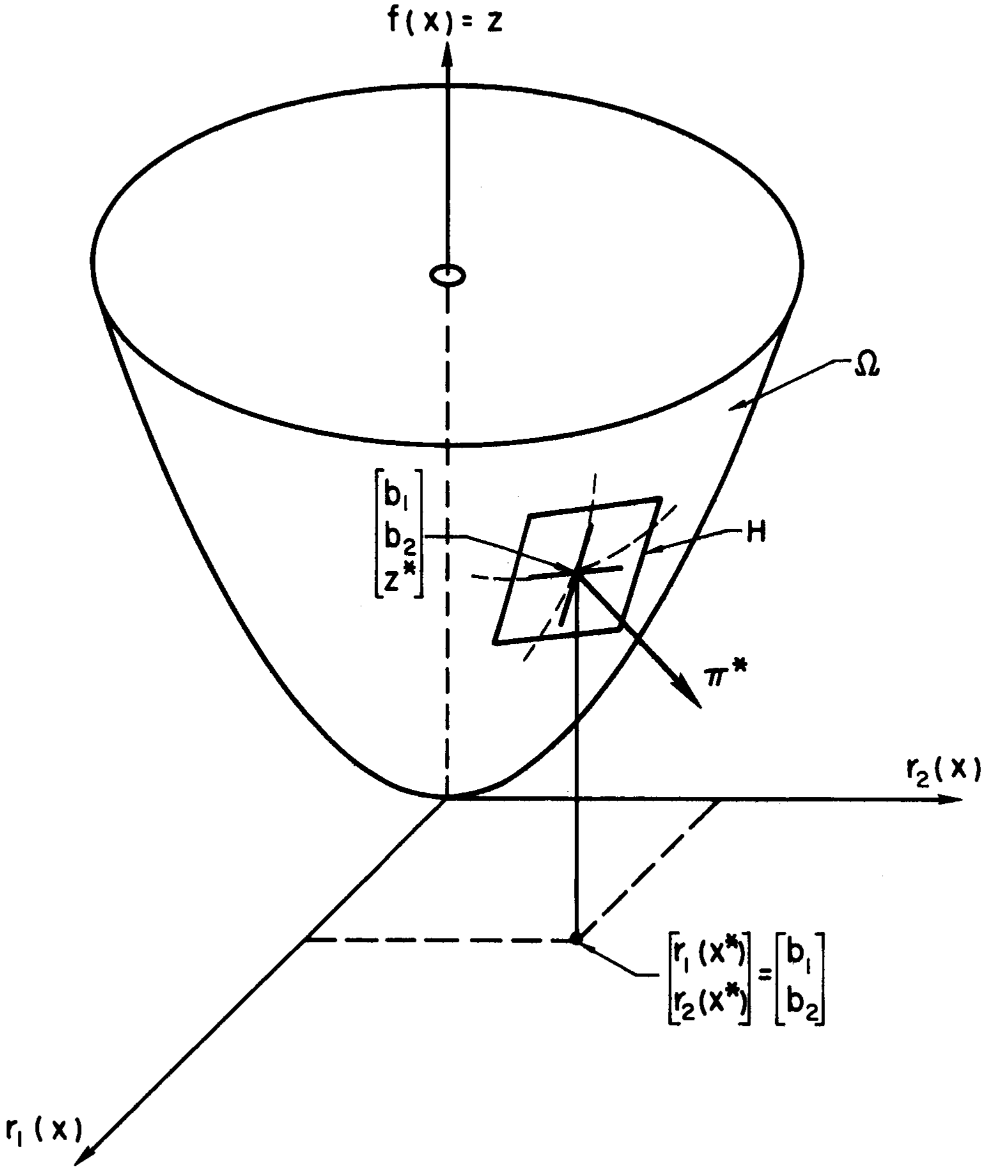
We will assume that the vector function r(x) induces on E_n a set $S \epsilon E_n$, - an infinite collection of n vectors x termed feasible solutions. These $x \epsilon S$ are feasible because they meet requirements of the problem expressed by r(x). In linear programming, they are the $x \geq 0$ which satisfy $Ax = b$. The m-vector b is often referred to in LP literature as the requirements vector.

We next assume that the feasible solutions, x, generate a convex set $\Omega \epsilon E_{m+1}$. Let us define an (m+1) column vector $P(x) \epsilon \Omega$ where

$$P(x) = \begin{bmatrix} r_1(x) \\ r_2(x) \\ | \\ | \\ | \\ r_m(x) \\ f(x) \end{bmatrix}$$

and, a forteriori, $x \epsilon S$. The set Ω for m + 1 = 3 is depicted in Figure 2-1.

The generalized programming problem may now be stated as follows:


f(x) = z
Ω
H
$\begin{bmatrix} b_1 \\ b_2 \\ z^* \end{bmatrix}$
π*
r₂(x)
r₁(x)
$\begin{bmatrix} r_1(x^*) \\ r_2(x^*) \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$

EXTENDED REQUIREMENTS SPACE CONTAINING Ω

FIGURE 2-1

(a) Given an objective function

$$z = f(x)$$

and the m constraint relations

$$r(x) = b$$

(b) Find that n vector x which minimizes z.

If we define

$$Q = \begin{bmatrix} b_1 \\ | \\ | \\ b_m \\ \hdashline 0 \end{bmatrix} \qquad U = \begin{bmatrix} 0 \\ 0 \\ | \\ | \\ 0 \\ \hdashline -1 \end{bmatrix}$$

we can rephrase the problem as one of finding x^* which minimizes z such that

$$zU + P(x) = Q \tag{2.29}$$

To convert this problem into computational form, we shall employ a "columnar procedure" described by Wolfe.[8] This consists of approximating the given (possibly nonlinear) mathematical programming problem Equation (2. 29) by a generalized linear programming problem amenable to solution by simplex methods.

A generalized linear programming formulation is characterized by its ability to accept new columns, and reject old ones, in such a manner that the linear approximation is steadily refined without an inordinate increase in the problem' s "A" matrix size. A description of generalized linear programming along with an interpretation extendable to optimal control computations follows.

Choose at least $t \geq m + 1$ grid points $x^j \epsilon S$, $j = 1, 2, \ldots, t$. If these are suitably chosen, we may write the desired solution vector x as a linear combination of the points x^j

$$x = \sum_{j=1}^{t} \lambda_j x^j , \quad \lambda^j \geq 0 \tag{2.30}$$

where $\sum_{i=1}^{t} \lambda_j = 1$. The t points x^j define corresponding vectors. $P(x^j) \in E_{m+1}$ such that the vector, $P(x) \in \Omega$, can be expressed as being in the convex hull of the $t \geq m + 1$ points $P(x^j)$, $j = 1, 2, \ldots, t$; that is,

$$P(x) = \sum_{j=1}^{t} \lambda_j \, P(x^j), \quad \lambda^j \geq 0 \tag{2.31}$$

where again $\sum_{i=1}^{t} \lambda_j = 1$. We are now able to approximate Equation (2.29) by an LP problem of the following form. By substituting Equation (2.31) into Equation (2.29) and adjoining the "coordination constraint" Equation (2.33) obtain

$$zU + \sum_{j=1}^{t} \lambda_j \, P^j = Q \tag{2.32}$$

$$\sum_{j=1}^{t} \lambda_j = 1 \tag{2.33}$$

where $P^j = P(x^j)$ and the $\lambda_j \geq 0$ are to be found so that z is minimized. The <u>Standard Form</u> for this problem, and its dual, may be diagrammed as in Section 2.4.

$$\begin{array}{c|ccccc|c}
 & \lambda_1 & \lambda_2 & \cdots\cdots\cdots & \lambda_t & (-z) & \\ \hline
(-v) & 1 & 1 & \cdots\cdots\cdots & 1 & 0 & 1 \\
\pi_1 & p_{11} & p_{12} & \cdots\cdots\cdots & p_{1,t} & 0 & b_1 \\
| & | & | & & | & | & | \\
\pi_m & p_{m,1} & p_{m,2} & \cdots\cdots\cdots & p_{m,t} & 0 & b_m \\
\pi_{m+1} & p_{m+1,1} & p_{m+1,2} & \cdots\cdots\cdots & p_{m+1,t} & 1 & 0 \\ \hline
 & 0 & 0 & \cdots\cdots\cdots & 0 & -1 &
\end{array}$$

Using this diagram, we may write:

(a) The Primal Problem: Find $\lambda_j \geq 0$ to minimize z such that

$$
\begin{aligned}
\lambda_1 + \quad \lambda_2 + \ldots\ldots + \quad \lambda_t &= 1 \\
p_{11}\lambda_1 + \quad p_{12}\lambda_2 + \ldots\ldots + \quad p_{1,\,t}\lambda_t &= b_1 \\
\\
p_{m1}\lambda_1 + \quad p_{m,\,2}\lambda_2 + \ldots\ldots + \quad p_{m,\,t}\lambda_t &= b_m \\
p_{m+1,\,1}\lambda_1 + p_{m+1,\,2}\lambda_2 + \ldots\ldots + p_{m+1,\,t}\lambda_t &= z
\end{aligned}
\tag{2.34}
$$

(b) The Dual Problem: Find π_i, unrestricted in sign, to maximize $\left[-v + \pi_1 b_1 + \pi_2 b_2 + \cdots + \pi_m b_m\right]$ such that

$$
\begin{aligned}
(-v) + p_{11}\pi_1 + \ldots\ldots\ldots + p_{m,\,1}\pi_m + p_{m+1,\,1}\pi_{m+1} &\leq 0 \\
(-v) + p_{12}\pi_1 + \ldots\ldots\ldots + p_{m,\,2}\pi_m + p_{m+1,\,2}\pi_{m+1} &\leq 0 \\
\\
(-v) + p_{1t}\pi_1 + \ldots\ldots\ldots + p_{m,\,t}\pi_m + p_{m+1,\,t}\pi_{m+1} &\leq 0 \\
\pi_{m+1} &\leq -1
\end{aligned}
\tag{2.35}
$$

The dual problem constraints stated in vector-matrix notation are

$$\pi[P] - \underline{v} \leq 0 \tag{2.36}$$

where $\underline{v} = [v, v, \ldots, v]$, an (m+1) row vector, and π is an (m+1) row vector with its (m+1)st component, $\pi_{m+1} \leq -1$. [P] is the following (m+1) x (t) matrix

$$[P] = \begin{bmatrix} p_{11} & p_{12} \cdot \cdots\cdots & p_{1,t} \\ p_{12} & p_{22} \cdot \cdots\cdots & p_{2,t} \\ \\ p_{m+1,1} & \cdots\cdots & p_{m+1,t} \end{bmatrix}$$

The geometrical interpretation of Equation (2.36) will be devloped further in Chapter III. For now, let us postmultiply Equation (2.36) by the column vector $\lambda^* = \left[\lambda_1, \lambda_2, \ldots, \lambda_t\right]$ corresponding to the optimal solution $x^* = \sum \lambda_j^* x^j$. From this operation, and recalling that $\sum_{j=1}^{t} \lambda_j = 1$, we obtain

$$\pi\ P(x^*) - v \leq 0 \quad ^{\dagger} \tag{2.37}$$

If now $\pi = \pi^*$ is the best value of π, we can write

$$\pi^*\ P(x^*) - v = 0 \tag{2.38}$$

which is the equation of a hyperplane H tangent to the set Ω at the point in (m+1) space (b_1, b_2, z^*). Figure 2-1 depicts this hyperplane with its defining vector π^*.

The computational problem associated with Equation (2.34) would become difficult even for the simplex method if there were not some method of limiting the number of columns $P^j = P(x^j)$ as the number of grid points required to search for, and to refine a solution is increased.

Recall now an important property of the simplex solution of Equation (2.34) consisting of m + 1 constraint equations and an

† $P(x) = \sum_{j=1}^{t} \lambda_j p^j \Rightarrow P(x^*)$ as $t \rightarrow \infty$

objective function z. That is, there will be at most only m + 1 non-zero λ^j's in the optimal solution of the approximating LP problem. As we refine the solution by adding more grid points, and hence more columns P^j, we could well drop some of the columns associated with those λ^j's which equal zero. The procedure of choosing which columns to drop, and the method of generating new columns to add, is the essence of the decomposition algorithm[6, 8] which shall be described in the next section.

2.7 Decomposition

Briefly, the decomposition procedure consists of formulating a master problem, or "coordinating set" of equations, such as Equation (2.34). The master problem produces, at each iteration, a "price" vector (dual variable vector), π. The "price" vector π is used to formulate one or more smaller sub-problems which are solved to produce new columns (sometimes called activity vectors) for the master problem. Under appropriate convexity assumptions, this process will converge to a solution.

Consider Equation (2.34) as a master problem with $t \geq m + 1$ linearly independent column (activity) vectors. Assume that by application of a Phase I and Phase II procedure we have found a solution $\bar{\lambda}_1, \bar{\lambda}_2, \ldots \bar{\lambda}_t$ and the corresponding (- v), $\bar{\pi}_1, \bar{\pi}_2, \ldots \bar{\pi}_{m+1}$[†]. As in Wolfe,[8] consider the question: "Of all possible points x^{t+1} that might be adjoined to the master problem to refine the approximating grid, which point would the simplex method choose as contributing most to the solution of the original

[†] Note: The bar over the λ and π in this section indicates that there are new values of λ and π obtained after an iteration of the algorithm.

problem ?" The answer lies in the Column Selection Rule of Section 2.2. For convenience, we repeat this rule here

<u>Column Selection Rule:</u> Choose the s' th column of the Canonical Form so that

$$\bar{c}_s = \operatorname*{Min}_j \bar{c}_j < 0 \qquad (2.39)$$

But we have seen in Equation (2.13) and Equation (2.16) that

$$\bar{c}_s = c_s - \sum_{i=1}^{m} \pi_i a_{ij}$$

Hence the column selection rule tells us to pick that x_{t+1} which will create a column such that

$$\bar{c}_{t+1} = \operatorname*{Min}_j \left[c_j - \sum \pi_i a_{ij} \right] \qquad (2.40)$$

or equivalently, such that

$$\bar{c}_{t+1} = \operatorname*{Max}_j \left[- c_j + \sum \pi_i a_{ij} \right] \qquad (2.41)$$

To link our analysis thus far to the generalized programming problem originally stated at the beginning of Section 2.6, we may make the following substitutions:

$$\begin{aligned} p_{oj} &= a_{oj} = 1 \\ p_{ij} &= a_{ij} = r_i(x^j) \qquad i = 1, 2, \ldots, t \\ p_{m+1,j} &= c_j = f(x^j) \end{aligned} \qquad (2.42)$$

The $(t+1)^{th}$ column to be added is thus obtained by solving the following sub-problem which depends on the $\bar{\pi}_1, \bar{\pi}_2, \ldots \bar{\pi}_{m+1}$ previously determined from the master problem:

$$\underset{x \in S}{\text{Maximize}} \left[\bar{\pi}_{m+1} f(x) + \sum_{i=1}^{m} \bar{\pi}_i r_i(x) \right] \qquad (2.43)$$

Since $\pi_{m+1} = -1$, this statement is identical to Equation (2.41)†. Any number of optimization methods may be employed to find the minimum of this subproblem. Note also that it need not be linear. Furthermore, it is frequently easier to solve than the original problem because the $\bar{\pi}_i$' s (Lagrange multipliers) are already specified.

After finding the x^{t+1} which maximizes Equation (2.43), we make the substitutions indicated in Equation (2.42) to determine the (t+1)' st column to be added to the master equation, that is

$$\begin{aligned} p_{o,t+1} &= a_{o,t+1} = 1 \\ p_{i,t+1} &= a_{i,t+1} = r_i(x^{t+1}) \\ p_{m+1,t+1} &= c_{t+1} \;\;= f(x^{t+1}) \end{aligned} \qquad (2.44)$$

The master equation is again solved with the new (t+1)st column included, and a new set of $\bar{\pi}_i$' s is determined. With this new set we again solve Equation (2.34) until the optimality criterion, $\bar{c}_j \geq 0$ for all j, is satisfied for all columns we are able to construct. Columns for which the λ_j remain zero for several iterations may be dropped, thus keeping the master problem to a tractable size. The procedure converges to a solution if the set Ω is convex.[6] It may possibly converge for more general problems, for example where convexity does not hold. Everett[9] presents his so-called "Main Theorem" which can be applied to the analysis sketched above. Everett' s "Main Theorem" alleges that the method is "fail-safe" in the sense that, if a solution can be found, it is a true solution. Everett' s Theorem employs a "generalized Lagrange multiplier"

† Note: We may always assume that π is normalized so that $\pi_{m+1} = -1$.

point of view which may be applied to a variety of problems (with discrete, nonlinear and nonanalytic properties). Brooks and Geoffrion[10] have elaborated on Everett' s work with "generalized Lagrange multipliers," and have linked them to the dual variables of linear and generalized programming.

One more interpretation of the decomposition procedure is worthy of note before leaving this section. As demonstrated above, the master program may be viewed as "coordinating" the (t+1)' st new column with the already existing columns. If each column is considered a member activity of a larger industrial combine, the subproblem, Equation (2.43), corresponds to the problem faced by the manager of that member activity. The only "instructions" given to the activity manager by the central authority (master program) are an objective, Equation (2.43), and a current price structure, the $\bar{\pi}_i$' s. This price structure is subject to change as a result of the coordination process taking place at "higher headquarters." This interpretation, due to Dantzig, will find increasing application in both micro and macro economics models, as well as in control theory computations.

In Chapter III, an analogy between the dual problem (Equation (2.35)) and various optimal control computation schemes will be developed. In particular, it will be shown how the Maximum Principle of Pontryagin can be derived under the duality aspects of generalized programming.

CHAPTER III

MINIMUM FUEL OPTIMAL CONTROL CONTINUOUS TIME CASE

3.0 Introduction

This chapter treats the continuous time minimum fuel control optimization process from the following points of view:

(a) Geometric (Neustadt, Reference 11)

(b) Maximum Principle (Pontryagin, Reference 12)

(c) Functional Analysis (Krasovski, Reference 13)

(d) Generalized Programming (Dantzig, Reference 14)

The relationship between these approaches will be demonstrated, along with their unity under the duality properties of mathematical programming.

We consider the following system of equations:

$$\dot{x} = A(t)x + B(t)u \qquad (3.1)$$

$$|u| \leq M$$

$$C(t) = \int_{t_o}^{t} |u(\tau)|\, d\tau$$

where x is an m-state vector, u is an r-vector available to the controller and C is a performance functional representing the control cost. Since we desire x(t) = 0 for some t = T, and since $\Phi(t)$ is a non-singular transition matrix, we write

$$-x(o) = \int_{t_o}^{t} \Phi(t_o, \tau)\, B(\tau)\, u(\tau)\, d\tau \qquad (3.2)$$

$$|u| \leq M$$

$$C(t) = \int_{t_o}^{t} |u(\tau)|\, d\tau$$

Our problem is to find $u(t)$ satisfying the conditions outlined above such that C, the amount of fuel used in control, is minimized. For clarity, let us consider the single input case, $r = 1$; also, allow A and B to be constant $m \times m$ and $m \times 1$ matrices respectively. The contents of this chapter apply equally well to the time varying multi-input case.

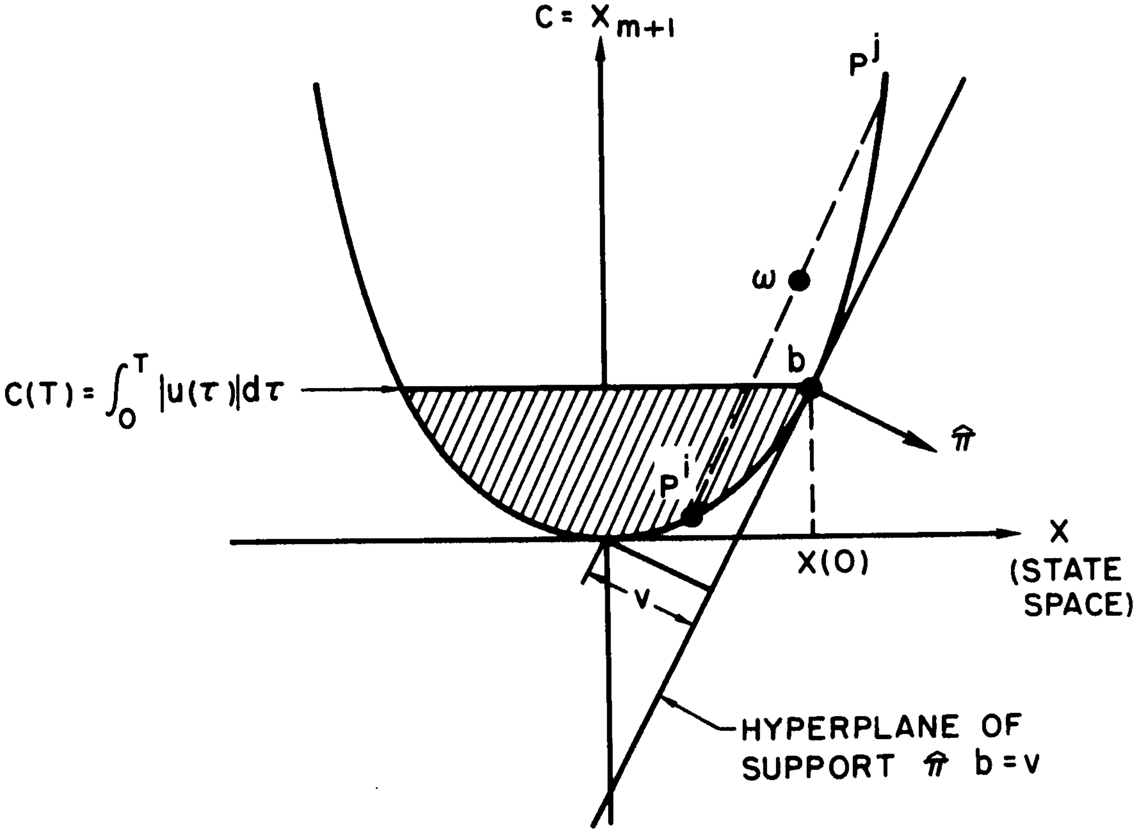
3.1 Geometric Approach

At a fixed terminal time, $t = T$, there is a set $\Omega(T)$ defined by

$$\Omega(T) = \left\{ - \int_o^T \Phi^{-1}(\tau) B\, u(\tau) d\tau,\ \int_o^T |u(\tau)| d\tau : |u(t)| \leq M \right\}$$

The set $\Omega(T)$ is closed, bounded and convex in $(m + 1)$ - dimensional Euclidean space. The $(m + 1)$st coordinate value represents the fuel cost, C. Since the allowable $u(t)$ are symmetric about the origin, and $x_{m+1} = \int_o^T |u(\tau)| d\tau$, it follows that the set $\Omega(T)$ is symmetric about the $(m+ 1)$ axis. $\Omega(T)$ is represented by the shaded area in Figure 3-1 where it is easy to see that, as T is allowed to increase, $C(T)$ increases. The properties of $\Omega(T)$ which make this intuitive viewpoint rigorous may be found in References 15 and 16. Meditch[16] shows that $T_2 > T_1$ implies $\Omega(T_2) \supseteq \Omega(T_1)$. Thus a larger range of initial conditions, $x(o)$, can be driven to zero as more time and/or possibly more fuel is allowed. This is rigorously demonstrated in Reference 16; it is also clear from Figure 3-1 and the physics of the situation.

For a given $t = T$ and initial value, $x(o)$, the minimum cost to reach the origin is the least value of $C = x_{m+1}$ for which the point $\left(x(o), x_{m+1}\right) \in \Omega(T)$. This is the boundary point b in Figure 3-1, for if point b were interior to the set $\Omega(T)$, the initial value $x(o)$ could be driven to zero at less cost.



THE CONVEX SET Ω(T)

FIGURE 3-1

Since b is a boundary point, and $\Omega(T)$ is convex, there exists a hyperplane of support to $\Omega(T)$ at the point (vector) b. This hyperplane has the vector equation equivalent to Equation (2. 38)

$$\hat{\pi} b = v$$

where $\hat{\pi}$ is an (m+1)-row vector, b an (m+1)-column vector. Note that $b = \left(x(o),\ x_{m+1}\right)$ so that x(o) has dimension m.

Meditch and Neustadt[17] develop the following relation:

$$\hat{\pi}\left(x(o),\ x_{m+1}\right) \geq \hat{\pi}\omega$$

for all points $\omega \in \Omega(T)$. This is apparent from the geometry of Figure 3-1. Next write

$$\hat{\pi}\left(x(o),\ x_{m+1}\right) \geq \hat{\pi}\,\omega = \int_o^T \hat{\pi}\left[\Phi^{-1}(\tau)Bu(\tau),\ |u(\tau)|\right] d\tau$$

If ω were to equal $\left(x(o),\ x_{m+1}\right)$ we could infer that for certain (as yet unknown) u(t) and $\hat{\pi}$.

$$\hat{\pi}\left(x(o),\ x_{m+1}\right) = \underset{\substack{u(t) \\ |u(t)| \leqq M}}{\text{Max}} \int_o^T \hat{\pi}\left[\Phi^{-1}(\tau)\ Bu(\tau), |u(\tau)|\right] d\tau \qquad (3.\ 3)$$

Another way of expressing the same equation is:

$$u^o(\hat{\pi}, t) = \arg \underset{\substack{u(t) \\ |u(t)| \leqq M}}{\text{Max}} \int_o^T \hat{\pi}\left[\Phi^{-1}(\tau)\ Bu(\tau),\ |u(\tau)|\right] d\tau$$

where $u^o(\hat{\pi}, t)$ designates the optimal control.

The vector $\hat{\pi}$ embodies the relationship between the plant's initial state x(o), the time T allowed to drive that state to zero, and the geometric shape of the convex set of recoverable states

$\Omega(T)$. In general, this relationship is known only qualitatively – the sum total of this knowledge being represented geometrically in Figure 3-1.

Before attempting to solve for $u^o(\hat{\pi}, t)$, allow the (m+1)st component of $\hat{\pi}$ to equal -1. This is permissible because the magnitude of $\hat{\pi}$ is arbitrary. Further motivation for this action can be derived from the representation of $\Omega(T)$ in Figure 3-1. In addition, this convention causes the maximization operation on the right-hand side of Equation (3. 3) to correspond to minimizing $C = \int_o^T |u(\tau)| d\tau$. Equation (3. 3) then becomes

$$\hat{\pi}\left(x(o),\ x_{m+1}\right) = \underset{\substack{u(t) \\ |u(T)| \leq M}}{\text{Max}} \int_o^T \left[q(\tau)\, u(\tau) - |u(\tau)|\right] d\tau \qquad (3.4)$$

where for convenience we set

$$q(\tau) = \pi \quad \Phi^{-1}(\tau)B$$

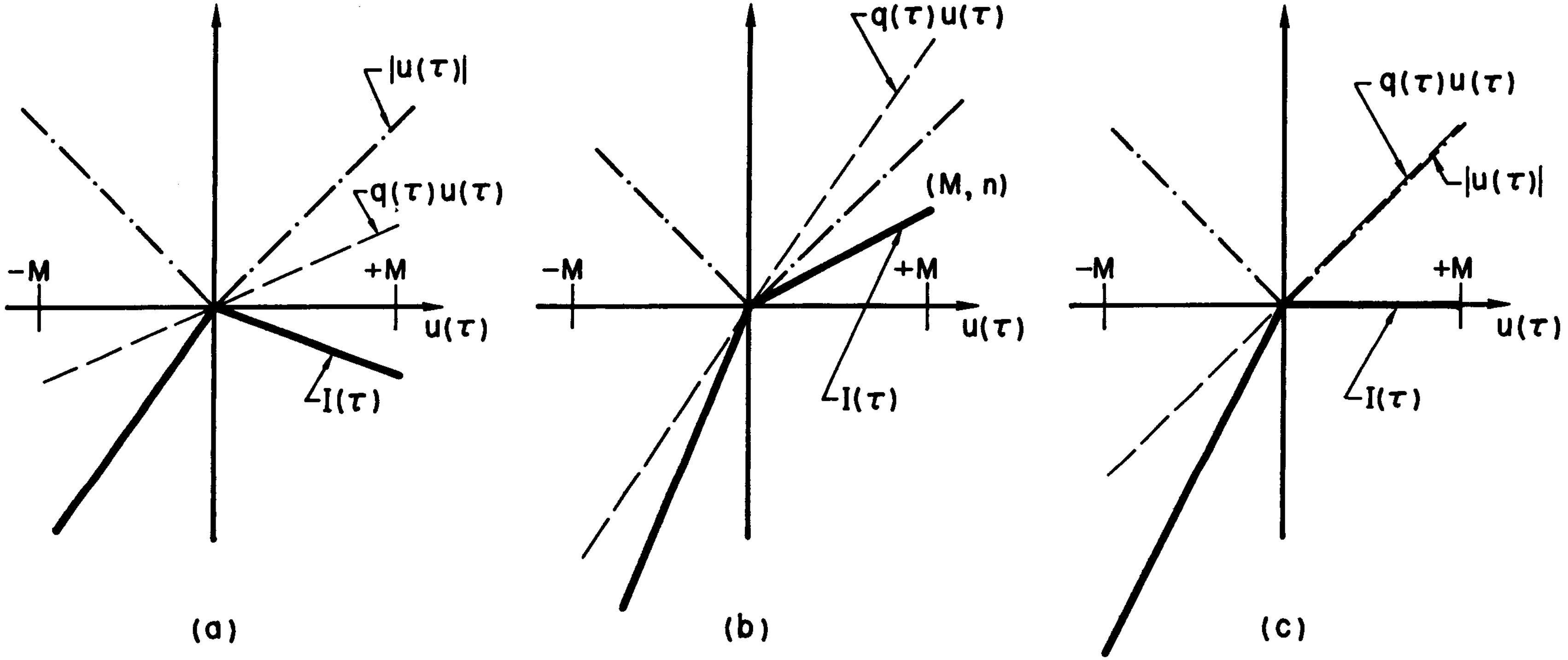
$$I(\tau) = q(\tau)\ u(\tau) - |u(\tau)|$$

π is an <u>m-vector</u>. The maximization operation of Equations (3. 3 and 3. 4) is carried out with the aid of Figure 3-2.

$$u^o(t, \pi) = \begin{cases} M \,\text{sgn}\left[\pi \quad \Phi^{-1}(t)B\right] & \text{for} \quad |\pi\ \Phi^{-1}(t)B| > M \\ 0 & \text{for} \quad |\pi\ \Phi^{-1}(t)B| < M \end{cases} \qquad (3.5)$$

Figure 2c shows that if $|\pi \quad \Phi^{-1}(t)B| = M$ identically for any measurable interval in the range $0 \leq t \leq T$, the control is undefined by the above derivation of necessary conditions. However, it is true that u(t) lies somewhere between - M and + M, that is

$$0 \leq |u(t)| \leq M$$



(a)
$0 < q(\tau) < M$
MAX $I(\tau) = 0$
AT $u(\tau) = 0$

(b)
$q(\tau) > M$
MAX $I(\tau) = n$
AT $u(\tau) = +n$

(c)
$q(\tau) = M$
MAX $I(\tau) = 0$
AT $0 \leq u(\tau) \leq n$

MAXIMIZATION OF $I(\tau) = \left[q(\tau)\, u(\tau) - |u(\tau)| \right]$

FIGURE 3-2

Thus, if $|\pi \Phi^{-1}(t)B| = M$ in a finite interval of t, we have a situation associated with singular solutions of the optimal control problem. Singular solutions of this type occur naturally in the minimum fuel control of a pure inertia with equations $\dot{x}_1 = x_2 = u$, $|u| \leq 1$. This is discussed by Meditch.[16] A more general discussion is contained in Leondes.[18]

Even in the case of non-singular control, the problem statement is far from complete. The optimum control $u^o(t, \hat{\pi})$ is expressed as a function of time and the (m+1)-vector, $\hat{\pi} = (\pi, -1)$. If we could find the vector π associated with the given initial conditions $x(o)$, we will have solved the "open loop" synthesis problem. If we can do this repeatedly in a time interval negligible with respect to the system' s (plant) dynamics, we can close the loop with a computer. Finally, if u^o could be found as a function of the plant' s state vector, we say that we have solved the "closed loop" synthesis problem.

Reference 17 develops a steepest ascent technique for finding $\hat{\pi} = (\pi, -1)$. An initial guess $\pi^{(o)}$ is first made and substituted into Equation (3. 5). This provides an initial control function $u\left(t, \pi^{(o)}\right) = G\left(t, \pi^{(o)}\right)$ which is used in Equation (3. 1) to yield a point, $x\left(T, \pi^{(o)}\right)$, on the boundary of $\Omega(T)$.

Thus

$$x\left(T, \pi^{(i)}\right) = \Phi(T)\left[x(o) + \int_o^T \Phi^{-1}(\tau)\, B(\tau)\, G\left(\tau, \pi^{(i)}\right) d\tau\right] \qquad (3.6)$$

The following recurrence relation can be developed from Equation (3. 6) and shown to converge to the optimal π for a convex $\Omega(T)$

$$\pi^{i+1} - \pi^{i} = - k\left[x(o) + \int_o^T \Phi^{-1}(\tau)\, B(\tau)\, G\left(\tau, \pi^{(i)}\right) d\tau\right] \qquad (3.7)$$

Reference 19 discusses a computational study of the steepest ascent method described above. The system selected for that study was third order with special advantageous properties. It was noted that when k was kept constant, the method failed to work. A variable gain (variable k) method was also tried but convergence was slow. Powell' s gradient acceleration method was reported to achieve good results.

3.2 Maximum Principle Approach

The Pontryagin Maximum Principle is applied extensively to derive necessary conditions for the optimal fuel problem. We form the Hamiltonian

$$H\left[x(t),\ \pi(t),\ u(t)\right] = \pi(t)\left[A\ x(t) + B\ u(t)\right] - |u(t)|$$

where $\pi(t)$ is a time varying m -vector, a solution to the adjoint equation

$$\dot{\pi}(t) = -\pi(t)A \qquad (3.8)$$

According to the maximum principle, if the optimal control $u^o(t)$ is applied to the system Equations (3. 1), there exists a $\pi^o(t)$ satisfying Equation (3. 8), and an optimal trajectory $x^o(t)$ such that the Hamiltonian Equation (3.7) is maximized. That is

$$H\left[x^o(t),\ \pi^o(t),\ u^o(t)\right] = \underset{\substack{u(t)\\ |u(t)| \leq M}}{\text{Max}}\ H\left[x(t),\ \pi(t),\ u(t)\right]$$

Equivalently,

$$\pi^o(t)B\ u^o(t) - |u^o(t)| = \underset{\substack{u(t)\\ |u(t)| \leq M}}{\text{Max}}\left[\pi(t)B\ u(t) - |u(t)|\right] \qquad (3.9)$$

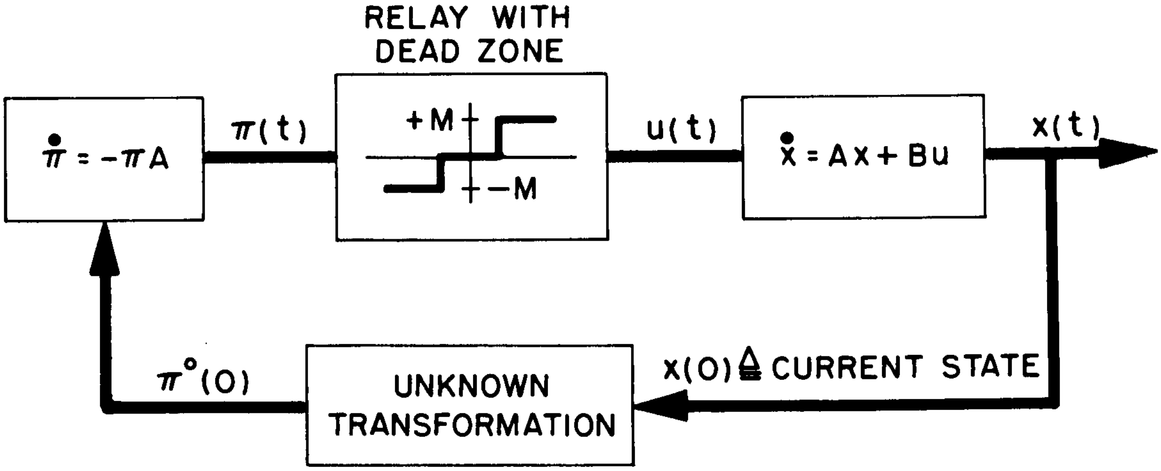
From this equation we may derive

$$u^{o}(t, \pi) = \left\{ \begin{array}{ll} M \operatorname{sgn}\left[\pi^{o}(t)B\right] & \text{for } |\pi^{o}(t)B| > M \\ 0 & \text{for } |\pi^{o}(t)B| < M \end{array} \right\} \qquad (3.10)$$

If $|\pi^{o}(t)B| = M$ for a non-zero interval of time, then we again have a singular control situation. Singular sub arcs of an optimal trajectory occur in cases where H is not explicitly a function of the control variable u(t) during a finite length of time. In this section such a case arises for $|\pi^{o}(t)B| = M$ identically.

Open loop synthesis of the optimal control (in the non-singular case) could be achieved if we knew $\pi^{o}(t)$.

The well known two point boundary value problem confronts us here. A way of depicting it (due to Athanassiades[20]) is as follows:



$\pi^o(t)$ is determined by an unknown transformation of the current state, $x(0)$. This relationship is expressed qualitatively in the sketch above. However, the problem of finding $\pi(t)$ quantitatively still remains. Thus we are faced with a stumbling block to synthesis comparable to that of the previous section.

This comparison is evident when we express the relationship between the Pontryagin Maximum Principle and Neustadt's geometrical approach. For the π of Section 3.1 and the $\pi(t)$ of this section $\pi(o)_{\text{Sec } 3.2} = \pi_{\text{Sec } 3.1}$. A simple Lemma will assist in making this more explicit.

<u>Lemma:</u> Let $\Phi(t)$ be the characteristic matrix of $\dot{x} = Ax$. Then $\left[\Phi^{-1}(t)\right]$ is the characteristic matrix of $\dot{\pi} = -\pi A$

<u>Proof:</u> By definition $\Phi(t)^{-1}\Phi(t) = I$

Differentiating obtain

$$\dot{\Phi}(t)^{-1}\Phi(t) + \Phi^{-1}(t)\dot{\Phi}(t) = 0$$

Then

$$\dot{\Phi}^{-1}(t) = -\Phi^{-1}(t)A$$

proving the Lemma.

Using this Lemma, we can write the solution to Equation (3.8) as

$$\pi(t) = \pi(o)\,\Phi^{-1}(t)$$

If $\pi(0) \triangleq \pi \triangleq \pi^o$ is the initial condition of the adjoint corresponding to the optimal solution satisfying the given initial conditions, then Equation (3.10) becomes

$$u^o(t,\pi) = \left\{\begin{array}{ll} M \operatorname{sgn}\left[\pi\ \Phi^{-1}(t)B\right] & \text{for } \left|\pi\ \Phi^{-1}(t)B\right| > M \\ & \\ o & \text{for } \left|\pi\ \Phi^{-1}(t)B\right| < M \end{array}\right\} \qquad (3.11)$$

which is identical to Equation (3.5).

This section has demonstrated that with certain conventions the control derived by application of the Pontryagin Maximum Principle and that derived by Neustadt' s method are identical. However, finding the vector $\pi_{\text{Sec. } 3.1} = \pi(o)_{\text{Sec. } 3.2}$ still remains at the core of the synthesis problem. For a rigorous derivation of the concepts summarized in this and the previous section, the reader is referred to Roxin.[21]

3.3 Functional Analysis Approach

The application of functional analysis to optimal control problems of various types is ably depicted by Kranc and Sarachik,[22] among others. This section treats only the highlights of the theory. Its purpose is to indicate how the dual vector π enters the theory underlying the functional analysis approach and how it relates to the previous two sections.

Equation (3.2) of the introduction states that[†]

$$- x(o) = \int_o^T \Phi^{-1}(\tau)B\, u(\tau)d\tau$$

or, by setting $\Phi^{-1}(\tau)B = h(\tau) = \left[h\ (\tau), \ldots, h_m(\tau)\right]^T$,

$$- x(o) = \int_o^T h(\tau)u(\tau)d\tau \tag{3.12}$$

We specialize our treatment to the single control input case in order to simplify notation. The results to follow are easily generalized.

Define an arbitrary m-vector π such that

$$- \pi \bullet x(o) = \int_o^T \pi \bullet h(\tau)\, u(\tau)d\tau \tag{3.13}$$

Next write

[†] Note: Constant coefficient system is assumed here without loss of generality.

$$\left| - \pi \bullet x(o) \right| = \left| \int_o^T \pi \bullet h(\pi)\, u(\tau) d\tau \right| \leq \int_o^T \left| \pi \bullet h(\tau) u(\tau) \right| d\tau$$

$$\leq \left\| \pi \bullet h(\tau) \right\|_q \left\| u \right\|_p \qquad (3.14)$$

where the second inequality is the Hölder Inequality and

$$\|u\|_p \triangleq \left[\int_o^T |u(\tau)|^p \, d\tau \right]^{1/p} \qquad (3.15)$$

$$\|\pi \bullet h(\tau)\|_q \triangleq \left[\int_o^T |\pi \bullet h(\tau)|^q \right]^{1/q} \qquad (3.16)$$

$$p^{-1} + q^{-1} = 1$$

Consider the right-hand side of Equation (3.12) as a linear functional defined over a function space $E = L_q(o, T)$ with the property that

$$f(h_i) = - x_i(o) = \int_o^T h_i(\tau) u(\tau) d\tau \quad i = 1, 2, \ldots, m$$

This functional is said to map E into a conjugate space E^* consisting of all functions u(t) with p norm defined by (3.15). It can be shown that $\|f\| = \|u\|_p$. Our problem is now one of finding a functional f of least norm having a finite number of given "moments," $- x_i(o)$, $i = 1, 2, \ldots, m$ (Neustadt Reference 23). The vector π such that $- \pi \bullet x(o) = 1$ will define a subspace. The norm found over this subspace equals the norm over the entire (infinite dimensional) space by the Hahn Banach Theorem. This problem was solved abstractly by Krein.[24] Krein's work gives a necessary and sufficient condition for the existence of f, i.e., f can be found if and only if

$$\min_{\pi} \|\pi \bullet h\| = M(T) \geq 1$$

such that

$$\pi \bullet x(o) = -1$$

If the norm $\|\pi \bullet h\|$ is properly chosen, e.g., as a q norm with $q = 1$, we have a linear programming formulation of the minimum fuel problem already. However, as $q = 1$, $p = \infty$ certain mathematical difficulties in Banach space occur. These have to do with u(t) approaching an impulse as p approaches ∞. This problem does not arise in the discrete time case and hence will not be further discussed here.

From (3.14) we can state

$$\|u\|_p \geq \frac{|-\pi \bullet x(o)|}{\|\pi \bullet h\|_q} \tag{3.17}$$

so that

$$\text{Min } \|u\|_p = \frac{|-\pi \bullet x(o)|}{\|\pi \bullet h\|_q} \tag{3.18}$$

for all π including the "worst case," that is when

$$\text{Min } \|u\|_p = \underset{\pi}{\text{Max}} \frac{|-\pi \bullet x(o)|}{\|\pi \bullet h\|_q} \tag{3.19}$$

By straightforward manipulation, this yields

$$\underset{\pi}{\text{Max}} \frac{|-\pi \bullet x(o)|}{\|\pi \bullet h\|_q} = \underset{\substack{\pi \\ \pi \bullet x(o) = -1}}{\text{Max}} \frac{1}{\|\pi \bullet h\|_q}$$

or

$$\text{Min } \|u\|_p = \frac{1}{\underset{\substack{\pi \\ \pi \bullet x(o) = -1}}{\text{Min}} \|\pi \bullet h\|_q} \tag{3.20}$$

Thus the determination of the minimum of $||u||_p$ reduces to finding that π which minimizes $||\pi \bullet h||_q$ under the constraint $\pi \bullet x(o) = 1$ (exactly as for Krein's "L Problem").

For Equation (3.19) to hold as an equality, the Hölder relation, Equation (3.14), must hold as an equality. Kranc and Sarachik[22] show that this occurs when

$$u(t) = \frac{|\pi \bullet h(\tau)|^{q-1}}{\left[||\pi \bullet h(\tau)||_q\right]^q} \quad \mathrm{sgn}\, [\pi \bullet h(\tau)] \qquad (3.21)$$

With some additional mathematical reasoning, it can be shown that when $q = 1$ and $p \to \infty$

$$||u||_\infty = \sup_{\tau \in (o, T)} |u(\tau)|$$

If $\sup_{\tau \in (o,T)} |u(\tau)| = M$, it is readily shown that $M = \left[||\pi \bullet h(\tau)||_1\right]^{-1}$ so that Equation (3.21) becomes

$$u(t, \pi) = M\, \mathrm{sgn}\left[\pi \bullet \Phi^{-1}(\tau)B\right]$$

This is the control which minimizes $||u||_p$ if π is chosen in accordance with Equation (3.20). When this control is used in Equation (3.12), the tradeoff between T and M may be easily seen. As M is allowed to increase, the time required to drive the plant to zero will decrease. The tradeoff curve is monotonic and continuous. It can be used to determine the Min $T = T_{Min}$ for a given control bound M. For this minimum time problem, the functional analysis approach is very effective.

Unfortunately the minimum fuel problem is not so easily handled. For a given $T > T_{Min}$, and a given M, there is freedom to choose trajectories which minimize C the amount of fuel used, where

$$C = \int_0^T |u(\tau)| d\tau = ||u||_1$$

With the Geometric and the Maximum Principle techniques we were able to identify "coast times," i.e., times when u(t) = 0. When the function space methods of this section are applied, however, one cannot bound the magnitude of the control when $p = 1$. The control law derived by setting $p = 1$ and $q = \infty$ represents C ideally, but prescribes impulses at those points, τ_i, where $\sup_{(o, T)} |\pi \cdot h(\tau)|$ is attained. Thus, the functional analysis approach to the minimum fuel problem with bounded control remains yet to be developed. Swiger[25] presents a numerical attack on this problem. Chapter V, concerned with the discrete time case, shows how linear programming can be employed.

The main purpose of this section is to indicate the commonality of the functional analysis approach with the other techniques of this chapter. This commonality centers around the "Lagrange Multiplier" properties of the dual vector π. These properties will be interpreted further in the chapters to follow.

3.4 Generalized Programming Approach

Dantzig[14] and Van Slyke[26] were the first to apply the methods of generalized programming and decomposition to arrive at the Maximum Principle. This section applies their results to the minimum fuel problem, and demonstrates the unity of these programming methods with the three previously described approaches to control optimization. The analysis employs the material of Sections 2.6 and 2.7.

Consider Equation (3.2) and the convex set $\Omega(T)$ depicted in Figure 3-1

$$- x(o) = \int_o^T \Phi^{-1}(\tau)B\,u(\tau)d\tau \qquad (3.22)$$

Letting $C(T) = \int_o^T |u(\tau)|d\tau$, we define the following:

$$Q = \left[\begin{array}{c} -x_1(o) \\ -x_2(o) \\ \\ -x_m(o) \\ \hline 0 \end{array}\right] = \left[\begin{array}{c} \int_o^T \Phi^{-1}(\tau)Bu(\tau)d\tau \\ \\ \hline \int_o^T |u(\tau)|d\tau \end{array}\right] + z \left[\begin{array}{c} 0 \\ 0 \\ \\ 0 \\ \hline -1 \end{array}\right] \qquad (3.23)$$

$$P = \left[\begin{array}{c} \int_o^T \Phi^{-1}(\tau)Bu(\tau)d\tau \\ \hline \int_o^T |u(\tau) d\tau \end{array}\right] \quad \text{and } U = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline -1 \end{array}\right]$$

so that

$$zU + P = Q \qquad (3.24)$$

where P is an element of the convex set $\Omega(T)$ generated by choosing all possible $u(t)$ such that $|u(t)| \leq M$. As will be seen, we are searching for that column vector $P = P^*$ which corresponds to point b of Figure 3-1. Note that the cost functional has been labeled the (m+1)st equation as in previous. sections.

With the definitions and conventions prescribed above, a generalized program for the problem at hand can be formulated. In such a program the column vectors of coefficients are variable — selected from a convex set, say $\Omega(T)$. The rules of the simplex optimization algorithm are employed to determine which column vector should be selected to decrease the cost functional. These rules, when applied to the generalized programming problem, amount to the same conditions for optimality derived by way of the Maximum Principle.

A generalized linear programming problem is of the following form (Reference 6):

$$\text{Minimize } z \text{ such that } zU + P = Q \tag{3.25}$$

where the terms are those defined in Equations (3.23 and 3.24).

P is any point in the convex set $\Omega(T)$, not necessarily the one which permits z to obtain its minimum. We can express P as follows:

$$P = \sum_{j=1}^{t} \lambda_j P^j , \quad \sum_{j=1}^{t} \lambda_j = 1 , \quad \lambda_j \geq 0 , \; t \geq m + 1$$

where P^j are any other points in the convex set. Equation (3.25) can be rewritten as an equivalent problem

$$\text{Find } \lambda, \; \lambda_i \geq 0 \quad i = 1, 2, \ldots, t$$

$$\text{To minimize } z \text{ such that } zU + \sum_{j=1}^{t} \lambda_j P^j = Q \tag{3.26}$$

$$\sum_{j=1}^{t} \lambda_j = 1 , \quad P^j \epsilon \Omega(T)$$

The dual of this problem is expressed as in Section 2.6

$$\text{Find } \pi , \; \pi_i \text{ unrestricted in sign, } \; i = 1, 2, \ldots, m+1$$

$$\text{To maximize } -v + \pi Q \text{ such that} \tag{3.27}$$

$$\pi P - v \leq 0 \quad \text{for all } P \epsilon \Omega(T)$$

One can readily see that setting the π of this section equal to $\hat{\pi}$ of

Section 3.1 and Figure 3-1, that $\pi P - v = 0$ of Equation (3.27) is the equation for the hyperplane of support depicted in Figure 3-1. Thus, $\pi_{\text{Sec. } 3.4} = \hat{\pi}_{\text{Sec. } 3.1}$. Further, the dual problem constraint, Equation (3.27),

$$\pi P - v \leq 0 \quad \text{for all} \quad P \in \Omega(T)$$

is precisely the condition for optimality derived geometrically in Neustadt's approach. The optimal $v = \hat{\pi}\, b$, where $\hat{\pi}$ and b are from Figure 3-1.

A constructive derivation of the Maximum Principle ensues directly from the algorithm for solving a generalized programming problem. This algorithm assumes we have initially on hand $t \geq m + 1$ choices $P^j \in \Omega(T)$ such that

$$zU_o + \lambda_1 P^1 + \lambda_2 P^2 + \dots\dots + \lambda_t P^t = Q \tag{3.28}$$

$$\lambda_1 + \lambda_2 + \dots\dots + \lambda_t = 1$$

has a feasible solution $\lambda^o \geq 0$. The $P^o = \sum \lambda_j^o P^j$ may correspond to the point ω of Figure 3-1 and hence not be optimal. In order to obtain this initial feasible solution, we employ the Phase I starting procedure mentioned in the previous chapter. In the process of obtaining the initial λ^o we also obtain an initial set of dual variables $\pi^o = \left(\pi_1^o, \pi_2^o, \dots \pi_{m+1}^o\right)$ and the scalar v. These are the variables of a problem which is dual to Equation (3.28). This dual is written out as Equation (2.35) of the previous chapter.

Given an initial "price" vector π^o our task will be to find a (t + 1)st column which, when substituted for one of the P^j in Equation (3.28), will yield a lower value for z.

The decomposition technique described in Section 2.7 calls for solving the subproblem

$$\max_{P^j \epsilon \Omega} \left[\pi P^j \right] \qquad j = t + 1, 2, \ldots$$

From this, a new P^j is constructed and adjoined to equation set (3.28). Thereafter the simplex algorithm is applied to the augmented set, Equation (3.28), to result in removal of a column P^s, $s = \{1, 2, \ldots, t\}$ indicated by $\lambda_s = 0$ in the new solution. A new price vector π^k is simultaneously generated and the process is repeated until all $\pi^k P^k \geq 0$, $k = t + 1, t + 2, \ldots$.

If Ω is a convex polyhedron the process will yield an optimal solution in a finite number of steps. If Ω is strictly convex the π^k and P^k will approach their optimal values π^* and P^* monotonically.

When developed as above, the π^* has the fundamental property that

$$\pi^* P \leq \pi^* P^* = v \quad \text{for all} \quad P \epsilon \, \Omega(T)$$

But note that

$$P^j = \begin{bmatrix} \int_o^T \Phi^{-1}(\tau) B u^j(\tau) d\tau \\ \hline \int_o^T |u^j(\tau)| \, d\tau \end{bmatrix} \qquad \text{for some } |u^j(t)| \leq M$$

and the criterion for entering P^j, $j = t+1, t+2, \ldots$ in the k^{th} master program, $k \geq t+1$ is

$$\operatorname*{Max}_{P\epsilon\,\Omega(T)} \left(\begin{bmatrix} \pi_1 \pi_2 & \pi_m & \pi_{m+1} \end{bmatrix} \begin{bmatrix} \int_o^T \Phi^{-1}(\tau) B\, u(\tau)\, d\tau \\ \hdashline \int_o^T u(\tau)\, d\tau \end{bmatrix} \right)$$

$$= \int_o^T \operatorname*{Max}_{|u(\tau)| \leq M} \left[\pi\, \Phi^{-1}(\tau) B\, u(\tau) - |u(\tau)| \right] \qquad (3.29)$$

where $\pi_{m+1} = -1$ and π is the corresponding adjoint initial condition of Section 3.2. The criterion for optimality expressed by Equation (3.29) above, is equivalent to the Maximum Principle' s necessary condition expressed by Equation (3.9).

For a more complete treatment of generalized programming, the reader is referred to Dantzig[6] and Van Slyke.[26] In those treatises it is shown that generalized programming converges under conditions which are general enough to be useful in control optimization problems.

A discrete time problem illustrating the theory developed in this section will be solved in Chapter V.

One added note to this section is worthy of attention. The vector Q, as defined in Equation (3.23), is the initial condition vector, a point in state space. It is easily seen that Q may be handled exactly as P in the algorithm. Hence it is possible to specify Q as a convex set, the set of allowable initial (or terminal) conditions.

CHAPTER IV

LINEAR PROGRAMMING COMPUTATION OF MINIMUM FUEL CONTROL SEQUENCES

4.0 Introduction

The preceding chapter outlined the theory underlying computation of fuel optimal control laws for systems modeled with time as an independent continuous variable. In such representations, the control force magnitude may change anywhere in the time continuum.

This chapter introduces a computational method for the discrete time, or sampled data, case. Here the control magnitude is permitted to change only at pre-specified discrete points.

The discrete control process arises naturally when a digital computer is to be used to generate the required control inputs. In other instances, the nature of the system being modeled requires that time be considered a discrete sequence, as for example, in inventory control. Then too, there exist physical systems wherein data upon which to base control computations, are available only intermittently. In these truly "sampled data" cases, it may also be advantageous to permit the control to change only at intermittent points. As control theory grows, the number of applications requiring discrete time analysis and computation will grow also.

The Linear Programming computation method to be introduced in this chapter is suitable for certain types of discrete time dynamic systems. In this report a computational method for minimum fuel optimal control will be developed. Small scale computations on experimental problems in this and the next chapter will illustrate the procedure. This will set the groundwork for the larger scale orbital rendezvous control computation of Chapter VI. We first derive the required difference equations.

4.1 Formation of Difference Equations

Let the state differential equation of the system to be controlled be as follows

$$\dot{y}(t) = A(t)\, y(t) + B(t)\, u(t) \tag{4.1}$$

Then

$$y(t) = \Phi(t, t_o)\, y(t_o) + \Phi(t, t_o) \int_{t_o}^{t} \Phi^{-1}(\tau, t_o)\, B(\tau)\, u(\tau) d\tau$$

or

$$y(t) = \Phi(t, t_o)\, y(t_o) + \int_{t_o}^{t} \Phi(t_o, \tau) B(\tau)\, u(\tau) d\tau$$

Assume that the control $u(t)$ can change only at instants of time t_k so that

$$u(\tau) = u(k) \quad \text{for} \quad t_k \leq \tau < t_{k+1}$$

Thus

$$y(t_{k+1}) = \Phi(t_{k+1}, t_k)\, y(t_k) + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)\, B(\tau)\, u(\tau) d\tau$$

Letting $y(t_{k+1}) = y(k+1)$ we obtain

$$y(k+1) = \Phi(t_{k+1}, t_k)\, y(k) + \left[\int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)\, B(\tau) d\tau \right] u(k)$$

Next define

$$\Phi(t_{k+1}, t_k) \quad \triangleq \quad A(k)$$

$$\left[\int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)\, B(\tau) d\tau \right] \triangleq \quad B(k)$$

So that the discrete form of the differential system Equation (4. 1) becomes

$$y(k+1) = A(k)\ y(k) + B(k)\ u(k) \tag{4. 2}$$

If the original differential equation has constant coefficients the following simplification is possible. By setting $T = t_{k+1} - t_k$

$$\Phi(t_{k+1}, t_k) = \Phi\left[(k+1)T - kT\right] = \Phi(T) \quad \triangleq A$$

$$\left[\int_{kT}^{(k+1)T} \Phi[(k+1)T - \tau]\,B\,d\tau\right] = \int_{o}^{T} \Phi(\tau)B\,d\tau \ \triangleq B$$

With these definitions, we obtain the constant coefficient discrete counterpart of Equations (4. 1 and 4. 2) namely

$$y(k+1) = A\ y(k) + B\ u(k) \tag{4. 3}$$

To avoid confusion, we must note that the symbols A, A(t), and A(k) in Equations (4. 3, 4. 2, and 4. 1) are different and distinct matrices. This is also true for B, B(t), and B(k).

We may now consider the state changes in our controlled system to be governed by the difference equation:

$$y(k+1) = A(k)\ y(k) + B(k)\ u(k) \tag{4. 4}$$

where y(k) is an m-dimensional state vector; u(k) is an r-dimensional input vector. A(k) is an mxm matrix which may vary with the time index k, or be a constant. Similarly B(k) is an mxr matrix with dependence on time index k, or a constant.

The terminal state y(K) can be computed from y(0), the initial state, and u(k), the input sequence, by the relation

$$y(K) = \prod_{j=0}^{K-1} A(j)\ y(0) + \sum_{j=0}^{K-1} \left\{ \prod_{i=j+1}^{K-1} A(i) \right\} B(j)\ u(j) \tag{4. 5}$$

where $A(i) = I$ for $i = K - 1$. For constant matrices A and B, the corresponding relation is written

$$y(K) = A^{K} y(0) + \sum_{j=0}^{K-1} A^{K-1-j} B\, u(j) \tag{4.6}$$

One form of terminal control problem considers $y(k)$ and $y(0)$ as fixed final and initial conditions, respectively. The integer K, corresponding to the final time, may be fixed or free.

A in Equation (4.6) is a non-singular constant transition matrix. Equation (4.6) may therefore be written:

$$A^{-K} y(K) - y(0) = \sum_{j=0}^{K-1} A^{-1-j} B\, u(j) \tag{4.7}$$

If $y(k) = 0$, the expression

$$- y(0) = + \sum_{j=0}^{K-1} A^{-1-j} B\, u(j) \tag{4.8}$$

represents all the initial states $y(0)$ from which equilibrium $\left(y(K) = 0\right)$ can be reached in K discrete time periods with an appropriate sequence of control inputs $\{u(j)\}$.

Define the matrix $R(j)$ as:

$$R(j) = A^{-1-j} B \tag{4.9}$$

and also note the recursion relation:

$$R(j+1) = A^{-1} R(j) \tag{4.10}$$

With this definition Equation (4.8) may be written as follows:

$$- y(0) = \sum_{j=0}^{K-1} R(j)\, u(j) \tag{4.11}$$

Here $y(0)$ is an m-vector; $R(j)$ is an mxr matrix and $u(j)$ is the r-vector representing multiple control of the plant at sample time j.

Case I Single Control: For the single control input, $u(j)$ is a scalar and B, in Equation (4.9), an m-vector. With this convention $R(j)$ defined in Equation (4.9) is perforce an m-vector. Equation (4.11) thus appears as

$$\begin{bmatrix} -y_1(0) \\ -y_2(0) \\ \\ -y_m(0) \end{bmatrix} = \begin{bmatrix} r_1(0) \\ r_2(0) \\ \\ r_m(0) \end{bmatrix} u(0) + \cdot \cdot \cdot + \begin{bmatrix} r_1(K-1) \\ r_2(K-1) \\ \\ r_m(K-1) \end{bmatrix} u(K-1)$$

or more concisely as

$$-y(0) = [R]u = \sum_{j=0}^{K-1} r(j)\,u(j) \tag{4.12}$$

where $y(0)$ is an m-vector at sample time zero, and

$[R]$ is an (mxK) - matrix

$r(j)$ is the j^{th} column vector of $[R]$

$u(j)$ is a scalar, the j^{th} component of the K-vector u

Case II Multiple Control: For situations involving more than one independent input (multiple control), a slightly different notation is useful. To develop it, expand Equation (4.11) as follows

$$\begin{aligned} -y(0) &= A^{-1}B\,u(0) + A^{-2}B\,u(1) + \text{-----} + A^{-K}B\,u(K-1) \\ -y(0) &= R(0)\,u(0) + R(1)\,u(1) + \text{-----} + R(K-1)\,u(K-1) \end{aligned} \tag{4.13}$$

where now

$$\begin{aligned} R(0) &= A^{-1}B \\ R(j+1) &= A^{-1}\,R(j) \end{aligned} \tag{4.14}$$

and

R(j) is an (m x r) - matrix

u(j) is an (r x 1) - vector

The Case II, multiple control, notation for $r = 2$ will be required in Chapter VI.

4.2 Linear Programming Approach

Consider the control of a plant whose behavior is characterized by the difference equation

$$y(k+1) = A\ y(k) + B\ u(k) \tag{4.15}$$

where $y(k)$, $k = 0, 1, \ldots, K$ is an m-vector, and $u(k)$, $k = 0, 1, \ldots, K\text{-}1$ an r-vector. To simplify the presentation, consider the single control input case where $u(k)$ is a scalar quantity. The development to follow holds in both the single as well as multiple input case. However, in the multiple input case, the arithmetic complexity of the problem grows with increased r.

As seen in the preceding section when $y(K) = 0$ and $y(0)$ are given, the following relation holds

$$-\ y(0) = \sum_{j=0}^{K-1} A^{-1-j} B\ u(j) \tag{4.16}$$

Letting

$$R(j) = A^{-1-j}\ B$$

The expression

$$-\ y(0) = \sum_{j=0}^{K-1} r(j)\ u(j) = [R]u \tag{4.17}$$

results. Here $y(0)$ is an m-vector, $r(j)$ is an m-vector, but $u(j)$ is a scalar. The matrix $[R]$ is $(m \times K)$. Vector u, in this context, becomes a K-vector with components u(j). To clarify, expand Equation (4) as follows:

$$\begin{Bmatrix} -y_1(0) \\ -y_2(0) \\ \vdots \\ -y_m(0) \end{Bmatrix} = \begin{bmatrix} r_{10} & r_{11} & \cdots & r_{1(K-1)} \\ r_{20} & r_{21} & & \vdots \\ \vdots & & & \vdots \\ r_{mo} & \cdots & \cdots & r_{m(K-1)} \end{bmatrix} \begin{Bmatrix} u(0) \\ u(1) \\ \vdots \\ u(K-1) \end{Bmatrix} \qquad (4.18)$$

The m-vector y(0) on the left hand side represents the initial condition in which the plant is found at the time when the control sequence $\{u(k)\}$ is to begin. This sequence is to return the plant to its equilibrium state, y(K) = 0, in a fixed length of time corresponding to K sampling periods. It must do this using minimum fuel where fuel performance is measured by

$$P(K) = \sum_{j=0}^{K-1} |u(j)|T \quad \text{where } T = \text{sampling period} \qquad (4.19)$$

This is a type of minimum effort final value control problem which is solvable by linear programming. Consider the following substitution. Let

$$u(j) = u_j' - u_j'' \qquad u_j' \text{ and } u_j'' \geq 0$$

Then

$$P = \sum_{j=0}^{K-1} \left(u_j' - u_j''\right) T \qquad (4.20)$$

For a given j, u_j' and u_j'' produce linearly <u>dependent</u> column vectors in the resulting constraint equation matrix. Hence a basic solution cannot contain both u_j' and u_j'' as non-zero components.

Hence

$$P = \sum_{j=0}^{K-1} |u(j)|T = \sum_{j=0}^{K-1} \left(u'_j - u''_j\right) T$$

The LP computational form, equivalent to the discrete time minimum fuel problem, can now be stated. Setting $T = 1$ without loss of generality, we have the following primal and dual statements:

<u>Primal:</u>

(a) Find $u = \left(u'_0 - u''_0 , u'_1 - u''_1 , \ldots , u'_{K-1} - u''_{K-1}\right)$

(b) Subject to $-y(0) = [R]u$ and

$u'_j , u''_j \geq 0 \quad j = 0, 1, 2, \ldots, (K-1)$ (4.21)

(c) To minimize

$$P = \sum_{j=1}^{K-1} \left(u'_j - u''_j\right)$$

<u>Dual:</u>

(a) Find $\pi = \left(\pi_1, \pi_2, \text{-----} \ \pi_m\right)$

(b) Subject to $\pi \, [R] \leq [1, 1, \text{-----} \ 1]$ (4.22)

(c) To maximize

$$D = \sum_{i=1}^{m} - y_i(0) \, \pi_i$$

The ordinary simplex method, as well as many of its variations can be used to solve both the primal and dual problems in the same computer run. If an optimal solution exists, P_{min} must equal D_{max}. The Primal and Dual Problems are characterized in the following tableau.

$$\begin{array}{c|ccccc|c} & u'_0 & u''_0 & & u'_{K-1} & u''_{K-1} & \\ \hline \pi_1 & (r_{10}) & (-r_{10}) & ------ & (r_{1,K-1}) & (-r_{1,K-1}) & -y_1(0) \\ \pi_2 & (r_{20}) & (-r_{20}) & & \vdots & \vdots & -y_2(0) \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ \pi_m & (r_{m0}) & (-r_{m0}) & ------ & (r_{m,K-1}) & (-r_{m,K-1}) & -y_m(0) \\ \hline & 1 & 1 & & 1 & 1 & \end{array}$$

(4.23)

4.3 Manual Computation Illustrated

In this section we shall illustrate the LP formulation of the minimum fuel problem with a simple hand calculated example. Pivot reduction will be used to execute the matrix operations described in Section 2.3. The reader is referred to Gass[27] for a more detailed description of manual calculations in Linear Programming.

Let the plant be governed by Equation (4.15) where

$$A = \begin{bmatrix} 1/3 & 0 \\ 0 & 1/2 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad y(0) = \begin{bmatrix} 10 \\ 2 \end{bmatrix}$$

For K = 3 (Equation (4.17) becomes

$$-y(0) = \begin{bmatrix} 3 \\ 2 \end{bmatrix} u(0) + \begin{bmatrix} 9 \\ 4 \end{bmatrix} u(1) + \begin{bmatrix} 27 \\ 3 \end{bmatrix} u(2) \qquad (4.24)$$

We wish to minimize

$$P(3) = |u(0)| + |u(1)| + |u(2)| \qquad (4.25)$$

Calculation for this problem are outlined in Computation Summary I. The hand calculation method begins by expressing the constraint equations, Equation (4.24), in detached matrix form as follows:

	u'_0	u''_0	u'_1	u''_1	u'_2	u''_2	$-y(0)$	e_1	e_2
e_1	3	-3	9	-9	27	-27	-10	1	0
e_2	2	-2	4	-4	8	- 8	- 2	0	1

(4.26)

where e_1 and e_2 represent unit column vectors. The variables listed along the top outer row are those of interest in the problem. A pivot operation has the effect of replacing a variable in the left outer column with one from the top row. Pivot elements are circled in Tableau A. Thus, in the first array, the pivot element ③, indicates that e_1 in the left outer column is to be replaced by u'_0.

In the third array of Tableau A, two pivot operations have been completed. The constraints are in Canonical Form with u'_0 and u''_1 basic. By taking the inner product of each row with the top row, and by setting the non-basic variables u''_0, u'_1, u'_2, u''_2 to zero, we obtain a basic feasible solution $u'_0 = 11/3$, $u''_1 = 7/3$.

At this point we have completed the equivalent of Phase I of the Simplex method. In terms of the derivation provided in Section 2. 3, we have accomplished the following:

(a) Starting with the given constraints as in Equation (4. 21)

$$[R]\left(\begin{array}{c} \uparrow \\ u \\ \downarrow \end{array}\right) = \{-y(0)\}\ ;\ [I] = \{e_1\} + \{e_2\}$$

(b) Apply the operator B^{-1} to obtain

$$\left[I \,\vdots\, B^{-1} R_{NB} \right] \left(\begin{array}{c} u_B \\ \hline u_R \end{array}\right) = \{-B^{-1} y(0)\} ; [B^{-1} I] = \{B^{-1} e_1\} + \{B^{-1} e_2\}$$

R_{NB} refers to the non-basic part of the tranformed $[R]$ matrix.

With the foregoing in mind, the third array in Tableau A may be keyed to the following diagram (after appropriate column rearrangement):

$$\begin{array}{c||c:c||c||cc||}
 & \leftarrow u_B \rightarrow & \leftarrow u_R \rightarrow & -y(0) & e_1 & e_2 \\ \hline\hline
\updownarrow u_B & I & B^{-1}R_{NB} & -B^{-1}y(0) & \multicolumn{2}{c||}{B^{-1}} \\ \hline
\end{array} \tag{4.27}$$

where $u_B = \left[u_0', \; u_1''\right]$ and $u_R = \left[u_0'', \; u_1', \; u_2', \; u_2''\right]$.

The optimization procedure may now be stated. This is done in Tableaus B and C. Referring to Section 2.3, we calculate the scalar ζ_j for each column of the array. Our cost coefficients in this problem all equal 1.0. Hence $C_B = [1, 1]$. The scalar ζ_j is recorded as shown in Tableau B.

Also recorded in Tableau B is the quantity $\zeta_j - c_j$. Furthermore, from Equation (2.13)

$$\bar{c}_j = -\left(\zeta_j - c_j\right) \tag{4.28}$$

We are now in position to apply the column selection rule of Section 2.2: Choose as a candidate for the basic feasible solution that column s such that

$$\bar{c}_s = \operatorname*{Min}_j \bar{c}_j < 0 \tag{4.29}$$

This implies that we choose that column in Tableau B such that

$$\left(\zeta_s - c_s\right) = \operatorname*{Max}_j \left(\zeta_j - c_j\right) > 0 \tag{4.30}$$

We see that the column under u_2'' meets this criterion. Application of the row selection rule then leads to designation of the element + 5 as a pivot element.

Tableau C completes the computation. All $\left(\zeta_j - c_j\right) \leq 0$ and an optimal solution has been obtained. If the detached coefficients in the lines designated by arrows at the left of Tableau C, were rewritten with their variables, we would have the Optimal Canonical Form of Equation (2.7). Note, however, that some $\left(\zeta_j - c_j\right) = 0$ indicating that other solutions exist with fuel cost equal to the minimum. As an example the control, $u(1) = + 13/18$, $u(2) = - 11/18$ (derived by pivoting on $+ 6/5$), is also an optimal solution.

Another significant output of Tableau C is the dual solution.

$$\pi = \left[c_B \, B^{-1}\right] = \left[- 1/3 \; , \; + 1\right] \tag{4.31}$$

These dual variables will be used in the work to follow.

COMPUTATION SUMMARY I
TEST PROBLEM: SECTION 4.3

Constraints:

$$- y_1(0) = 3\left(u'_o - u''_o\right) + 9\left(u'_1 - u''_1\right) + 27\left(u'_2 - u''_2\right)$$

$$- y_2(0) = 2\left(u'_o - u''_o\right) + 4\left(u'_1 - u''_1\right) + 8\left(u'_2 - u''_2\right)$$

Performance Criterion:

$$\text{Minimize} \quad P = \left|u_o\right| + \left|u_1\right| + \left|u_2\right|$$

$$\text{or} \qquad P = \left|u'_o - u''_o\right| + \left|u'_1 - u''_1\right| + \left|u'_2 - u''_2\right|$$

u'_j, $u''_j \geq 0$ and not both different from zero for same $j = 0, 1, 2$.

COMPUTATION SUMMARY I (Continued)

TABLEAU A

	u'_0	u''_0	u'_1	u''_1	u'_2	u''_2	$-y(0)$	e_1	e_2
e_1	$\textcircled{3}$	- 3	9	- 9	27	- 27	- 10	1	0
e_2	2	- 2	4	- 4	8	- 8	- 2	0	1
u'_0	1	- 1	3	- 1	9	- 9	- 10/3	1/3	0
e_2	0	0	-2	$\textcircled{+2}$	-10	+ 10	+ 14/3	- 2/3	1
u'_0	1	- 1	0	0	- 6	+ 6	+ 11/3	- 2/3	3/2
u''_1	0	0	-1	1	- 5	+ 5	+ 7/3	- 1/3	1/2

End Phase I: Feasible Solution is

$u'_0 = + 11/3 \quad u''_1 = + 7/3$

Begin Phase II: Calculate Min P

TABLEAU B

	u'_0	u''_0	u'_1	u''_1	u'_2	u''_2	$-y(0)$	e_1	e_2
u'_0	1	- 1	0	0	- 6	+ 6	+ 11/3	- 2/3	3/2
u''_1	0	0	- 1	1	- 5	$\textcircled{+5}$	+ 7/3	- 1/3	1/2
ζ_j	1	- 1	- 1	1	- 11	+ 11	+ 18/13	- 1	+ 2
$\zeta_j - c_j$	0	- 2	- 2	0	- 12	+ 10	——	——	——

COMPUTATION SUMMARY I (Continued)

TABLEAU C

	u'_0	u''_0	u'_1	u''_1	u'_2	u''_2	$- y(0)$	e_1	e_2
⇒ u'_0	1	- 1	6/5	- 6/5	0	0	13/15	- 4/15	9/10
⇒ u''_2	0	0	- 1/5	1/5	- 1	1	7/15	- 1/15	1/10
ζ_j	1	- 1	1	- 1	- 1	+ 1	4/3	- 1/3	1
⇒ $\zeta_j - c_j$	0	- 2	0	- 2	- 2	0	——	——	——
								π_1	π_2

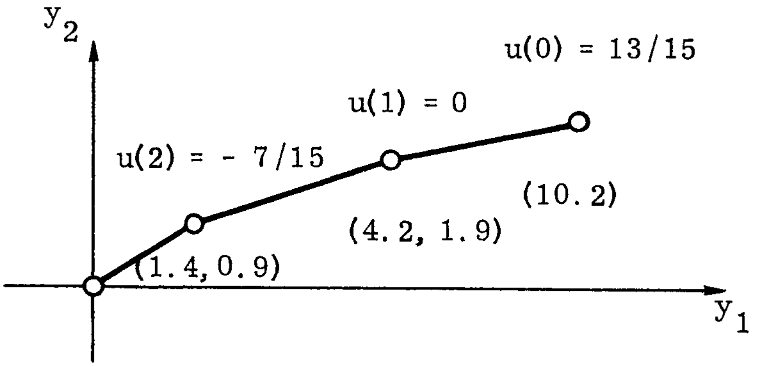
Solution: $\pi_1 = - 1/3$, $\pi_2 = +1$

$u(0) = + 13/15$, $u(1) = 0$, $u(2) = - 7/15$

Min $P = 4/3$

Max $D = - 10\pi_1 - 2\pi_2 = 4/3$

Trajectory:



END COMPUTATION SUMMARY I

4. 4 Machine Computation Illustrated

The last section demonstrated the manual calculation of fuel optimal controls for the system given by Equation (4. 15) with K = 3, and

$$A = \begin{bmatrix} 1/3 & 0 \\ 0 & 1/2 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad y(0) = \begin{bmatrix} 10 \\ 2 \end{bmatrix} \tag{4.32}$$

No constraint was imposed on the magnitude of $u(j)$. This section will present the results of machine computation of optimal control sequences for the same system with $K = 10$, and with bounded control $|u(j)| \leq 1$, $j = 0, 1, 2, \ldots, 9$.

The "M3 Linear and Separable Programming System" at the University of California, Berkeley, was used to make these calculations. The M3 code employs the revised simplex method in which the inverse (B^{-1}) is maintained in product form.[27] Problems with up to 300 rows and 4000 variables can be handled by the M3 system. The test problem presented here by no means taxes the M3 code' s computing power and versatility. The calculations were made merely to check out machine procedures, and to determine the general characteristics of control laws derived by the LP method.

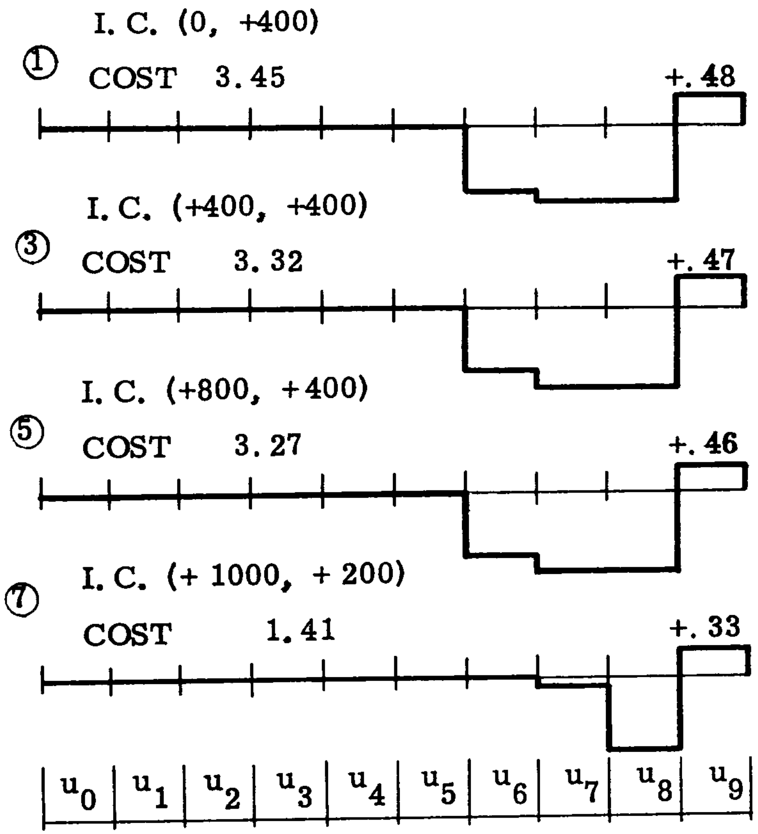
Computation Summary II lists runs were made without difficulty. Representative control functions and trajectories for Run Numbers 1, 3, 5, and 7 are sketched in Computation Summary II. Because of the scaling difficulties in presenting the entire trajectory only the terminal portions, beginning with $k = 4$ are depicted. It should be noted that the "relay with dead zone" characteristic does not appear in these control sequences as it would in a continuous time case. Instead, the control sequence always contains an intermediate value of control magnitude. Lindorff[28] discusses this phenomenon for the sampled data minimum time control solution.

COMPUTATION SUMMARY II
TEST PROBLEM: SECTION 4.4

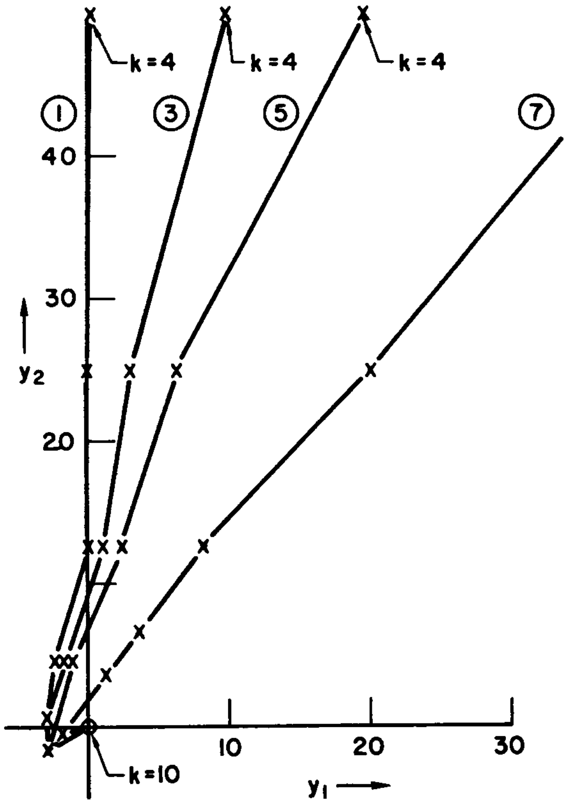
RUN NO.	INITIAL CONDITIONS		FUEL COST	CONTROL SEQUENCE K = 10 1(j) = 0 j ≤ 4 ; j = 10				
	$y_1(0)$	$y_2(0)$		u(5)	u(6)	u(7)	u(8)	u(9)
1	0	+ 400	3.45	0	- .97	- 1.0	- 1.0	+ .48
2	+ 200	+ 400	3.40	0	- .93	- 1.0	- 1.0	+ .475
3	+ 400	+ 400	3.36	0	- .89	- 1.0	- 1.0	+ .47
4	+ 600	+ 400	3.32	0	- .85	- 1.0	- 1.0	+ .465
5	+ 800	+ 400	3.27	0	- .81	- 1.0	- 1.0	+ .46
6	+ 1000	+ 400	3.23	0	- .77	- 1.0	- 1.0	+ .456
7	+ 1000	+ 200	1.41	0	0	- .08	- 1.0	+ .33
8	+ 1000	0	0.15	0	0	0	+ 1.0	- .05
9	+ 1000	- 200	1.72	0	0	+ .33	+ 1.0	- .39
10	+ 1000	- 400	3.78	+ .28	+ 1.0	+ 1.0	+ 1.0	- .50
11	+ 800	- 400	3.71	+ .21	+ 1.0	+ 1.0	+ 1.0	- .50
12	+ 600	- 400	3.63	+ .14	+ 1.0	+ 1.0	+ 1.0	- .49
13	+ 400	- 400	3.57	+ .08	+ 1.0	+ 1.0	+ 1.0	- .49
14	+ 200	- 400	3.49	+ .01	+ 1.0	+ 1.0	+ 1.0	- .48
15	0	- 400	3.45	0	+ .97	+ 1.0	+ 1.0	- .48

COMPUTATION SUMMARY II (Continued)

(a) CONTROL FUNCTIONS



(b) TRAJECTORIES



END COMPUTATION SUMMARY II

4.5 Application of Dual Problem Analysis

The dual of the problem treated in Section 4.3 offers an opportunity to demonstrate a design technique for a closed loop sampled data system. This design will cause the system to be nulled in exactly K sample times with minimum fuel. The technique demonstrated here is a graphical one limited to second order systems without control bounds.

For the second order system of Section 4.3, the dual problem constraints may be plotted as in Figure 4-2. The convex set of dual feasible solutions appears as the shaded area. For K = 3 there are three "negative" constraint lines and three "positive" ones, corresponding to column vectors associated with u''_j and u'_j respectively. For the system parameters A and B given in Section 4.3, the set of feasible dual solutions has four extreme points. Since the cost function and constraints are linear, the optimum solution to the dual problem, $\pi = \pi^*$, must lie at one of these extreme points.

From Section 2.3 and Equation (4.31) we know that

$$\pi^* = c_B B^{-1} = [1, 1] B^{-1} \qquad (4.33)$$

The cost coefficients of the primal problem all equal 1.0. Hence $c_B = [1, 1]$, no matter what components of $u = [u'_1 \; u''_1 \; u'_2 \; u''_2 \; u'_3 \; u''_3]$ are basic.

If the primal solution, yielding the minimum fuel cost, were unique, there would be a unique B^{-1} for each y(0). The corresponding control sequence would be given by

$$u^* = - B^{-1} y(0) \qquad (4.34)$$

However, when the primal solution is not unique, there are several $B^{-1}_{(i)}$, $i \geq 2$ which yield

$$u^* = - B_{(i)}^{-1} \; y(0) \text{ such that } \sum_j |u(j)| \text{ is minimized.}$$

For example, in the problems of Section 4.3, the two control sequences

$$u_{(1)}^* = \left(\frac{13}{15}, 0, -\frac{7}{15}\right)$$
$$u_{(2)}^* = \left(0, \frac{13}{18}, -\frac{11}{18}\right) \qquad (4.35)$$

will both null $y(0) = [10, 2]^T$ at a minimum fuel cost, $P(K) = \frac{4}{3}$. There are others. Corresponding to each of them, there is a different $B_{(i)}^{-1}$. For the case at hand

$$B_{(1)}^{-1} = \begin{bmatrix} -\frac{4}{15} & \frac{9}{10} \\ -\frac{1}{15} & \frac{1}{10} \end{bmatrix} \text{ and } B_{(2)}^{-1} = \begin{bmatrix} -\frac{2}{9} & \frac{3}{4} \\ -\frac{1}{9} & \frac{1}{4} \end{bmatrix} \qquad (4.36)$$

such that

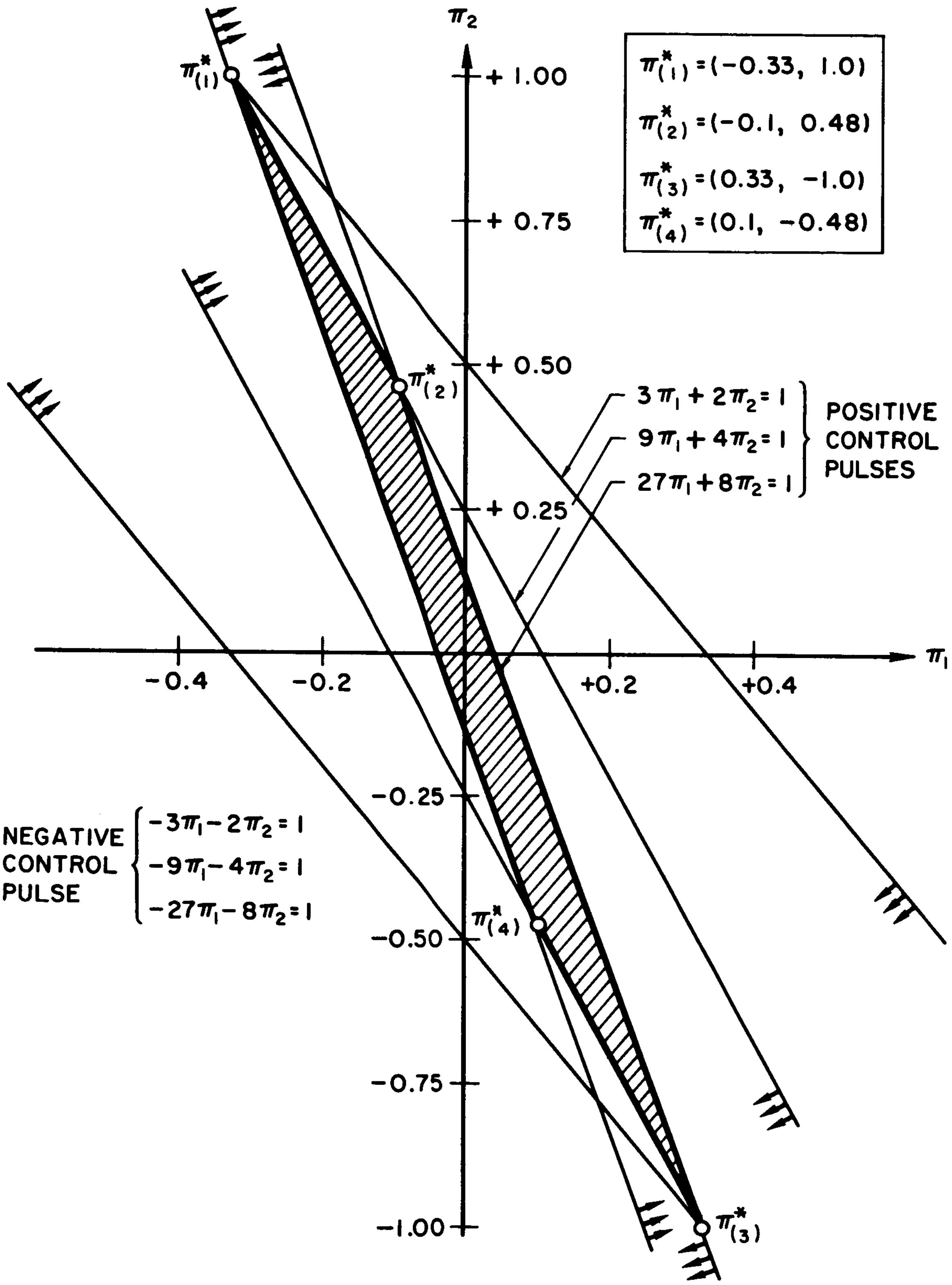
$$u_{(1)}^* = B_{(1)}^{-1} \begin{bmatrix} -10 \\ -2 \end{bmatrix} \text{ and } u_{(2)}^* = B_{(2)}^{-1} \begin{bmatrix} -10 \\ -2 \end{bmatrix} \qquad (4.37)$$

yield different trajectories with the same fuel usage.

Returning now to the dual problem graphed in Figure 4-2, we observe that the point $\pi^* = [-\frac{1}{3}, 1]$ is one of four optimal solutions which maximizes $D = -\pi\, y(0)$. To the extreme point $[-\frac{1}{3}, 1]$ there correspond several $B_{(i)}^{-1}$, $i \geq 2$. Two, $B_{(1)}^{-1}$ and $B_{(2)}^{-1}$, are stated in the previous paragraph. Let us arbitrarily select one of them, k, such that $k \in \{i\}$. Now, from Equation (4.33)

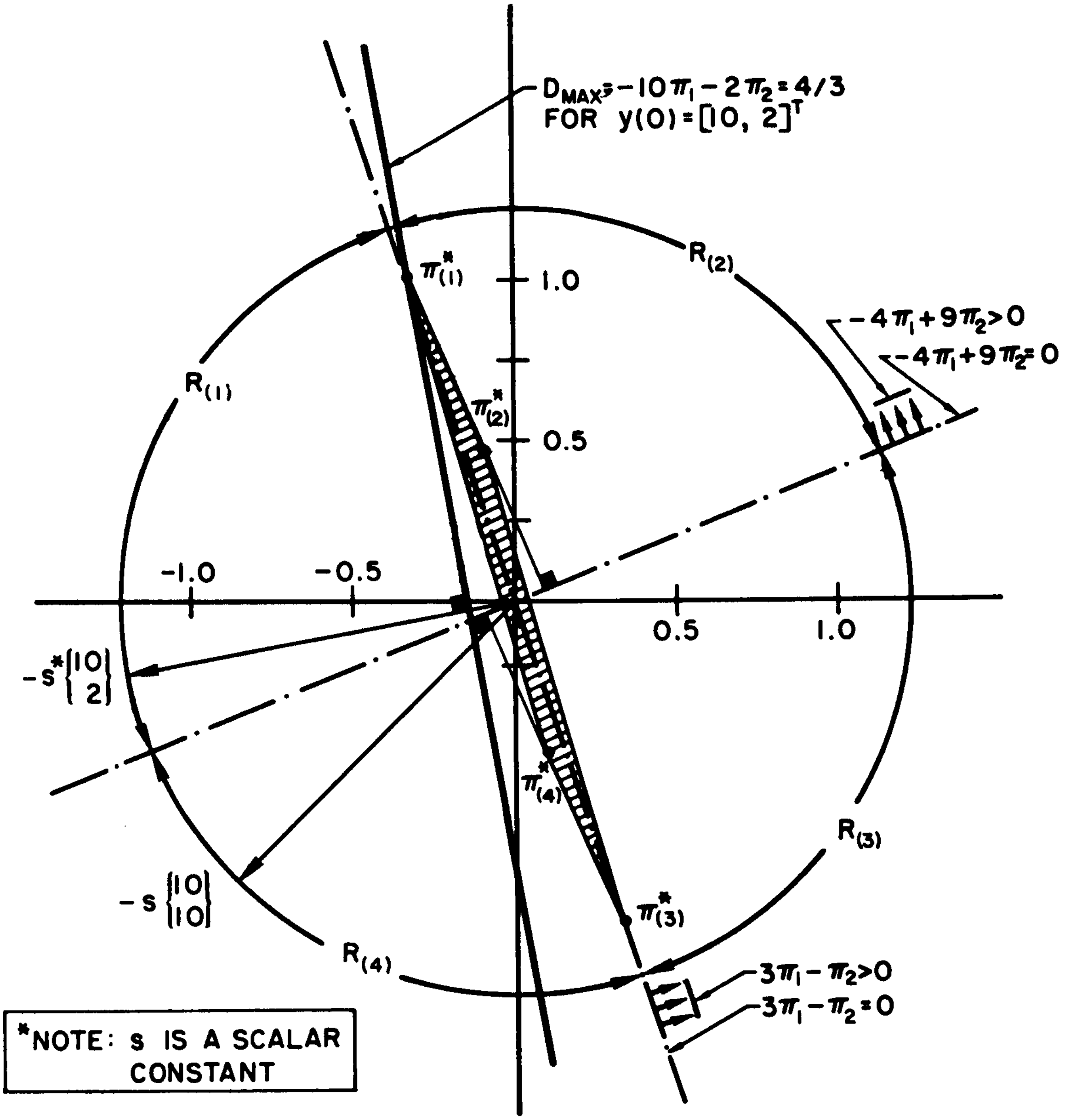
$$\pi_{(k)}^* = c_B \, B_{(k)}^{-1} = \left[-\frac{1}{3}, 1\right]; \quad k \in \{i\}$$

Note that $\pi_{(k)}^*$ is an optimal solution to the dual for a wide range of initial conditions $y(0)$. This range is labled $R_{(k)}$, $k = 1, 2, 3, 4$, in Figure 4-3. In Figure 4-3 there also appears an



PLOT OF DUAL CONSTRAINTS

FIGURE 4-1



LINEAR CONTROL REGION $R_{(k)}$

FIGURE 4-2

optimal $\pi^*_{(k)}$ for each range $R_{(k)}$ in which $y(0)$ is located. Each $\pi^*_{(k)}$, $k = 1, 2, 3, 4$ has an associated $B^{-1}_{(k)}$ which can be used to compute the fuel optimum control sequence. Thus,

$$y(0) \in R_{(k)} \Rightarrow - \pi^*_{(k)}\, y(0) = D_{\text{Max}}\Big(k,\ y(0)\Big) \tag{4.38}$$

$$D_{\text{Max}}\Big(k,\ y(0)\Big) = P_{\text{Min}}\Big(k,\ y(0)\Big)\ ,\quad k = 1, 2, 3, 4$$

The end product of our dual analysis has been the observation that, for each $R_{(i)}$ $i = 1, 2, 3, 4$, a different (not necessarily unique) $B^{-1}_{(i)}$ exists such that the fuel optimal control sequence u^* can be computed from

$$u^*_{(k)} = B^{-1}_{(k)}\, y(0) \qquad y(0) \in R_{(k)} \tag{4.39}$$

Thus the optimal control is linear within each of the regions $R_{(k)}$. For closed loop design we need only select the appropriate $B^{-1}_{(k)}$. Thus the two dimensional property of the dual permits the following graphical determination of the regions $R_{(k)}$. From Figure 4-3, we determine the following

$$\begin{gathered}
\left\{\begin{array}{ll} & -\,3\,y_1(0) + y_2(0) < 0 \\ \text{and} & +\,4\,y_1(0) - 9\,y_2(0) \geq 0 \end{array}\right] \Rightarrow -\,y(0) \in R_{(1)} \\
\left\{\begin{array}{ll} & -\,3\,y_1(0) + y_2(0) \geqslant 0 \\ \text{and} & +\,4\,y_1(0) - 9\,y_2(0) \geq 0 \end{array}\right] \Rightarrow -\,y\,(0) \in R_{(2)} \\
\left\{\begin{array}{ll} & -\,3\,y_1(0) + y_2(0) \geqslant 0 \\ \text{and} & +\,4\,y_1(0) - 9\,y_2(0) < 0 \end{array}\right] \Rightarrow -\,y(0) \in R_{(3)} \\
\left\{\begin{array}{ll} & -\,3\,y_1(0) + y_2(0) < 0 \\ \text{and} & +\,4\,y_1(0) - 9\,y_2(0) < 0 \end{array}\right] \Rightarrow -\,y(0) \in R_{(4)}
\end{gathered} \tag{4.40}$$

It remains only to determine the $B^{-1}_{(k)}$ for each of these regions. $B^{-1}_{(1)}$ corresponding to initial condition $y(0) = [10, 2]$ is available from Tableau C of Computation Summary I, that is

$$B^{-1}_{(1)} = \begin{bmatrix} -4/15 & 9/10 \\ -1/15 & 1/10 \end{bmatrix}$$

To find $B^{-1}_{(4)}$ we use the same computation schemata as given in Section 4. 3 but with a different $y(0)$ — one for which $-y(0)$ is in the range $R_{(4)}$. Let us choose $y(0) = [10, 10]^T$ so that $-y(0) = [-10, -10]^T \in R_{(4)}$. Repeating the procedure for $R_{(3)}$ and $R_{(2)}$ we obtain

$$\begin{aligned} B^{-1}_{(1)} = B^{-1}_{(2)} &= \begin{bmatrix} -\frac{4}{15} & \frac{9}{10} \\ -\frac{1}{15} & \frac{1}{10} \end{bmatrix} \\ B^{-1}_{(3)} = B^{-1}_{(4)} &= \begin{bmatrix} \frac{4}{15} & -\frac{9}{10} \\ \frac{1}{15} & -\frac{1}{10} \end{bmatrix} \end{aligned} \qquad (4.41)$$

The schemata and trajectories associated with these results are shown in Computation Summary III. Our "closed loop" design may be represented as in Part F of Computation Summary III.

COMPUTATION SUMMARY III
TEST PROBLEM: SECTION 4.5

(A) Starting with:

	u'_0	u''_0	u'_1	u''_1	u'_2	u''_2	(1)	(3)	(4)	(2)	e_1	e_2
e'_1	3	-3	9	-9	27	-27	-10	+10	-10	+10	1	0
e'_2	2	-2	4	-4	8	-8	-2	+2	-10	+10	0	1

Column (1) corresponds to $-y(0) = [-10, -2]^T \epsilon R_{(1)}$ Figure 4-2

(3) corresponds to $-y(0) = [+10, +2]^T \epsilon R_{(3)}$ Figure 4-2

(4) corresponds to $-y(0) = [-10, -10]^T \epsilon R_{(4)}$ Figure 4-2

(2) corresponds to $-y(0) = [+10, +10]^T \epsilon R_{(2)}$ Figure 4-2

(B) Apply Pivot Reductions to obtain:

	u'_0	u''_0	u'_1	u''_1	u'_2	u''_2	(1)	(3)	(4)	(2)	e_1	e_2
u'_0	1	-1	$\frac{6}{5}$	$-\frac{6}{5}$	0	0	$\frac{13}{15}$	$-\frac{13}{15}$	$-\frac{19}{3}$	$+\frac{19}{3}$	$-\frac{4}{15}$	$\frac{9}{10}$
u''_2	0	0	$-\frac{1}{5}$	$\frac{1}{5}$	-0	1	$\frac{7}{15}$	$-\frac{7}{15}$	$-\frac{1}{3}$	$+\frac{1}{3}$	$-\frac{1}{15}$	$\frac{1}{10}$
u''_0	-1	1	$-\frac{6}{5}$	$\frac{6}{5}$	0	0	$-\frac{13}{15}$	$\frac{13}{15}$	$\frac{19}{3}$	$-\frac{19}{3}$	$\frac{4}{15}$	$-\frac{9}{10}$
u'_2	0	0	$\frac{1}{5}$	$-\frac{1}{5}$	1	-1	$-\frac{7}{15}$	$\frac{7}{15}$	$\frac{1}{3}$	$-\frac{1}{3}$	$\frac{1}{15}$	$-\frac{1}{10}$

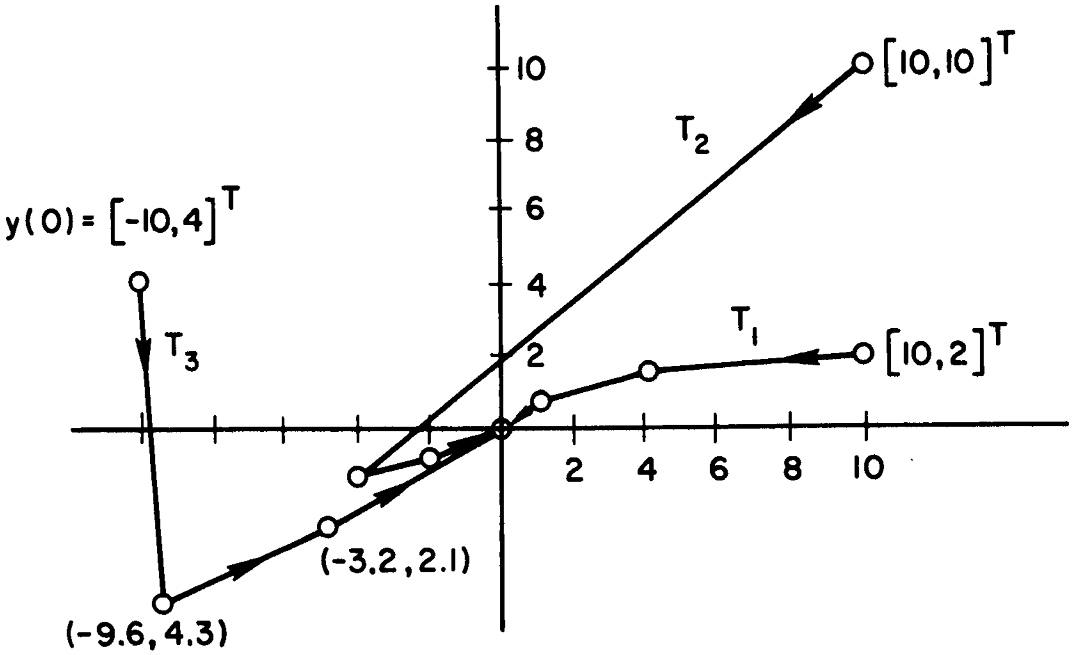
(C) Results:

For $4\,y_1(0) - 9\,y_2(0) \geq 0$
$$B^{-1}_{(1)} = B^{-1}_{(2)} = \begin{bmatrix} -\frac{4}{15} & \frac{9}{10} \\ -\frac{1}{15} & \frac{1}{10} \end{bmatrix}$$

For $4\,y_1(0) - 9\,y_2(0) < 0$
$$B^{-1}_{(3)} = B^{-1}_{(4)} = \begin{bmatrix} \frac{4}{15} & -\frac{9}{10} \\ \frac{1}{15} & -\frac{1}{10} \end{bmatrix}$$

COMPUTATION SUMMARY III (Continued)

(D) Trajectories T_1, T_2, T_3



(E) Example: Trajectory T_3

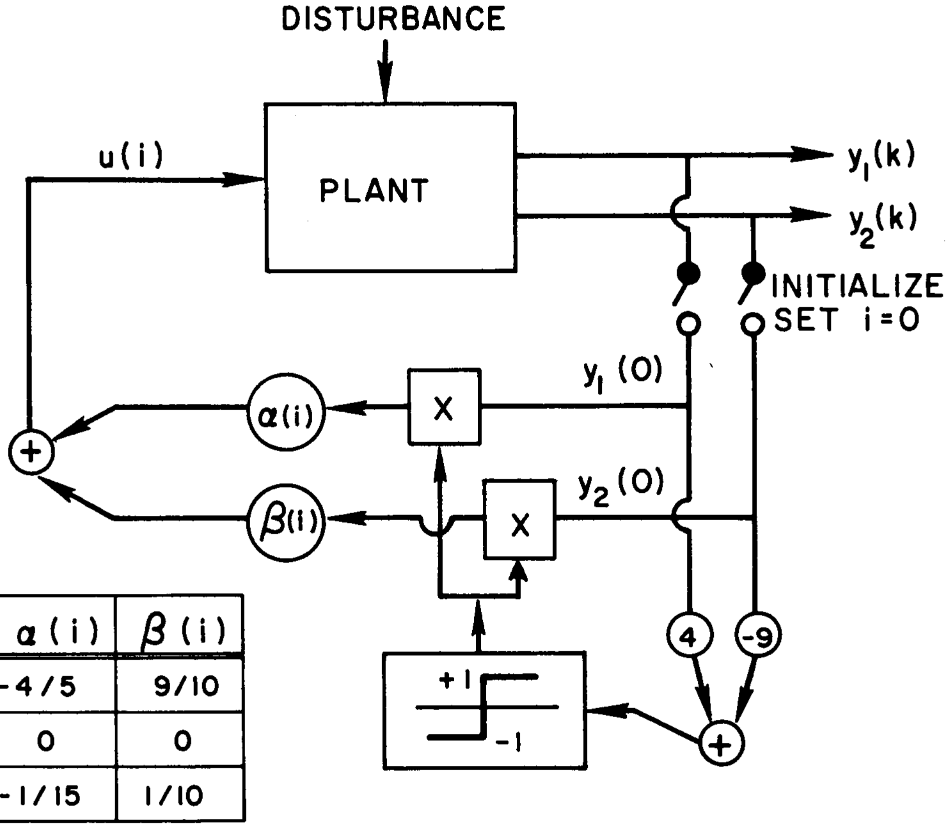
$$y(0) = [-10, 4] \; ; \; 4\, y_1(0) - 9\, y_2(0) = -76 < 0 \Longrightarrow - y(0) \in R_{(3)}$$

$$\text{Therefore} \quad B^{-1}_{(3)} = \begin{bmatrix} \frac{4}{15} & -\frac{9}{10} \\ \frac{1}{15} & -\frac{1}{10} \end{bmatrix}$$

$$u^* = B^{-1}_{(3)} \begin{bmatrix} 10 \\ -4 \end{bmatrix} = \begin{bmatrix} \frac{94}{15}, & \frac{16}{15} \end{bmatrix} \Longrightarrow \left\{ \begin{array}{lcr} u(0) & = & -\frac{94}{15} \\ u(1) & = & 0 \\ u(2) & & +\frac{16}{15} \end{array} \right\}$$

COMPUTATION SUMMARY III (Continued)

(F) Closed Loop Representation



i	α (i)	β (i)
0	-4/5	9/10
1	0	0
2	-1/15	1/10

END COMPUTATION SUMMARY III

CHAPTER V

MINIMUM FUEL OPTIMAL CONTROL – DISCRETE TIME CASE

5.0 Introduction

This chapter parallels the treatment of the continuous time minimum fuel optimization problem presented in Chapter III. The geometric, the maximum principle, the functional analysis, and the generalized programming approaches discussed there, will be treated here for the discrete time case. The relationship between the four approaches will be demonstrated by means of the numerical example introduced in the previous chapter.

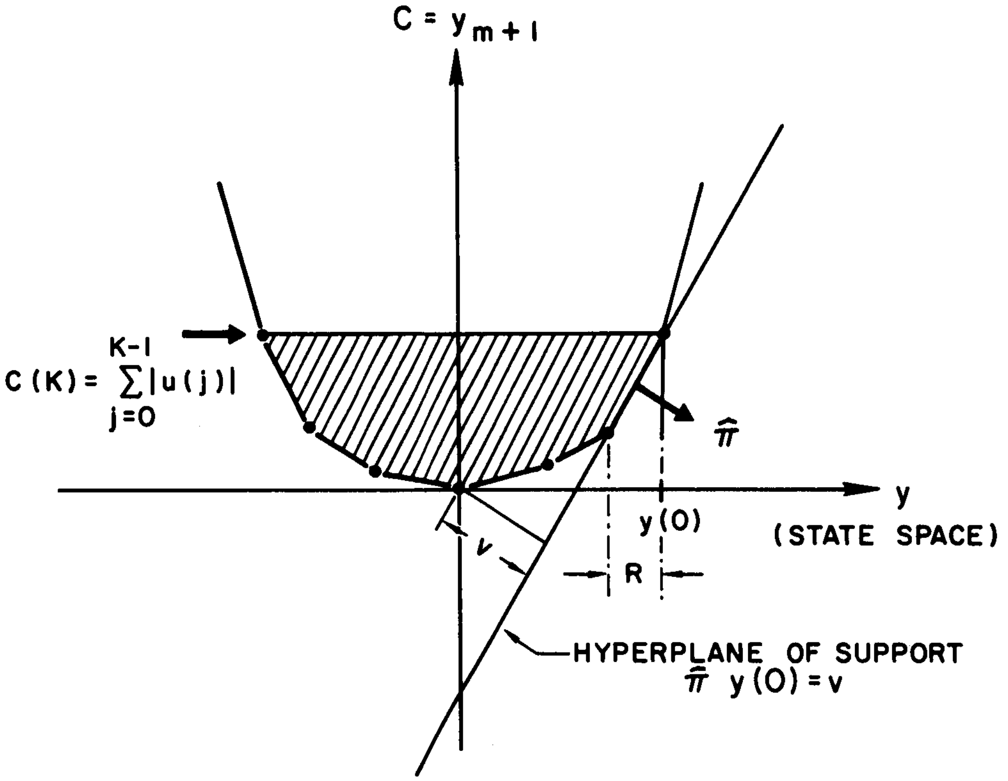
5.1 Geometric Approach

In this section, the implications of attempting to use Neustadt' s "Geometric Approach" for the discrete time case will be stated. When the convex set of recoverable states $\Omega(T)$, is generated by a differential equation system, a "smooth"[†] boundary for $\Omega(T)$ results. In this case (and if $\Omega(T)$ is strictly convex), a unique optimal control law can be defined. On the other hand, the set $\Omega(K)$ generated by a difference equation system is a convex polyhedron. Figure 5-1 depicting $\Omega(K)$ is the discrete time version of Figure 3-1. The sketch shows the possibility of having the same $\hat{\pi}$ (same hyperplane of support) for a finite range of initial conditions, R.

Now consider the use of the geometric approach to compute the optimal control sequence $\{u^*(k)\}$ for the following system

$$y(k+1) = A\ y(k) + B\ u(k) \tag{5.1}$$

[†] Note: A "smooth" boundary is taken here as one with a continuous first derivative.



THE CONVEX POLYHEDRON $\Omega(K)$

FIGURE 5-1

The initial state y(0) is to be nulled in three time periods, and $z = \sum_{j=0}^{2} |u(j)|$ is to be minimized. The steepest ascent method described in the last few paragraphs of Section 3.1 might be attempted. The discrete time version of Equation (3.6) is

$$y\left(3, \pi^{(i)}\right) = A^3 y(0) + \sum_{j=0}^{2} A^{2-j} B u\left(j, \pi^{(i)}\right) \qquad (5.2)$$

Hence the recurrence relation for $\pi^{(i)}$ in Equation (3.7) becomes

$$\pi^{(i+1)} - \pi^{(i)} = -k\left[y(0) + \sum_{j=0}^{2} A^{-1-j} B u\left(j, \pi^{(i)}\right)\right] \qquad (5.3)$$

The method (refer to Chapter III) calls for an initial guess $\pi^{(0)}$ to be made and the recursion relation Equation (5.3) to be solved until $\pi^{(i+1)} \to \pi^{(i)}$. From π^* we obtain u^*. In the continuous time case (with strictly convex $\Omega(T)$) the optimal control u^* is a bi-unique "smooth" function of π^*. But for discrete time, this is not so. As seen in Figure 5-1, a range of state space labeled R is associated with the same value of π. Even if Equation (5.3) could produce this value of π, we would still require another computing step to find the control magnitude u(j) which would cause the trajectory to match the given initial condition within R.

From a Linear Programming point of view, Neustadt' s method undertakes the problem of optimal control computation by calling for the solution to the dual problem. However, as should be expected from the foregoing observations, the analogy breaks down when the discrete-time minimum fuel problem is attempted. If the relationship between π^*, the initial condition y(0), and the optimal control u^* is not unique; we do not automatically satisfy the prescribed system equation end conditions. This difficulty might be remedied by developing supplemental necessary conditions, and by adding steps to the computations required under the geometric approach. However, this would lead us too far afield. Hence, a numerical example for the discrete time version of the geometric approach will not be carried out. Instead we will proceed immediately to examples for the discrete maximum principle, the functional analysis approach, and generalized programming.

5.2 Maximum Principle Approach

To introduce notation, this section begins with a statement of the Discrete Maximum Principle paralleling that given by Kleinman and Athans.[29]

Theorem: MAXIMUM PRINCIPLE FOR DISCRETE SYSTEMS

(1) Given a system of difference equations

$$y(k+1) - y(k) = f\big(y(k),\ u(k)\big) \tag{5.4}$$

where y and f are m dimensional column vectors. Let $\{y^*(k),\ k = 0, 1, \ldots, (k-1)\}$ be the trajectory of the given system corresponding to the optimal control sequence $\{u^*(k)\}$, where $|u^*(k)| \leq M$. $\{u^*(k)\}$ drives the system from an initial condition $y(0)$ to a target set in E_m called S.

(2) Then, in order that $\{u^*(k)\}$ minimize a cost functional $L\big(x(k),\ u(k)\big)$, it is necessary that there exist a sequence of m dimensional row vectors $\{\pi(k),\ k = 0, 1, \ldots, (k-1)\}$ called "costate" or "adjoint variables" such that

(A) The Hamiltonian scalar function

$$H\big(y^*(k),\ \pi^*(k+1),\ u(k)\big) = \pi^*_{m+1} L\big(y^*(k),\ u(k)\big) + \pi^*(k)\, f\big(y^*(k),\ u(k)\big) \tag{5.5}$$

has an absolute maximum as a function of $u(k)$, $|u(k)| \leq M$ at $u(k) = u^*(k)$ for every $k = 0, 1, \ldots, (k-1)$.

(B) The evolution of $\pi^*(k)$ in time is given by the vector difference equation

$$\pi^*(k+1) - \pi^*(k) = - \frac{\partial H\left(y^*(k),\ \pi^*(k+1),\ u^*(k)\right)}{\partial\, y(k)} \qquad (5.6)$$

for all $k = 0, 1, \ldots, (k-1)$.

(C) $\pi^*(K)$ is normal to the target set $S = \{y(k) \colon\ g_i[y(k)] = 0$ $i = 1, 2, \ldots, k \leq m\}$, that is, there are real numbers β_i, $i = 1, 2, \ldots, k$ such that

$$\pi^*(K) = \sum_{i=1}^{\beta} \beta_i \frac{\partial g_i\left(y(K)\right)}{\partial y} \qquad (5.7)$$

This statement of the maximum principle is more general than that which we require here. Let us therefore specify

$$L\left(y(k),\ u(k)\right) = \sum_{k=0}^{k-1} |u(k)| \qquad (5.8)$$

corresponding to Equation (4.19) with $T = 1$, and

$$f\left(y(k),\ u(k)\right) = [A - I]\ y(k) + B\ u(k) \qquad (5.9)$$

corresponding to Equation (4.15). In addition, our target set S reduces to a point, that is, k in item 2(C) above, equals m.

The necessary conditions given by the theorem above are dependent on (1) various important properties of the functions f and L of Equations (5.8 and 5.9), and (2) on the rationale used in their derivation.

With respect to the latter, the usual derivation of the maximum principle begins with postulating an optimal trajectory $x^*(t)$ or

$\{y^*(k)\}$. The control $u^*(t)$ or $\{u^*(k)\}$ is then perturbed, and the effect on the trajectory is expressed by the so called "Variational Equations." From these, and from knowledge of the properties of the target set S, or the set of reachable states $\Omega(T)$ or $\Omega(K)$ (e.g., compactness, convexity), the necessary conditions are derived. Jordan and Polak[30] point out a conceptual difficulty which arises when this rationale is applied in the discrete time case. Recall that in the continuous time case, it is possible to choose $\delta u(t)$ (perturbations to $u^*(t)$) which range in magnitude over the entire allowable control set and nevertheless affect the trajectory $x^*(t)$ only slightly. Because of this second order effects are kept negligible. Conceptually, this can be done by allowing the perturbation pulse width to become smaller and smaller as the magnitude of $\delta u(t)$ grows. This permits one to search out all of the control space for global maxima. In the discrete time case, however, the pulse width of the control function is always relatively large, and frequently fixed. Hence any large perturbation in the magnitude of the control sequence $\{u^*(k)\}$ has a large, possibly "nonlinear," effect on the optimal trajectory $\{x^*(k)\}$. Therefore, only small changes of magnitude in the control are permitted – with the result that only local conditions can be discovered. This observation reduces to the fact that, in arriving at a "maximum principle" for discrete time systems (even those which are linear), the analysis must consider second order terms. These terms determine the nature of stationary points of the function H. By considering their behaviour we can deduce a "Maximum Principle" from what otherwise would be a weaker "Stationarity Principle." The reader is referred to Jackson and Horn[31] for details on how to proceed with the necessary "second order" analysis.

One can avoid the above described difficulties in the derivation rationale by initially assuming certain properties for the functions f and L. These properties can be chosen to insure that the so called "second order" effects discussed above are negligible. Kleinman and Athans[29] describe Holtzman's "directional convexity" assumption. This assumption is stated as follows:

> Directional Convexity Assumption: Assume that for all pairs of allowable controls u and v, and for all real numbers $\alpha \in [0, 1]$, there exists a vector W_α within the set of allowable controls and a scalar $\beta \geq 0$ such that for all $k = 0, 1, \ldots, (K-1)$, the $(m+1)$ dimensional column vector
>
> $$Y\big(y(k),\ u(k)\big) = \begin{bmatrix} f\big(y(k),\ u(k)\big) \\ \text{------} \\ L\,\big(y(k),\ u(k)\big) \end{bmatrix}$$
>
> satisfies the relation
>
> $$Y\big(y(k),\ W_\alpha\big) = \alpha Y\big(y(k),\ v(k)\big) + (1-\alpha)\ Y\big(y(k),\ u(k)\big) + \beta z$$
>
> where z is the (m+1) column vector $[0, 0, \ldots 0, 1]^T$.

The directional convexity property protects the "discrete time" model from allowing a state change to take place at point k which reduces the value of the functional L. This danger can arise when a physical system is modeled directly as a discrete time process. However, if the discrete time model is derived from a continuous time model of a physical situation which can be solved by the Maximum Principle, the directional convexity assumption holds automatically.[29] It is the directional convexity assumption which permits us to convert a weaker "Stationarity Principle" into a "Maximum Principle" for discrete time systems. Directional convexity is required for the theorem stated at the beginning of this section to hold true. In particular, it allows us to use the term "absolute maximum" in necessary condition 2(A).

Fortunately, directional convexity holds for systems of the type we shall consider. More generally, systems for which

(a) Each $f\big(y(k),\ u(k)\big)$ is linear in $u(k)$ for all $k = 0, 1, \ldots, (K - 1)$, and

(b) The set of allowable controls and the last function are convex in $u(k)$ for all $k = 0, 1, \ldots, (K - 1)$,

have this property.[29]

The foregoing discussion outlined some of the recently developed concepts concerning the Discrete Maximum Principle. Its purpose was to set the stage for a numerical example in which the optimal fuel control sequence will be computed by Linear Programming and compared with a solution obtainable by use of the Maximum Principle.

Consider the system

$$\begin{Bmatrix} y_1(k+1) \\ y_2(k+1) \end{Bmatrix} = \begin{bmatrix} 1/3 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{Bmatrix} y_1(k) \\ y_2(k) \end{Bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k) \qquad (5.10)$$

$$A = \begin{bmatrix} 0.33 & 0 \\ 0 & 0.50 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For $K = 8$ and $y(0) = [111.1,\ 100.0]^T$, we desire $y(K) = [0, 0]^T$ and $P(K) = \sum_{k=0}^{7} |u(k)|$ to be minimized. The adjoint equation may be stated as

$$\begin{Bmatrix} \pi_1(k+1) \\ \pi_2(k+1) \end{Bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{Bmatrix} \pi_1(k) \\ \pi_2(k) \end{Bmatrix} \qquad (5.11)$$

and the Hamiltonian as

$$H\Big(y(k),\ u(k),\ \pi(k+1)\Big) = -\ |u(k)| + \pi(k+1)\ y(k+1) \qquad (5.12)$$

$$= -\ |u(k)| + \pi(k+1)\Big[A\,y(k) + B\,u(k)\Big]$$

The Maximum Principle approach calls for maximizing Equation (5.12) over all $u(k)$ such that $|u(k)| \leq M$, where M is the upper control bound. Without loss of generality we set $M = 1$. Then

$$\text{Max } \bar{H} = \bar{H}^*(k) = \max_{\substack{u(k) \\ |u(k)| \leq 1}} \Big\{\pi(k+1)\ B\ u(k)\ -\ |u(k)|\Big\} \qquad (5.13)$$

Equation (5.13) implies that the optimal control sequence $\{u^*(k)\}$ must satisfy, for all $k = 0, 1, \ldots, (K-1)$,

$$\left\{\begin{array}{lll} u^*(k) = 0 & \text{if} & |\pi(k+1)\,B| < 1 \\ u^*(k) = \text{sgn } \pi(k+1)B & \text{if} & |\pi(k+1)\,B| > 1 \\ u^*(k) \ \ \text{is undetermined} & \text{if} & |\pi(k+1)\,B| \equiv 1 \end{array}\right\} \qquad (5.14)$$

To continue the computation under the Maximum Principle approach we would insert Equation (5.14) into Equation (5.10) and simultaneously solve Equation (5.10) and Equation (5.11). Since we have split boundary conditions $y(0) = [111.1,\ 100.0]^T$ and $y(8) = [0, 0]^T$ an iterative process must be used. There is the added complication of the singularity condition when $|B'\ \pi(k+1)| \equiv 1$.

It is at this point where the LP computational method developed in Chapter IV can be used to advantage. Recall the fact demonstrated in Chapter III that the optimal simplex multipliers (dual variables) of Section 3.4 are the initial conditions to the adjoint equations of the Maximum Principle Approach.

The result of solving the problem at hand using the M-3 Linear Programming Code in the manner described by Section 4.4 is as follows

$$
\begin{aligned}
u(k) &= 0 \qquad\qquad k = 0, 1, 2, 3 \\
u(4) &= -0.77 \\
u(5) &= -1.0 \\
u(6) &= -1.0 \\
u(7) &= 0.46 \\
\pi_1(0) &= +0.001949 \\
\pi_2(0) &= -0.046053 \\
P(8) &= 3.23
\end{aligned}
$$

Computation Summary IV depicts these results along with the Maximum Principle computation if $\pi_1(0)$ and $\pi_2(0)$ given by the LP solution are used as adjoint initial conditions.

When the control variable enters linearly in the Hamiltonian, the singularity condition (for the problem at hand, $|\pi(k+1)B| \equiv 1$) may be expected to arise. This is true for both the discrete time and continuous time cases. In general, singular subarcs in the control function occur when the inequality

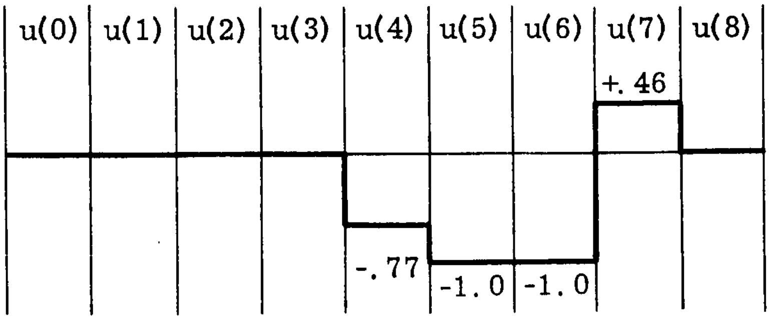
$$H(u_1^* \; u_2^* \; --- \; u_r^*) \leqq H(u_1^* + \delta u_1, \; ----- u_r^* + \delta u_r)$$

is met as an equality during a non-zero (finite) interval of time.[32] An appropriate analysis can show that this situation implies that $\frac{\partial H}{\partial u_i} = 0$ for a finite interval of time for some control component, indexed i. This situation was demonstrated in Section 3.2 and Figure 3-2 (c) for the continuous time case.

The singularity condition, $\frac{\partial H}{\partial u_i} = 0$ occurs more frequently in the discrete time case. Let us portray this situation heuristically.

COMPUTATION SUMMARY IV
TEST PROBLEM SECTION 5.2

(A) OPTIMAL CONTROL; LP SOLUTION; $y(0) = [111.1, 100]^T$



(B) TRAJECTORY USING OPTIMAL CONTROL

k	u(k)	$y_1(k)$	$y_2(k)$
0	0	+ 111.1	+ 100.0
1	0	+ 37.03	+ 50
2	0	+ 12.34	+ 25
3	0	+ 4.113	+ 12.5
4	- .776	+ 1.37	+ 6.25
5	-1.0	- .317	+ 2.35
6	-1.0	- 1.106	+ .175
7	+ .046	- 1.368	- .912
8	0	0	0

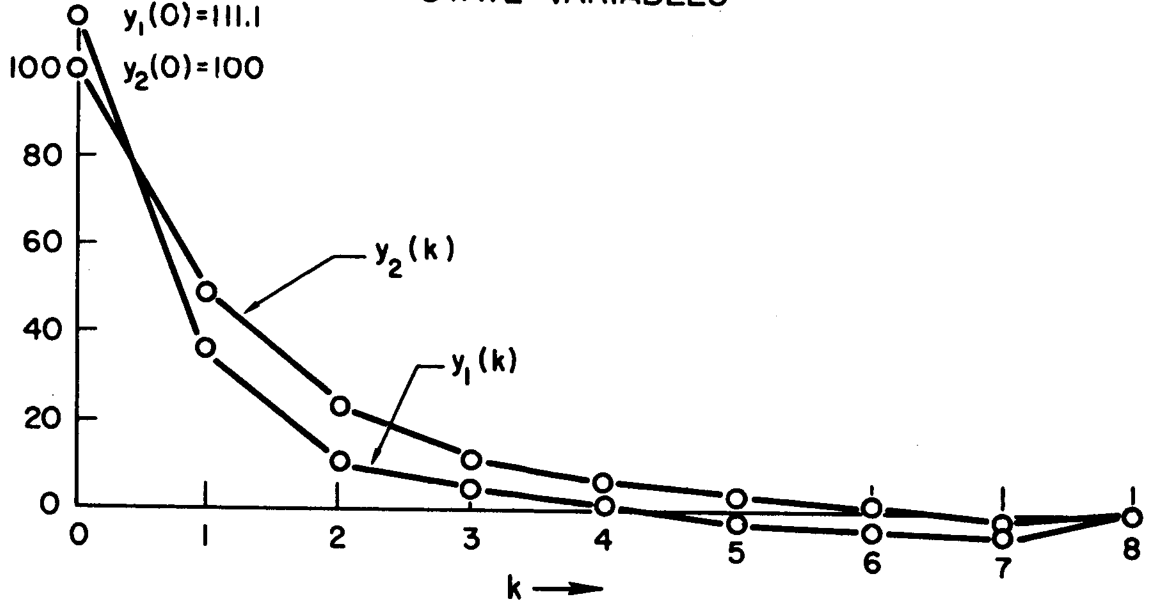
(C) DETERMINATION OF OPTIMAL CONTROL: MAX PRINCIPLE

k	$\pi_1(k)$	$\pi_2(k)$	$\pi(k)B$	MAX PRIN. u(k)	LIN PROG. u(k)
0	+ .00195	- 0.0460	< 1.0	0	0
1	.00585	- 0.0921	< 1.0	0	0
2	.01754	- 0.1842	< 1.0	0	0
3	.05262	- 0.3684	< 1.0	0	0
4	.15787	- 0.7368	< 1.0	UNDET.	- .776
5	.47361	- 1.4737	- 1.00	- 1.0	-1.00
6	1.42082	- 2.9474	- 1.526	- 1.0	-1.00
7	4.26246	- 5.8948	- 1.632	UNDET.	+0.46
8	12.7874	- 11.7896	+ 1.00	0	0

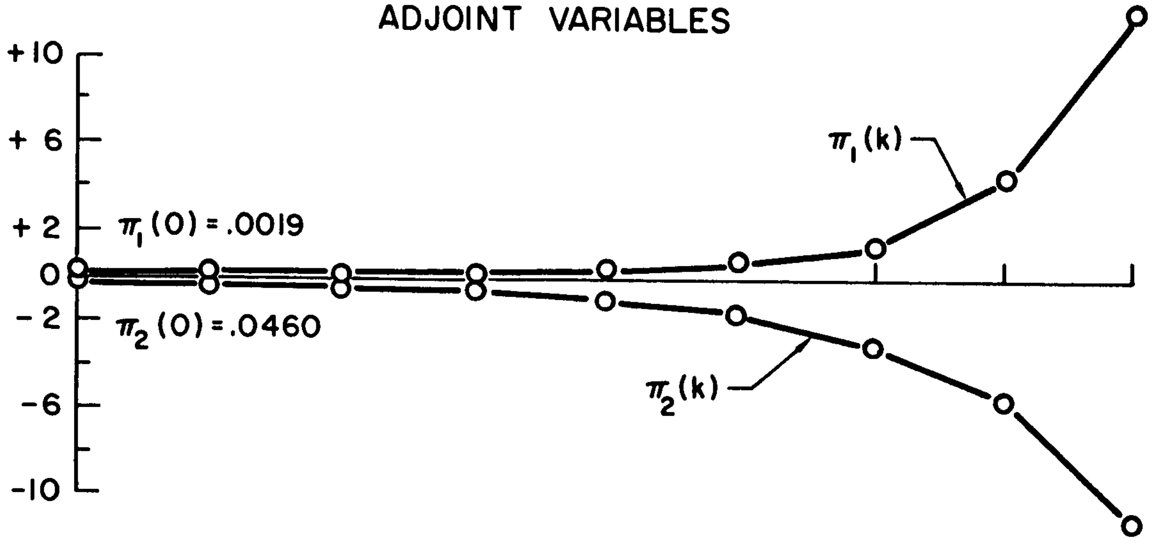
COMPUTATION SUMMARY IV (Continued)

(D) STATE AND ADJOINT VARIABLE PLOTS

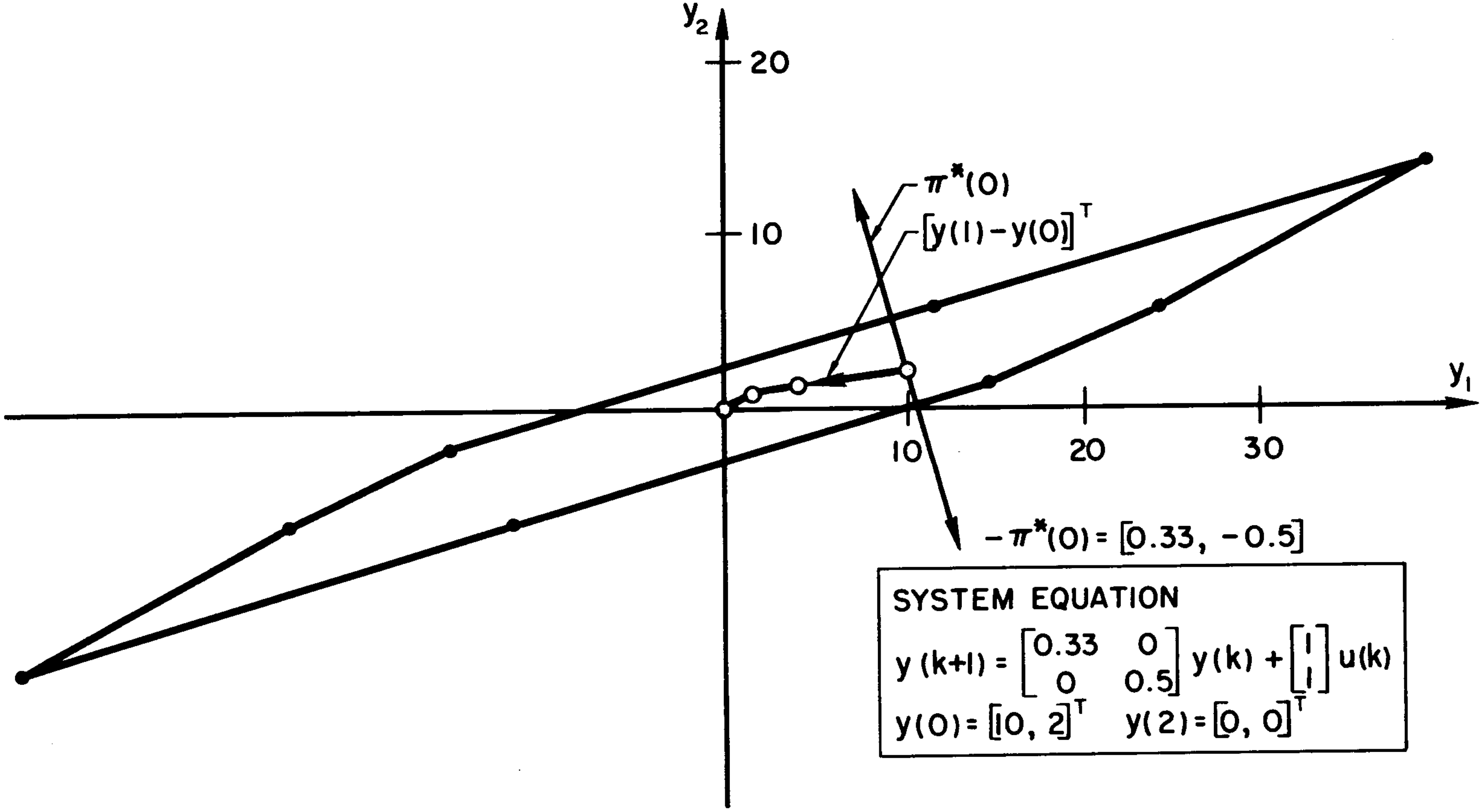
STATE VARIABLES



ADJOINT VARIABLES



END COMPUTATION SUMMARY IV



THE POLYHEDRON OF RECOVERABLE STATES FOR K = 3

FIGURE 5-2

Figure 5-2 depicts the polyhedron of recoverable states for the system associated with our prior illustrative problems. The trajectory for $K = 3$ and $y(0) = [10, 2]^T$ is sketched along with the optimal value of $\pi^*(0)$ computed in Section 4.3. The Hamiltonian, $H\left(u^*(0)\right)$ is the dot product of the vector $\left[y(1) - y(o)\right]^T$ and $\pi^*(0)$. A perturbation about $u^*(0) = 13/15$ (the optimal control value for $k = 0$ computed in Section 4.3) will not change this dot product. Thus $\frac{\delta H}{\delta u^*(0)} = 0$ for a finite Δt. This is the condition for singular control. The Maximum Principle will fail to provide enough information to compute the optimal control function in this instance. However, as has been shown, the LP approach will easily determine the required intermediate control magnitude.

5.3 Functional Analysis Approach

Referring to Equation (4.12) in Section 4.1, let the row vector of matrix [R] be designated

$$r_i = \left(r_{i0}, r_{i1}, r_{i2} \ldots r_{i,K-1}\right)$$

$$i = 1, 2, \ldots, m$$

The r_i's form a basis for a K-dimensional space over which we will define a functional f. This functional, specified by the vector $u = (u_0\ u_1 \ldots u_{K-1})$, is to have the property

$$f(r_i) = -y_i(0) \qquad i = 1, 2, \ldots, m \tag{5.15}$$

The Hölder inequality for sums (Kolmogorov and Fomin, Reference 33, Page 20) is introduced next. Given the above mentioned K-vector u, and any vector r in the subspace defined by $\{r_i \mid i=1, 2, \ldots, m\}$, we write the Hölder inequality as

$$f(r) = \sum_{k=0}^{K-1} r_k u_k \leq \sum_{k=0}^{K-1} |r_k u_k| \leq \left[\sum_{k=0}^{K-1} |r_k|^q\right]^{q^{-1}} \left[\sum_{k=0}^{K-1} |u_k|^p\right]^{p^{-1}} \quad (5.16)$$

where $p^{-1} + q^{-1} = 1$; $p, q \geq 1$. Note that here the k' s index the K components of the r and u vectors. Norms for the r and u vectors are defined as follows

$$\|r\|_q = \left[\sum_{k=0}^{K-1} |r_k|^q\right]^{q^{-1}} \quad (5.17)$$

$$\|u\|_p = \left[\sum_{k=0}^{K-1} |u_k|^p\right]^{p^{-1}} \quad (5.18)$$

When $p = 1$

$$\|u\|_1 = \sum_{k=0}^{K-1} |u_k| \quad (5.19)$$

which, by letting $u_k \equiv u(k)$, corresponds to fuel performance $P(K)$ defined in Section 4. 2, Equation (4. 19). For $p = 1$, q approaches infinity, and the corresponding (conjugate) norm for r is written,

$$\|r\|_\infty = \underset{\substack{k \\ k=0, 1, 2, \ldots, (K-1)}}{\text{Max}} |r_k| \quad (5.20)$$

At this point, the minimum fuel control problem, to which we addressed ourselves in Section 4. 2, can be stated in a functional analysis context:

> "Determine the functional f over the K-dimensional subspace generated by $\{r_i | i, = 1, \ldots, m\}$ such that $f(r_i) = - y_i(0)$, and further that $\|u\|_1$ is a minimum."

Before the solution to this problem is apparent, we must develop a relationship between f and $\|u\|$.

To accomplish this, we choose an arbitrary m-dimensional vector π and write the dot product

$$\pi\Big(-y(0)\Big) = \sum_{i=1}^{m} \pi_i \Big(-y_i(0)\Big) = \sum_{i=1}^{m} \pi_i\, f(r_i) \tag{5.21}$$

where π_i represents the i^{th} component of the row vector π. Since f is a linear functional

$$\pi\Big(-y(0)\Big) = \sum_{i=1}^{m} \pi_i\, f(r_i) = f\left(\sum_{i=1}^{m} \pi_i r_i\right) \tag{5.22}$$

where r_i is, as before, a K-vector, the i^{th} row of matrix R. If we refer to Equation (5.16), and consider

$$r = \sum_{i=1}^{m} \pi_i r_i \quad \text{or} \quad r_k = \sum_{i=1}^{m} \pi_i r_{ik}\,; \quad r = \Big(r_0, ---r_k, ---r_{K-1}\Big) \tag{5.23}$$

we can rewrite the Hölder Inequality for $f(r)$ Equation (5.16) as follows

$$\begin{aligned} f\left(\sum_{i=1}^{m} \pi_i r_i\right) &\leq \sum_{k=0}^{K-1} \sum_{i=1}^{m} \pi_i r_{ik} u_k \leq \sum_{k=0}^{K-1} \left| \sum_{i=1}^{m} \pi_i r_{ik} u_k \right| \\ &\leq \left[\sum_{k=0}^{K-1} \left| \left(\sum_{i=1}^{m} \pi_i r_{ik}\right)\right|^{q}\right]^{1/q} \left[\sum_{k=0}^{K-1} |u_k|^{p}\right]^{1/p} \\ &\leq \|r\|_q \|u\|_p \end{aligned} \tag{5.24}$$

The above implies

$$\|u\|_p \geq \frac{f(r)}{\|r\|_q} = \frac{\left|\pi\Big(-y(0)\Big)\right|}{\|r\|_q} \tag{5.25}$$

Thus $\|u\|_p$ is minimum when equality holds for Equation (5.25) and, collaterally, for the Hölder Inequality, Equation (5.16). However, since r depends on an arbitrary π, equality must also hold for the "worst case" i.e., when π is chosen so as to maximize f(r). Thus we require

$$\left[\|u\|_p\right]_{\text{Min}} = \underset{\pi}{\text{Max}} \frac{|\pi(-y(0))|}{\|r\|_q} \tag{5.26}$$

As in Kranc and Sarachic[22] we observe

$$\underset{\pi}{\text{Max}} \frac{|\pi(-y(0))|}{\|r\|_q} = \underset{\pi(-y(0))=1}{\text{Max}} \left[\frac{1}{\|r\|_q}\right] = \frac{1}{\underset{\pi(-y(0))=1}{\text{Min}} \|r\|_q} \tag{5.27}$$

Hence

$$\left[\|u\|_p\right]_{\text{Min}} = \frac{1}{\underset{\pi(-y(0))=1}{\text{Min}} \|r\|_q} \tag{5.28}$$

The same result is obtainable from another but similar viewpoint; it appears in the literature under the title "Krein's L Problem."[24]

With Equation (5.28), we have a means of finding $\left[\|u\|_p\right]_{\text{Min}}$. However, for the control problem, our interest lies in finding u itself. This is possible since, for the Hölder Inequality Equation (5.24) to be satisfied as an equality, it is sufficient that

$$u(k) \equiv u_k = C \left|\sum_{i=1}^{m} \pi_i r_{ik}\right|^{q-1} \text{signum}\left(\sum_{i=1}^{m} \pi_i r_{ik}\right) \text{ ; C a constant} \tag{5.29}$$

To satisfy Equations (5.16) and (5.24) as equalities we require:

$$C = \frac{1}{\left\{\left\|\sum_{i=1}^{m} \pi_i r_i\right\|_q\right\}^q} = \frac{1}{\left\{\|r\|_q\right\}^q} \tag{5.30}$$

Recall that our minimum fuel performance figure P is expressible as a p norm with p = 1, that is, as:

$$P = \|u\|_1 = \sum_{k=0}^{k-1} |u_k| \tag{5.31}$$

When p = 1, q approaches infinity. Then,

$$\|r\|_q = \left\| \sum_{i=1}^{m} \pi_i r_i \right\|_q = \underset{k}{\text{Max}} \left(\sum_{i=1}^{m} |\pi_i r_{ik}| \right) \tag{5.32}$$

q approaching infinity

r_{ik} the kth component of vector r_i

To summarize we rewrite Equations (5.28) and (5.31) as follows:

$$\text{MIN } P = \left[\|u\|_1 \right]_{\text{Min}} = \frac{1}{\underset{\pi \text{ -y(0) } = 1}{\text{Min}} \left\| \sum_{i=1}^{m} \pi_i r_i \right\|_q}$$

$$= \frac{1}{\underset{\substack{\pi \\ \pi \text{ -y(0) } = 1}}{\text{Min}} \left[\underset{k}{\text{Max}} \left(\sum_{i=1}^{m} |\pi_i r_{ik}| \right) \right]} \tag{5.33}$$

or

$$\text{MIN } P = \frac{1}{\underset{k}{\text{Max}} \left(\sum_{i=1}^{m} |\pi_i^* r_{ik}| \right)} \tag{5.34}$$

where π^* is the π which minimizes the denominator of Equation (5.33). The problem of finding the minimum fuel cost is therefore one of finding the π^* which minimizes,

$$\left\| \sum_{i=1}^{m} \pi_i r_i \right\|_q = \underset{\pi}{\text{Min}} \left[\underset{k}{\text{Max}} \left(\sum_{i=1}^{m} |\pi_i r_{ik}| \right) \right] \tag{5.35}$$

subject to

$$\pi\Big(-y(0)\Big) = 1$$

In some cases (e.g., when q = 2 p = 2), this minimization can be accomplished by calculus. In the case q approaches infinity, considerable ingenuity and insight must be brought to bear even in the simplest (e.g., m = 2, second order) case.

We shall resort to Linear Programming to provide a computational method for the high order "q → infinity, p = 1" case. Consider the problem posed by Equation (5.35) and assume we have found π^*. Note that r_{ik} in Equation (5.35) is the ikth element of matrix [R] of Equation (4.12). Equation (5.35) therefore requires that we find the maximum over k (the column index of [R]) of the function $\sum_{i=1}^{m} |\pi_i^* r_{ik}| = \pi^* r(k)$ where r(k) is the kth column vector of [R]. From tableau (4.23) of Section 4.2, we observe that there are 2K constraints

$$\left.\begin{matrix} -\pi^* r(k) \leq 1 \\ +\pi^* r(k) \leq 1 \end{matrix}\right\} \text{ or } |\pi^* r(k)| \leq 1 \qquad (5.36)$$

to the dual problem. Further u(k), $k \in \{0, 1, \ldots, (K-1)\}$, is a basic variable if

$$|\pi^* r(k)| = 1.0 = \operatorname*{Max}_{k} |\pi r(k)| \qquad (5.37)$$

This is precisely what Equation (5.35) asks us to do, i.e., find the optimum basic variables. In linear programming terms we can find these basic variables by using π^* to "price out" (see Dantzig Reference 6, Chapter IX) each column of the matrix [R]. In the functional analysis approach we do likewise after determining π^* subject to $\pi^*\Big(-y(0)\Big) = 1$. Hence the π^* found under LP and the functional analysis methods differ only by a constant factor equal to $P_{Min} = D_{Max}$; P and D are as defined in Section 4.2.

To clarify the notational complexities which may have arisen in our comparison of the functional analysis and LP computations, consider the following two-state variable plant where the single control affects both states simultaneously:

$$y(k+1) = A\, y(k) + B\, u(k)$$

$$A = \begin{bmatrix} 0.33 & 0 \\ 0 & 0.50 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad (5.38)$$

$$y(0) = \begin{bmatrix} 10 \\ 2 \end{bmatrix} \qquad y(K) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This is the same familiar system we have used before. The system is stable and, without control, approaches the final state $y(K) = 0$ asymptotically. Our problem is to find the minimum fuel control which will bring the plant to this equilibrium state exactly in a time interval KT, $T = 1$. Let $K = 3$. From Chapter IV we can write

$$-\, y(0) = \sum_{j=0}^{K-1} \begin{bmatrix} 0.333 & 0 \\ 0 & 0.5 \end{bmatrix}^{-1-j} \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(j)$$

or for $K = 3$

$$-\, y(0) = \left\{ \begin{bmatrix} 3 \\ 2 \end{bmatrix} u(0) + \begin{bmatrix} 9 \\ 4 \end{bmatrix} u(1) + \begin{bmatrix} 27 \\ 8 \end{bmatrix} u(2) \right\} \qquad (5.39)$$

We require P to be minimum where

$$P = \sum_{j=0}^{2} |u(j)|$$

Linear Programming Computation:

The linear programming solution is depicted in Computation Summary I of Chapter IV. Both primal and dual problems are solved in one computational pass of the "by hand" simplex method.

The solution and illustrative check are as follows:

Primal: $u_0^* = +13/15,\ u_1^* = 0,\ u_2^* = -7/15$

Dual: $\pi_1 = -1/3$ and $\pi_2 = 1$

Primal Cost: $P = 13/15 + 0 + 7/15 = 4/3$

Dual Cost: $$D = \pi_1\left(-y_1(0)\right) + \pi_2\left(-y_2(0)\right)$$

$$= -1/3\,(-10) + 1\,(-2) = 4/3$$

Optimality Check: $P = D = 4/3$

Functional Analysis Computation:

As outlined above, the functional analysis approach requires the following computation:

$$\pi\left(-y(0)\right) = \pi_1(-10) + \pi_2(-2)$$

$$r = \sum_{i=1}^{2} \pi_i r_i = \pi_1(3, 9, 27) + \pi_2(2, 4, 8) \qquad (5.40)$$

$$\|r\|_\infty = \left\| \sum_{i=1}^{2} \pi_i r_i \right\|_\infty = \underset{\substack{j \\ j=0,1,2}}{\text{Max}} \left(\sum_{i=1}^{m} |\pi_i r_{ij}| \right)$$

or

$$\|r\|_\infty = \text{Max}\left\{ |3\pi_1 + 2\pi_2|\,,\ |9\pi_1 + 4\pi_2|\,,\ |27\pi_1 + 8\pi_2| \right\} \qquad (5.41)$$

next, we must determine π^* where

$$\pi^* = \left\{ \pi \,\middle|\, \underset{\substack{\pi \\ \pi\ -\ y(0) = 1}}{\text{Min}} \quad \text{Max}\left[|3\pi_1 + 2\pi_2|,\ |9\pi_1 + 4\pi_2|\,,\ |27\pi_1 + 8\pi_2| \right] \right\} \qquad (5.42)$$

Kranc and Sarachik[22] refer to Chebychev approximation theory to solve this problem but they do not develop a method. Swiger[25] points out the fact that the calculus can be employed if the absolute value sign could be removed. This may be done when p or q is an even integer. For the case at hand, however, $p = 1$ and $q = \infty$; thus, the calculus approach is not possible. A Chebychev approximation method could be developed or, in the two state case, we can determine π^* by some judicious choice and application of ingenuity. Fortunately, the linear program solution can be applied easily and advantageously.

Consider the dual cost computed in Section 4.3

$$D = \pi_1^{LP}\left(- y_1(0)\right) + \pi_2^{LP}\left(- y_2(0)\right)$$

$$D = -\frac{1}{3}(-10) + 1\,(-2) = \frac{4}{3} \tag{5.43}$$

where π_1^{LP}, π_2^{LP} are the dual variables of the linear programming formulation. If we set

$$\pi_1 = \frac{\pi_1^{LP}}{D}\;,\; \pi_2 = \frac{\pi_2^{LP}}{D}$$

we can satisfy the constraint

$$\pi\left(- y(0)\right) = 1 \quad \text{by} \quad \frac{\pi^{LP}\left(- y(0)\right)}{D} = 1$$

Using this substitution

$$\pi_1^* = \frac{-1/3}{4/3} = -1/4\;,\; \pi_2^* = \frac{+1}{4/3} = +3/4 \tag{5.44}$$

Thus

$$\operatorname*{Max}_j \left(\sum_{i=1}^{2} \pi_i^* \, r_{ij}\right)$$

$$= \text{Max}\left[\left|-3/4 + 6/4\right|,\ \left|-9/4 + 12/4\right|,\ \left|-27/9 + 24/4\right|\right]$$

$$= 3/4 \tag{5.45}$$

Note that, in this case, the solution is not unique. We could have determined this from the LP solution also. Continuing

$$\text{MIN P} = \left[\|u\|_1\right]_{\text{Min}} = \frac{1}{\underset{j}{\text{Max}}\left(\sum_{i=1}^{2}|\pi_i^* r_{ij}|\right)} \tag{5.46}$$

whence

$$P = 4/3$$

This checks with the LP solution obtained by the simplex method.

To compute the $u^*(k)$, $k = 0, 1, 2$, we turn to Equations (5.29 and 5.30). Let

$$\left|\sum_{i=1}^{m}\pi_i r_{ij}\right| = \underset{k}{\text{Max}}\left|\sum_{i=1}^{m}\pi_i r_{ik}\right| \tag{5.47}$$

Then we may write Equations (5.29 and 5.30) as

$$u(j) = \frac{\left|\sum_{i=1}^{m}\pi_i r_{ij}\right|^{-1} \text{signum}\left(\sum^{m}\pi_i r_{ij}\right)}{\left|\sum_{i=1}^{m}\pi_i r_{ij}\right|^{-q}\left|\sum_{i=1}^{m}\pi_i r_{ij}\right|^{+q}\left[\sum_{\substack{k=0\\k\neq j}}^{K-1}\frac{\left|\sum_{i=1}^{m}\pi_i r_{ik}\right|^{q}}{\left|\sum_{i=1}^{m}\pi_i r_{ij}\right|^{q}} + 1\right]} \tag{5.48}$$

As q approaches infinity Equation (5.48) reduces to

$$u(j) = \frac{k_j}{\sum_{i=1}^{m}\pi_i r_{ij}} \qquad \text{for } j = \{J\}$$

$$u(j) = 0 \qquad \text{for } j \neq \{J\} \tag{5.49}$$

where $\{J\}$ is the set of all column indices for which Equation (5.47) holds. The set $\{J\}$ allows for the case when Equation (5.47) produces more than one $\left| \sum_{i=1}^{m} \pi_i r_{ik} \right|$ which has the same value as the maximum. Another requirement for the multiple maxima situation is

$$\sum_{j \in \{J\}} k_j = 1 \tag{5.50}$$

Applying this analysis to our numerical example, we note that Equation (5.45) produces three j's all with the same maximum. Hence $\{J\} = \{0, 1, 2\}$. As a result,

$$u_0 = \frac{k_0}{3/4}, \quad u_1 = \frac{k_1}{3/4}, \quad u_2 = \frac{k_2}{3/4}$$

To determine the k_j's solve the following equation set

$$\begin{aligned} -10(3/4) &= 3k_0 + 9k_1 - 27k_2 \\ -2(3/4) &= 2k_0 + 4k_1 - 8k_2 \\ 1 &= k_0 + k_1 + k_2 \end{aligned}$$

We obtain $k_0 = 13/20$, $k_1 = 0$ (arbitrary) and $k_2 = 7/20$. This yields

$$u_0 = +13/15 \text{ and } u_2 = -7/15$$

which again corresponds to the LP solutions of Section 4.3.

5.4 Generalized Programming Approach

This section illustrates the generalized programming method for the discrete time case. Here, the set of recoverable states Ω is a convex polyhedron and the solution can always be obtained in a finite number of iterations. This contrasts to the continuous time case in which the solutions obtained at each step of the generalized algorithm merely approach the optimal monotonically.

Consider the following problem. Given the same system as used in prior illustrations.

$$y(k+1) = A\,y(k) + B\,u(k) \qquad A = \begin{bmatrix} 0.33 & 0 \\ 0 & 0.50 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{5.51}$$

The initial state $y(0)$ is to be nulled in three time periods with the requirement that $z = \sum_{j=0}^{2} |u(j)|$ is to be minimized. Analogous to the definitions of Sections 2.4 and 3.4 we define

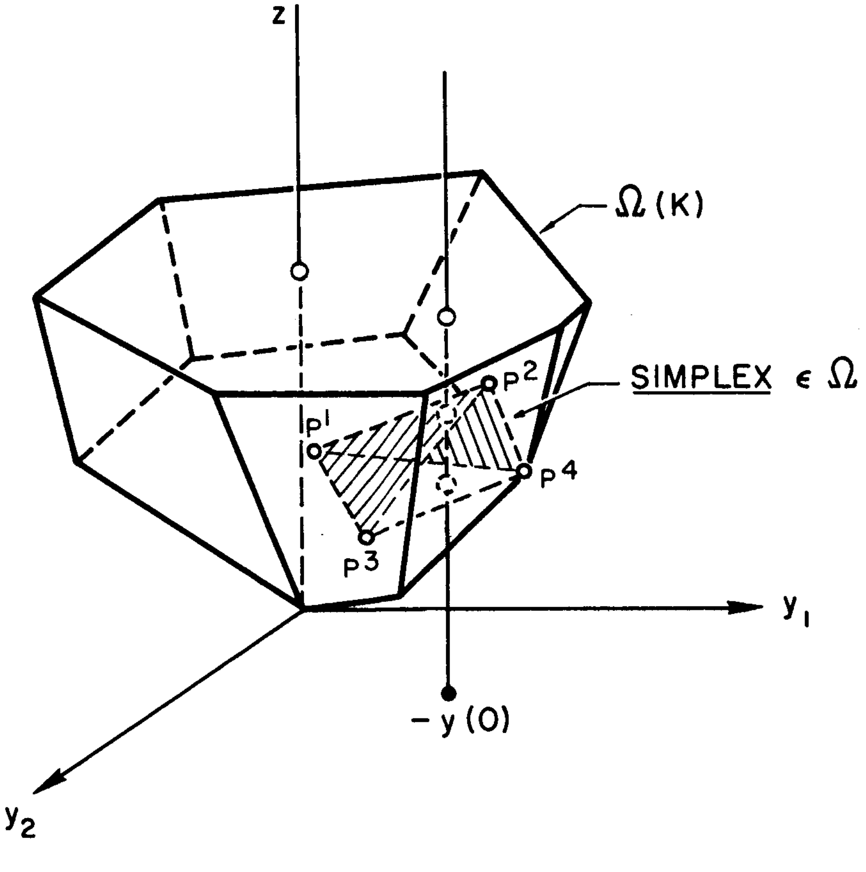
$$P = \left[\begin{array}{c} \sum_{j=0}^{K-1} A^{-1-j}\,B\,u(j) \\ \hdashline \sum_{j=0}^{K-1} |u(j)| \end{array}\right] \tag{5.52}$$

$$Q = \left[\begin{array}{c} -10 \\ -2 \\ \hdashline 0 \end{array}\right] \qquad U = \left[\begin{array}{c} 0 \\ 0 \\ \hdashline -1 \end{array}\right]$$

The minimum fuel problem can now be restated as:

$$\text{"Minimize } z \text{ such that } zU + P = Q\text{"} \tag{5.53}$$

As in Section 2.1 we take E_{m+1} to represent the extended requirements space containing, in this instance, the convex set (a polyhedron) of recoverable states, $\Omega(K)$. This is depicted in Figure 5-3. We shall call the line, which contains the requirement vector $-y(0)$, and which is parallel to the cost coordinate, the <u>Requirement Line</u>.



THE CONVEX POLYHEDRON Ω
CONTAINING A SIMPLEX

FIGURE 5-3

Assume that we select four allowable control functions $u^j(i)$, $i = 0, 1, 2, \ldots, (K-1)$; $j = 1, 2, 3, 4$. Using Equation (5.52) these generate the four column vectors P^j, $j = 1, 2, 3, 4$ represented in Figure 5-3. Their convex hull, a tetrahedron in Ω, is also shown. The initial P^j's may have all been extreme points of Ω. In Figure 5-3 only P^4 was assumed to be so.

In the ordinary simplex method we are given n P^j's at the start of the problem. For an m-dimensional requirement vector, m + 1 of the n P^j's define a hyperplane in E_{m+1}. This hyperplane contains a point of the requirement line if the P^j's are chosen so as to produce an initial feasible solution. If we generate one more P^j in addition to those defining the given hyperplane, we may form a simplex in E_{m+1}. For $m + 1 = 3$ the simplex is a tetrahedron shown in Figure 5-3. The simplex method operates to examine each plane in this simplex to discover which possesses the lowest point on the Requirement Line. The simplex method also determines which P^j to eliminate and which to add. We thus form a new simplex wherewith we can repeat the process. Simplex rules determine when the optimum solution is obtained. For a detailed description of this "simplex interpretation of the simplex method" the reader is referred to Dantzig (Reference 6, Chapter VII).

In the generalized programming method, we may begin the problem with only (m + 1) P^j's. Additional P^j's, unknown at first, are generated at some time later in the algorithm. Although the problem at hand may be more efficiently solved by the ordinary simplex method, it is instructive to solve it also by generalized programming.

We proceed by writing the vector P of Equation (5.52) as a convex combination of m + 1 linearly independent vectors P^j

$$P = \sum_{j=1}^{m+1} \lambda_j P^j \; ; \; \sum_{j=1}^{m+1} \lambda_j = 1 \; ; \; \lambda_j \geq 0 \tag{5.54}$$

Each P^j is generated by an allowable control $u^j(i)$, $i = 0, 1, 2, \ldots, K-1$ The control which produces the vector P is thus

$$u(i) = \sum_{j=1}^{m+1} \lambda_j u^j(i) \; ; \; \sum_{j=1}^{m+1} \lambda_j = 1 \; ; \; \lambda_j \geq 0 \tag{5.55}$$

The state diagram of Figure 5-4 may be used to ease the selection of the first three P^j's in order to create a satisfactory initial solution. It is clear that $-y(0)$ should lie in the convex hull of the P^j's so selected. The points ①, ②, and ③ of Figure 5-4 will serve our purpose. Their relevant properties are given in the following chart.

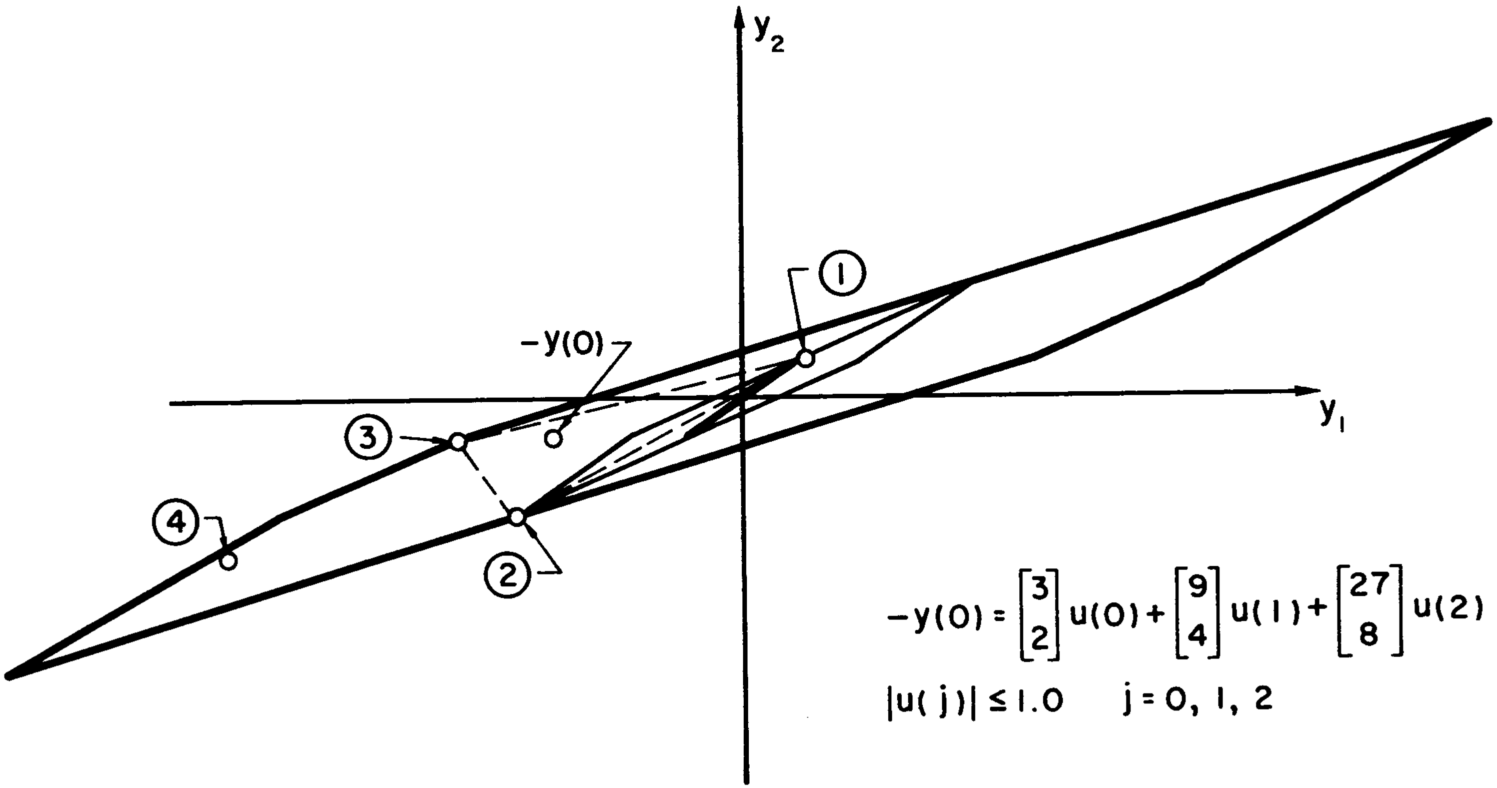
Point Number	Control Function	P^j
①	$u^1 = (1, 0, 0)$	$[3, 2, 1]^T$
②	$u^2 = (-1, -1, 0)$	$[-12, -6, 2]^T$
③	$u^3 = (1, 1, -1)$	$[-15, -2, 3]^T$

(5.56)

Since $zU + P = Q$ and for $j = 1, 2, 3$,

$$P = \sum_{j=1}^{3} \lambda_j P^j \; ; \; \sum_{j=1}^{3} \lambda_j = 1 \; ; \; \lambda_j \geq 1 \tag{5.57}$$

we may express the Master Problem in the following version of a "Tucker" diagram previously described in Sections 2.3 and 2.4.


y_2
①
$-y(0)$
y_1
③
④
②
$-y(0) = \begin{bmatrix} 3 \\ 2 \end{bmatrix} u(0) + \begin{bmatrix} 9 \\ 4 \end{bmatrix} u(1) + \begin{bmatrix} 27 \\ 8 \end{bmatrix} u(2)$
$|u(j)| \leq 1.0 \quad j = 0, 1, 2$

THE POLYHEDRON OF RECOVERABLE STATES FOR K = 3

FIGURE 5-4

	λ_1	λ_2	λ_3	z	e_1	e_2	e_3	e_4	
(-v)	1	1	1	0	1	0	0	0	1
π_1	3	- 12	- 21	0	0	1	0	0	- 10
π_2	2	- 6	- 2	0	0	0	1	0	- 2
π_3	1	2	3	- 1	0	0	0	1	0
	0	0	0	1	0	0	0	1	

(5.58)

This First Master Problem is solved in Computation Summary V to obtain the initial values of $\lambda = (\lambda_1, \lambda_2, \lambda_3, \; z)$ and $\pi = (-v \; \pi_2 \; \pi_3 \; \pi_4)$. A pivot reduction routine similar to the one performed in Computation Summary I, Chapter IV, yields both λ and π.

To see how the dual variable π is determined, apply Equation (4.31) to the problem represented in Tableau (5.58) above. We may write the dual of this Master Problem as

$$(-v, \pi_1, \ldots, \pi_{m+1}) \left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ \hline & & & 0 \\ & & & \\ & p_{ij} & & \\ & & & 0 \\ & & & -1 \end{array}\right] = (0, 0, 0, \ldots 0, 1) \qquad (5.59)$$

Designating the matrix in Equation (5.58) as [B], and postmultiplying both sides by $[B]^{-1}$ we determine that

$$(-v, \pi_1 \text{------} \pi_{m+1}) = \left(\hat{b}_{(m+1),1} \text{------} \hat{b}_{(m+1),(m+1)}\right)$$

(5.60)

where the $\hat{b}_{(m+1),j} \left(j = 1, 2, \ldots, (m+1)\right)$ are the elements in the last row of the matrix $[B]^{-1}$. For the problem at hand there are four rows.

COMPUTATION SUMMARY V
GENERALIZED PROGRAMMING EXAMPLE: SECTION 5.4

(A) FIRST MASTER PROBLEM:

	λ_1	λ_2	λ_3 —	λ_4	z	e_1	e_2	e_3	e_4	Q
e_1	1	1	1	1	0	1	0	0		1
e_2	3	- 12	- 15	- 27	0	0	1	0	0	- 10
e_3	2	- 6	- 2	- 8	0	0	0	1	0	- 2
e_4	1	2	3	1	- 1	0	0	0	1	0
λ_1	1	1	1	1	0	1	0	0	0	1
e_2	0	- 15	- 18	- 30	0	- 3	1	0	0	- 13
e_3	0	- 8	- 4	- 10	0	- 2	0	1	0	- 4
e_4	0	1	2	0	- 1	- 1	0	0	1	- 1
λ_1	1	0	- 1/5	- 1	0	4/5	1/15	0	0	2/15
λ_2	0	1	6/5	2	0	1/5	- 1/15	0	0	13/15
e_3	0	0	28/5	6	0	- 2/5	- 8/15	1	0	44/15
e_4	0	0	4/5	2	- 1	- 6/5	1/15	0	1	- 28/15
λ_1	1	0	0	- 11/14	0	11/14	1/21	1/28	0	5/21
λ_2	0	1	0	5/7	0	2/7	1/21	- 3/14	0	5/21
λ_3	0	0	1	15/14	0	- 1/14	- 2/21	5/28	0	11/21
z	0	0	0	20/7	1	8/7	- 1/7	1/7	- 1	16/7
						$-v$	π_1	π_2	π_3	

(B) SECOND MASTER PROBLEM:

	λ_1	λ_2	λ_3 —	λ_4	z	e_1	e_2	e_3	e_4	Q
λ_1	1	0	0	- 11/14	0	11/14	1/21	1/28	0	5/21
λ_2	0	1	0	5/7	0	2/7	1/21	- 3/14	0	5/21
λ_3	0	0	1	15/14	0	- 1/14	- 2/21	5/28	0	11/21
z	0	0	0	20/7	1	8/7	- 1/7	1/7	- 1	16/7
λ_1	1	11/10	0	0	0	11/10	1/10	- 1/5	0	1/2
λ_4	0	7/5	0	1	0	2/5	1/15	- 3/10	0	1/3
λ_3	0	- 3/2	1	0	0	- 1/2	- 1/6	1/2	0	1/6
z	0	- 4	0	0	1	0	- 1/3	1	- 1	4/3
						$-v^*$	π_1^*	π_2^*	π_3^*	

END COMPUTATION SUMMARY V

By referring to Computation Summary V we determine the First Master Problem solution as

$$\begin{aligned}\left(-v, \pi_1, \pi_2, \pi_3\right) &= \left(\frac{8}{7}, -\frac{1}{7}, \frac{1}{7}, -1\right) \\ \left(\lambda_1\ \lambda_2\ \lambda_3\ z\right) &= \left(\frac{5}{21}, \frac{5}{21}, \frac{11}{21}, \frac{16}{7}\right)\end{aligned} \qquad (5.61)$$

To continue our numerical procedure, note that

$$P^j = \begin{bmatrix} \sum_{i=0}^{2} A^{-1-i} B\, u^j(i) \\ \hdashline \sum_{i=0}^{2} |u^j(i)| \end{bmatrix} \qquad (j = 1, 2, 3) \qquad (5.62)$$

where A and B are defined as in Equation (5.51). According to the method of Sections 2.4 and 3.4, we are to maximize over u(i) the function

$$\pi P = \left(\pi_1\ \pi_2\right) \left[\sum_{i=0}^{2} A^{-1-i} B\, u(i)\right] - \sum_{i=0}^{2} |u(i)| \qquad (5.63)$$

The function πP in this case is separable, —the sum of three independent functions. Therefore it is sufficient to maximize each subfunction individually, that is find S_i where

$$S_i = \operatorname*{Max}_{u(i)} \left(\pi_1\ \pi_2\right)\left\{\left[A^{-1-i} B\, u(i)\right] - |u(i)|\right\} \qquad (5.64)$$

for i = 0, 1, 2. The above expression is to be considered a Sub-Problem which will generate another column for the Master Problem. Its correspondence with Equation (5.13) derived by the Maximum Principle should be noted. Equation (5.13) and Equation (5.64) are equivalent.

Continuing with the Generalized Programming procedure we compute Equation (5.64) for each i as follows

$$\begin{aligned} i = 0 &: \text{ maximize } \left(3\pi_1 + 2\pi_2\right) u(0) - |u(0)| \\ i = 1 &: \text{ maximize } \left(9\pi_1 + 4\pi_2\right) u(1) - |u(1)| \\ i = 2 &: \text{ maximize } \left(27\pi_1 + 8\pi_2\right) u(2) - |u(2)| \end{aligned} \tag{5.65}$$

resulting in

$$\begin{aligned} i = 0 &: \text{MAX } \{ - 0.14\, u(0) - |u(0)| \} = 0 \quad \text{for } u(0) = 0 \\ i = 1 &: \text{MAX } \{ - 0.71\, u(1) - |u(1)| \} = 0 \quad \text{for } u(1) = 0 \\ i = 2 &: \text{MAX } \{ - 2.72\, u(2) - |u(2)| \} = 1.72 \quad \text{for } u(2) = -1 \end{aligned} \tag{5.66}$$

Note that, as in Equation (5.13), the restriction $|u(i)| \leq 1$ must be used to avoid infinite magnitude impulses.

The solution to the Subproblem (5.64) calls for a new column to be added to the master program, that is, referring to table (5.56)

Point Number	Control Function	P_j
④	$u^4 = (0, 0, -1)$	$[-27, -8, 1]$

This column P^4 is generated from Equations (5.66) by the control $u(0) = 0$, $u(1) = 0$, $u(2) = -1$. The simplex method is used again to solve the Second Master Problem with the new column included, and an old one dropped. We can simplify the arithmetic by referring to the state diagram in Figure 5-4. Since $\lambda_j \geqq 0$ for all j, we drop that column which preserves this property and has the highest cost component. P^3 has the highest cost component but dropping it will cause the master solution to lose its non-negativity property. Hence we drop P^2 instead. This subterfuge is used only to ease hand computation. The machine algorithm employing the simplex method preserves non-negativity intrinsically.

Solution of the Second Master Problem yields the following values:

$$\left(-v, \pi_1, \pi_2, \pi_3\right) = \left(0, -\frac{1}{3}, 1, -1\right)$$

$$\left(\lambda_1, \lambda_4, \lambda_3, z\right) = \left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}, \frac{4}{3}\right)$$

A second attempt to solve the Subproblem $\left(\text{with } \pi_1 = -\frac{1}{3} \text{ and } \pi_2 = +1\right)$

$$S_i = \underset{u(i)}{\text{Max}} \left\{ \left(\pi_1 \; \pi_2\right) \left[A^{-1-i} B\, u(i)\right] - |u(i)| \right\}$$

for $i = 0, 1, 2,$ yields all $S_i \leqq 0$. Hence, no other column which yields lower fuel cost can be added. The algorithm terminates with the following solution.

$$u = \lambda_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \lambda_3 \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} + \lambda_4 \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

Hence $u^*(0) = 2/3$, $u^*(1) = 1/6$ and $u^*(2) = -1/2$. The fuel cost is $\sum_{i=0}^{2} |u(i)| = 4/3$. As noted before, the optimal control for this problem with $K = 3$ is not unique. Other solutions yielding this same fuel cost (but no lower) exist. Two of these were computed in Section 4. 3.

CHAPTER VI

APPLICATION TO ON-ORBIT MIDCOURSE AND TERMINAL CONTROL MANEUVERS

6.0 Introduction

There is a continuing need for application studies of optimal control theory, particularly to problems of missile and space vehicle guidance and control. Some results have been obtained in this problem area (where fuel consumption, payload, and mission time are especially critical) but much more remains to be done.

Near earth space operations may be divided into several mission phases as follows: launch, orbit injection, orbit transfer, midcourse and terminal maneuvering. For present purposes, the orbit transfer phase is considered to be one which transfers the spacecraft between earth orbits differing widely in associated energy – altitude differences of hundreds of nautical miles for example. After execution of such a transfer maneuver, the spacecraft' s velocity and position at a specified time may still differ from a desired position and velocity. In the terminal maneuver phase, we will assume this difference to be anywhere from 1 to 10 nautical miles, and one to three hundreds of feet per second, depending on a great many mission factors.

Midcourse and terminal maneuvers are consequently required to

(a) Bring the spacecraft to a pre-specified position, velocity and time state so that its subsequent free-fall trajectory follows a desired ephemeris.

(b) Match the position and velocity of another orbiting spacecraft in order to achieve a rendezvous or station-keeping condition.

Terminal maneuvers are also needed to maintain a desired ephemeris in the face of conservative and non-conservative perturbation forces.

This chapter is intended to demonstrate how the Linear Programming formulation of the "Minimum Fuel" problem can be applied to program the required thrust acceleration for terminal and midcourse manuevers of the type outlined in the previous paragraph. Pertinent features of this application include the following:

(a) Thrust is assumed to be obtained by fixed thrusters mounted orthogonally.

(b) Discrete time linearized equations of motion (of the Wiltshire-Clohessey type, Reference 34) are employed.

(c) A target centered coordinate system is postulated. In this system the linearized equations for the applicable relative distances are sufficiently accurate. For example, at $x = y = 50$ NM, the nonlinear terms amount to an accelerating force of approximately 0.01 ft/sec^2 in the x and y directions.[35]

(d) Fuel usage for discrete time control functions are minimized.

(e) Initial and target orbits may be circular or elliptical, and may involve plane changes within allowable thrust acceleration, maneuver time, and linearization ranges.

(f) The state determination (navigation) system is assumed to be sufficiently precise to make plausible a fuel minimization problem which is a deterministic.

6.1 Equations of Relative Motion

Let $\underline{R}_s$ be the position vector of a controlled satellite, and $\underline{R}_T$ be the position vector of either a "target" satellite, or a desired terminal position vector. These vectors are depicted in Figure 6-1, along with the coordinate system representation needed in what follows. Equations of motion may be written:

$$\ddot{\underline{R}}_T = \frac{-\mu}{R_T^3} \underline{R}_T + \underline{P}_T + \underline{F}_T \qquad (6.1)$$

$$\ddot{\underline{R}}_S = \frac{-\mu}{R_S^3} \underline{R}_S + \underline{P}_S + \underline{F}_S \qquad (6.2)$$

where $\underline{P}_T$, $\underline{P}_S$ are perturbation forces on the target satellite, and the controlled satellite respectively. $\underline{F}_T$ and $\underline{F}_S$ are thrust accelerations. Note that $\underline{F}_T$ and $\underline{P}_T$ may be set to zero if $\underline{R}_T$ is to represent a desired position, vector only.

Let $\underline{r}$ be the vector difference between the position vector of the controlled satellite and the desired position vector. Thus,

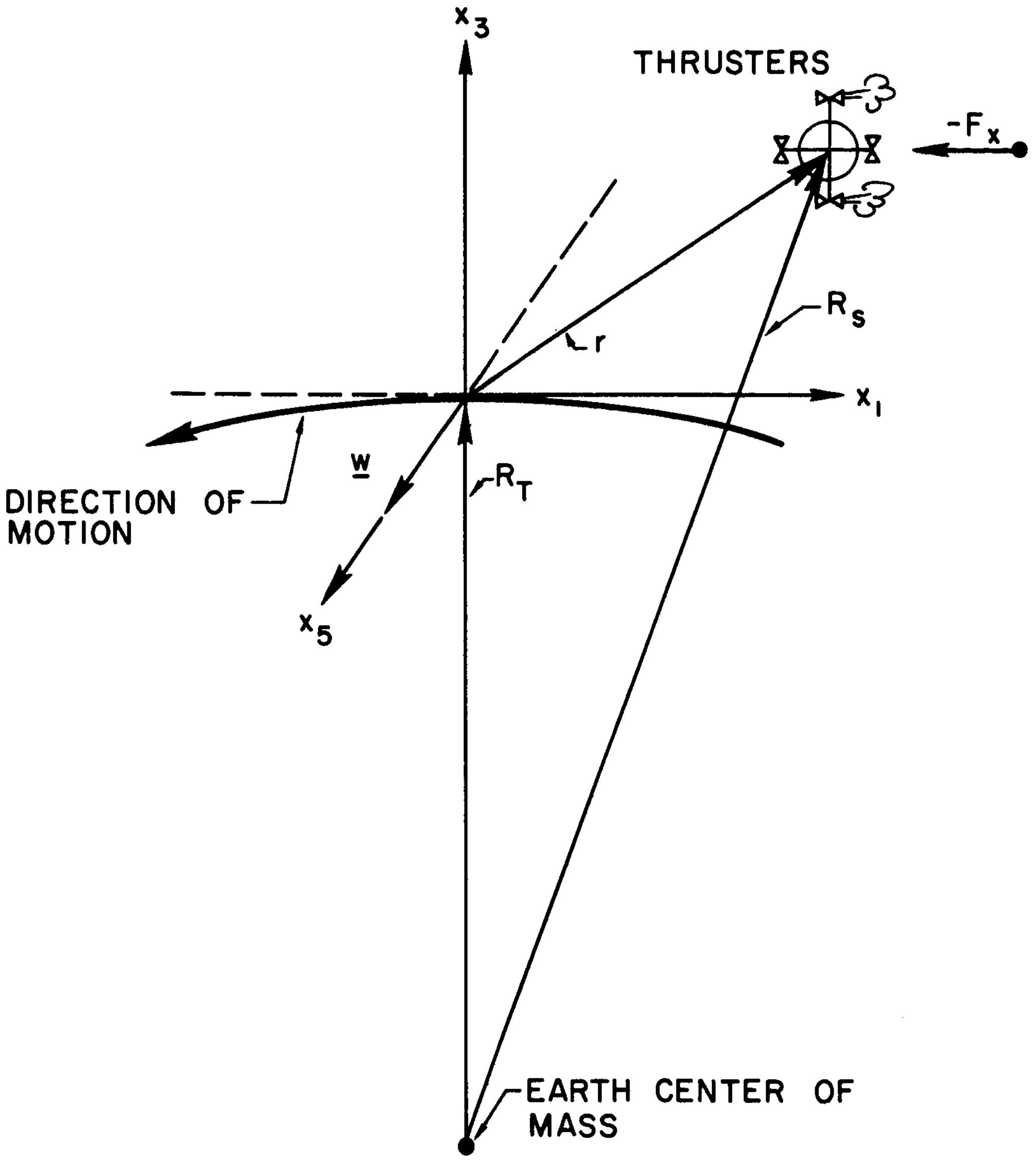
$$\underline{r} = \underline{R}_S - \underline{R}_T \qquad (6.3)$$

Using Equations (6.1), (6.2), and (6.3), the following expression may be written:

$$\ddot{\underline{r}} = -\mu \left[\frac{\left(\underline{R}_T + \underline{r}\right)}{\left[\left(\underline{R}_T + \underline{r}\right) \bullet \left(\underline{R}_T + \underline{r}\right)\right]^{3/2}} - \frac{\left(\underline{R}_T\right)}{\left[\left(\underline{R}_T\right) \bullet \left(\underline{R}_T\right)\right]^{3/2}} \right] +$$

$$+ \left(\underline{P}_S - \underline{P}_T\right) + \left(\underline{F}_S - \underline{F}_T\right) \qquad (6.4)$$

Equation (6.4) may be written as



COORDINATE SYSTEM ROTATING AT ORBITAL RATE $\underline{w}$

FIGURE 6-1

$$\underline{\ddot{r}} = \frac{-\mu}{R_T^3}\left\{\frac{\underline{r}}{\left[1+\left(2\underline{r}\bullet \underline{R}_T\right)/R_T^2+\left(r/R_T\right)^2\right]^{3/2}}\right.$$

$$\left. + \frac{\underline{R}_T - \left[1+\left(2\underline{r}\bullet\underline{R}_T\right)/R_T^2+\left(r/R_T\right)^2\right]^{3/2}\underline{R}_T}{\left[1+\left(2\underline{r}\bullet R_T^2\right)/R_T^2+\left(r/R_T\right)^2\right]^{3/2}}\right\}$$

$$+ \left(\underline{P}_S - \underline{P}_T\right) + \left(\underline{F}_S - \underline{F}_T\right) \qquad (6.5)$$

Next apply the algebraic expansion,

$$(1+a)^{-3/2} = 1 - \frac{3}{2}\, a + \frac{3.5}{2.4}\, a^2 - \cdots$$

to the denominators of the bracketed terms in Equation (6. 5) to obtain

$$\left[1+\left(2r\bullet R_T\right)/R_T^2+\left(r/R_T\right)^2\right]^{-3/2}$$

$$= 1 - \frac{3}{2}\left[\left(2\underline{r}\bullet\underline{R}_T\right)/R_T^2+\left(r/R_T\right)^2\right] + \cdots \qquad (6.6)$$

The bracketed term in Equation (6. 5) may then be written as

$$\left\{\underline{r}\left[1 - 3\,\cancelto{0}{\left(\underline{r}\bullet\underline{R}_T\right)/R_T^2} - \cancelto{0}{\frac{3}{2}\left(r/R_T\right)^2} + \cdots\right]\right.$$

$$\left. + \left[\underline{R}_T - \underline{R}_T - 3\underline{R}_T\left(\underline{r}\bullet\underline{R}_T\right)/R_T^2 - \frac{3\underline{R}_T}{2}\cancelto{0}{\left(r/R_T\right)^2} + \cdots\right]\right\}$$

Dropping all terms with factors of magnitude $\left(r/R_T\right) << 1$, we obtain the following approximation to Equation (6. 4) and (6. 5).

$$\ddot{\underline{r}} \cong \frac{-\mu}{R_T^3}\left[\underline{r} - \frac{3\underline{R}_T\left(\underline{r}\bullet \underline{R}_T\right)}{R_T^2}\right] + \left(\underline{P}_S - \underline{P}_T\right) + \left(\underline{F}_S - \underline{F}_T\right) \tag{6.7}$$

For $\underline{r}$ small compared to $\underline{R}_T$ or $\underline{R}_S$, $\left(\underline{P}_S - \underline{P}_T\right)$ may be assumed negligible. Furthermore, if no thrust is applied to the "target" satellite, $\underline{F}_T = 0$.

The state equations for the rendezvous control or orbit correction process will be based on the target centered rotating coordinate system depicted in Figure 6-1. Orbital rotation rate is $\underline{w}$. Unit vector $\underline{e}_z$ is parallel to $\underline{w}$, $\underline{e}_y$ is along $\underline{R}_T$ at all times and $\underline{e}_x$ is oriented to construct a right-handed orthogonal coordinate set. This relative coordinate system is target centered and rotates about $\underline{e}_z$ in inertial space at the target orbital rate $\underline{w}$.

Equations of motion and state in this coordinate system are obtained as follows:

$$\underline{w} = w(t)\,\underline{e}_z$$

$$\underline{r} = x\underline{e}_x + y\underline{e}_y + z\underline{e}_z$$

$$\dot{\underline{r}} = (\dot{x} - wy)\,\underline{e}_x + (\dot{y} + wx)\,\underline{e}_y + (\dot{z})\,\underline{e}_z \tag{6.8}$$

$$\ddot{\underline{r}} = (\ddot{x} - 2\,w\dot{y} - \dot{w}y - w^2x)\,\underline{e}_x + (\ddot{y} + 2w\dot{x} + \dot{w}y - w^2y)\,\underline{e}_y + (\ddot{z})\,\underline{e}_z$$

$$\underline{R}_T = R_T\underline{e}_y$$

$$\underline{r}\bullet\underline{R}_T = yR_T$$

Using these expressions in Equation (6.7) the following relations result:

$$\begin{aligned}
\ddot{x} - 2w\dot{y} - \dot{w}y - w^2x &= \frac{-\mu}{R_T^3}\, x + F_x \\
\ddot{y} + 2w\dot{x} + \dot{w}y - w^2y &= \frac{+2\mu}{R_T^3}\, y + F_y \\
\ddot{z} &= \frac{-\mu}{R_T^3}\, z + F_z
\end{aligned} \qquad (6.9)$$

Two of the three second order differential equations in Equations (6.9) are coupled. In general they are time varying. Note that the expression $(F_S - F_T)$ has been decomposed into its three components F_x, F_y and F_z. These represent three orthogonal thrusts per unit mass which are to be used to reduce the x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ to zero from some arbitrary initial condition.

When the target orbit is circular, w and R_T are constant. Further, $w^2 = \mu/R_T^3$ so that

$$\begin{aligned}
\ddot{x} - 2w\dot{y} &= F_x \\
\ddot{y} + 2w\dot{x} - 3w^2y &= F_y \\
\ddot{z} + w^2z &= F_z
\end{aligned} \qquad (6.10)$$

These are the rendezvous equations frequently attributed to Clohessy and Wiltshire.[34]

6.2 Differential Equations of State

Consider the first two equations of Equation (6.10). These can be used to represent in-plane maneuvers independent of the third out-of-plane equation in z. Set $x = x_1$, $\dot{x}_1 = x_2$, $y = x_3$, $\dot{x}_3 = x_4$, $F_x = u_1$ and $F_x = u_2$ to obtain:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2w \\ 0 & 0 & 0 & 1 \\ 0 & -2w & +3w^2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad (6.11)$$

In like manner, for the equation in z, set $z = x_5$, $\dot{x}_5 = x_6$ and $u_3 = F_z$ to obtain:

$$\begin{bmatrix} \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ w^2 & 0 \end{bmatrix} \begin{bmatrix} x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_3 \qquad (6.12)$$

The in-plane equations, Equations (9) are in the form,

$$\underline{\dot{x}} = F\underline{x} + B\underline{u}$$

where F and B are constant matrices. The characteristic matrix is obtained by the Laplace Transform method as follows:

$$\Phi(t) = L^{-1}\left\{\left[Is - F\right]^{-1}\right\}$$

where L^{-1} denotes the inverse Laplace Transform operation and I is the identity matrix. For reference purposes we record,

$$[Is - F]^{-1} = \begin{bmatrix} \frac{1}{s} & \frac{s^2 - 3w^2}{s^2(s^2+w^2)} & \frac{6w^3}{s^2(s^2+w^2)} & \frac{2w}{s(s^2+w^2)} \\ 0 & \frac{s^2 - 3w^2}{s(s^2+w^2)} & \frac{6w^3}{s(s^2+w^2)} & \frac{2w}{s^2+w^2} \\ 0 & -\frac{2w}{s(s^2+w^2)} & \frac{s^2+4w^2}{s(s^2+w^2)} & \frac{1}{s^2+w^2} \\ 0 & -\frac{2w}{s^2+w^2} & \frac{3w^2}{s^2+w^2} & \frac{s}{s^2+w^2} \end{bmatrix} \qquad (6.13)$$

from which we obtain:

$$\Phi(t) = \left[\begin{array}{c|c|c|c} 1 & \frac{4}{w} \sin wt - (3t) & 6wt - 6 \sin wt & \frac{2}{w} - \frac{2}{w} \cos wt \\ 0 & 4 \cos wt - (3) & 6w - 6w \cos wt & 2 \sin wt \\ 0 & \frac{2}{w} \cos wt - \left(\frac{2}{w}\right) & 4 - 3 \cos wt & \frac{1}{w} \sin wt \\ 0 & -2 \sin wt & 3w \sin wt & \cos wt \end{array}\right] \tag{6.14}$$

6.3 Difference Equations of Motion

With the derivations contained in Section 4.1 difference equations may be formulated from Equations (6.11 and 6.14). For the application at hand

$$y(k+1) = A\, y(k) + B\, u(k) \tag{6.15}$$

where

$$A = \Phi(T) = \left[\begin{array}{c|c|c|c} 1 & \frac{4}{w} \sin wT - (3T) & 6wT - 6 \sin wt & \frac{2}{w} - \frac{2}{w} \cos wT \\ 0 & 4 \cos wT - 3 & 6w - 6w \cos wt & 2 \sin wT \\ 0 & \frac{2}{w} \cos wT - \frac{2}{w} & 4 - 3wT & \frac{1}{w} \sin wT \\ 0 & -2 \sin wT & 3w \sin wT & \cos wT \end{array}\right] \tag{6.16}$$

$$B = \left[\begin{array}{c|c} -\frac{4}{w^2} \cos wT - \left(\frac{3T^2}{2}\right) + \left(\frac{4}{w^2}\right) & \frac{2T}{w} - \frac{2}{w^2} \sin wT \\ \frac{4}{w} \sin wT - \left(3T\right) & -\frac{2}{w} \cos wT + \left(\frac{2}{w}\right) \\ \frac{2}{w^2} \sin wT - \left(\frac{2T}{w}\right) & -\frac{1}{w^2} \cos wT + \left(\frac{1}{w^2}\right) \\ \frac{2}{w} \cos wT - \left(\frac{2}{w}\right) & \frac{1}{w} \sin wT \end{array}\right] \tag{6.17}$$

$$u(k) = \left[u_1(k) \;,\; u_2(k) \right]$$

The computer results to follow are based on a target state corresponding to a 300 NM circular orbit. The sampling interval is chosen to be T = 20 secs. With these parameters wT = .022 radians and

$$A = \left[\begin{array}{r:r:r:r} 1.000 & 19.994 & 0.000 & 0.438 \\ 0. & 0.999 & 0.000 & 0.044 \\ 0. & -\ 0.438 & 1.000 & 19.998 \\ 0. & -\ 0.044 & 0.000 & 1.000 \end{array}\right] \qquad (6.18)$$

$$B = \left[\begin{array}{r:r} 199.969 & 2.923 \\ 19.994 & 0.438 \\ -\ 2.923 & 199.992 \\ -\ 0.438 & 19.998 \end{array}\right] \qquad (6.19)$$

Various initial conditions $y(0) = \left[y_1(0),\ y_2(0),\ y_3(0),\ y_4(0) \right]^T$ are to be nulled so that $y(K) = \left[0,\ 0,\ 0,\ 0, \right]^T$ where

$y_1(0)$ = relative displacement in the instantaneous tangential direction (x_1 Figure 6-1)

$y_2(0)$ = relative tangential velocity component ($\dot{x}_1$)

$y_3(0)$ = relative displacement in the radial direction (x_3 Figure 6-1)

$y_4(0)$ = relative radial velocity component ($\dot{x}_3$)

From Section 4.1 we obtain

$$-\ y(0) = \sum_{j=0}^{K-1} A^{-1-j}\, B\, u(j) \qquad (6.20)$$

Let

$$R(j) = A^{-1-j} B \tag{6.21}$$

so that

$$R(j+1) = A^{-1} R(j) \tag{6.22}$$

The Linear Programming Problem (Primal) then becomes:

(a) Find $u(k) = \left[u_1(k),\ u_2(k)\right]^T$

(b) Subject to

$$-y(0) = \sum_{j=0}^{K-1} R(j)\, u(j)\ ;\ |u(j)| \leqq M$$

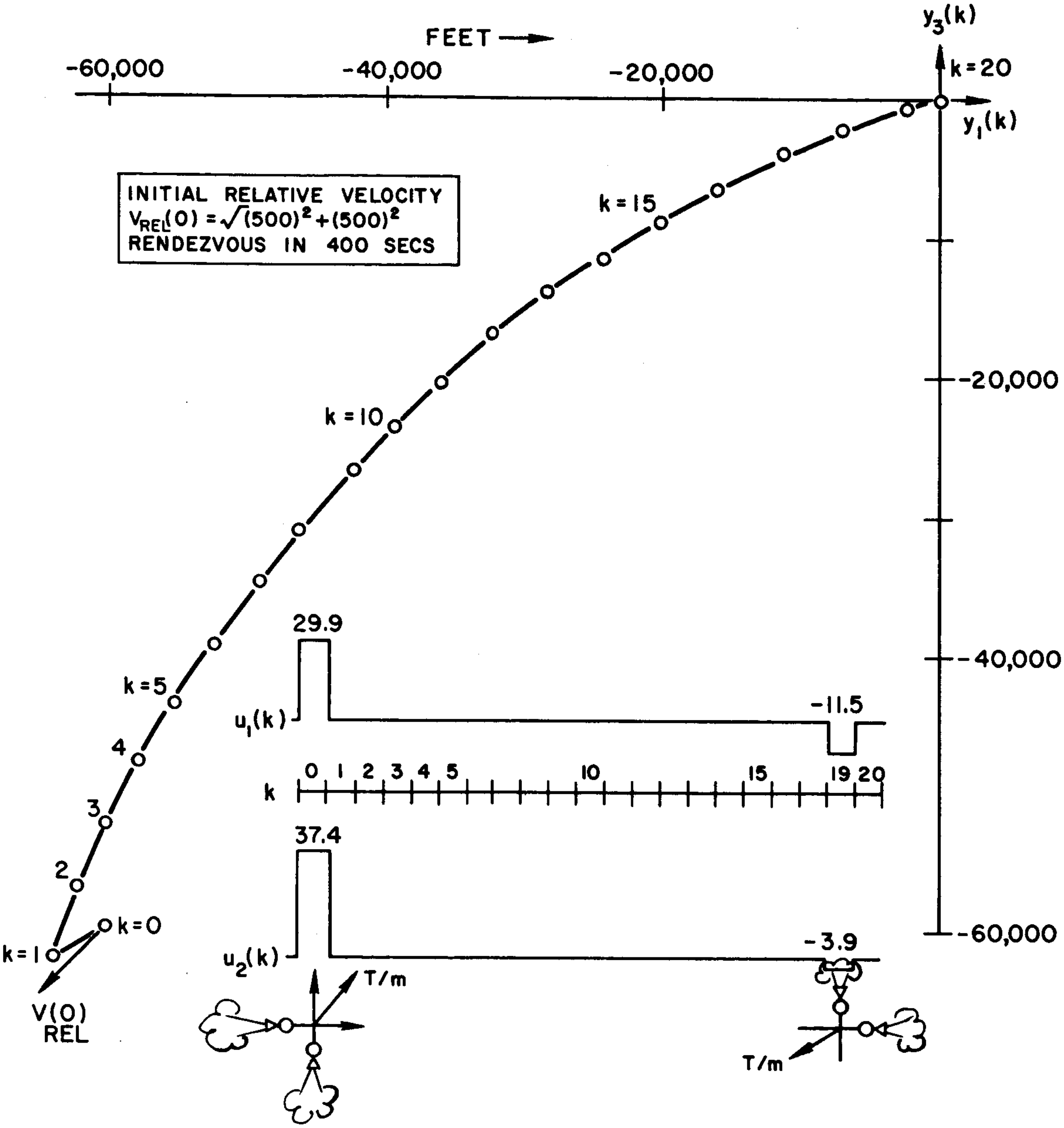
(c) To Minimize

$$\sum_{k=0}^{K-1} \|u(k)\| = \sum_{k=0}^{K-1} |u_1(k)| + |u_2(k)|$$

The results of using the LP/90 operating system for Linear Programming[36] are reported in the following section.

6.4 Computational Results

Figure 6-2 depicts a typical trajectory with relatively high initial relative velocity. The optimal acceleration time history was computed for a 400 second rendezvous time with sample time taken as 20 seconds. If the sample time were reduced, the pulses shown would be narrower but of higher amplitude. In the limit, the theoretical optimum impulse control history would result. A similar control history is also obtainable by the so called "orbital mechanics" approach to rendezvous analysis. Total ΔV required for this trajectory is computed as (20 secs.) times (29.9 + 11.5 + 37.4 + 3.9) ft/sec^2 or 1654 ft/sec. This represents a substantial ΔV expenditure for the type of terminal homing or station keeping application defined in Section 6.0.



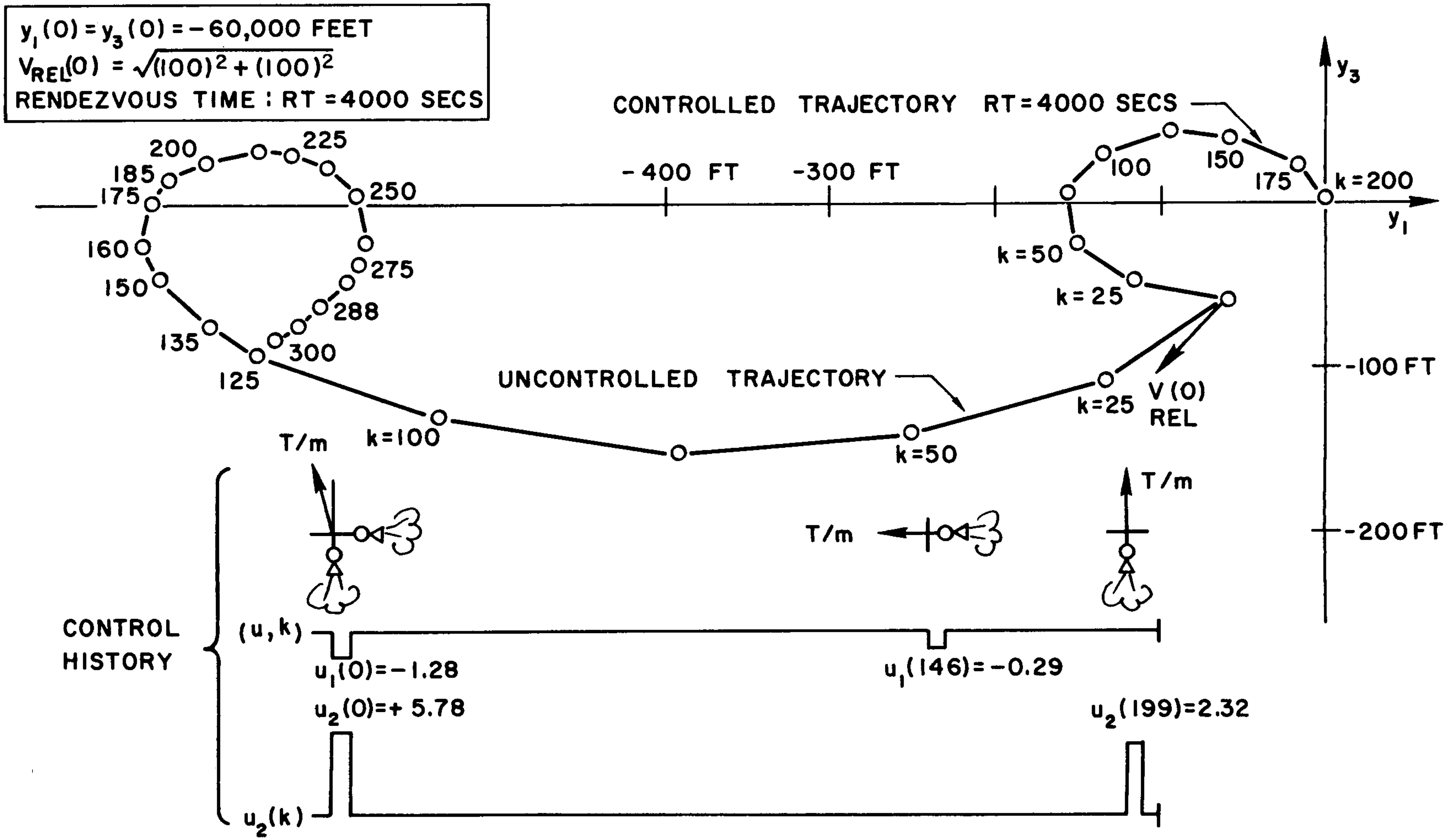
TYPICAL TRAJECTORY – HIGH RELATIVE VELOCITY

FIGURE 6-2

More realistic initial relative velocities are used for the sample trajectories of Figures 6-3 and 6-4. As rendezvous time is increased, better ΔV performance is achieved. This results from the fact that longer allowed time for rendezvous allows the "pursuer" satellite to wait for more favorable geometry before applying acceleration pulses. Also shown on Figures 6-4 and 6-5 are the relative motion of the "pursuer" spacecraft as seen from the target position for the given initial conditions with no applied control. The cycloidal motion is typical for initial relative velocities which have a y_1 directional component. If $y_1(0) = y_2(0)$ were zero, the idealized trajectory would be an ellipse.

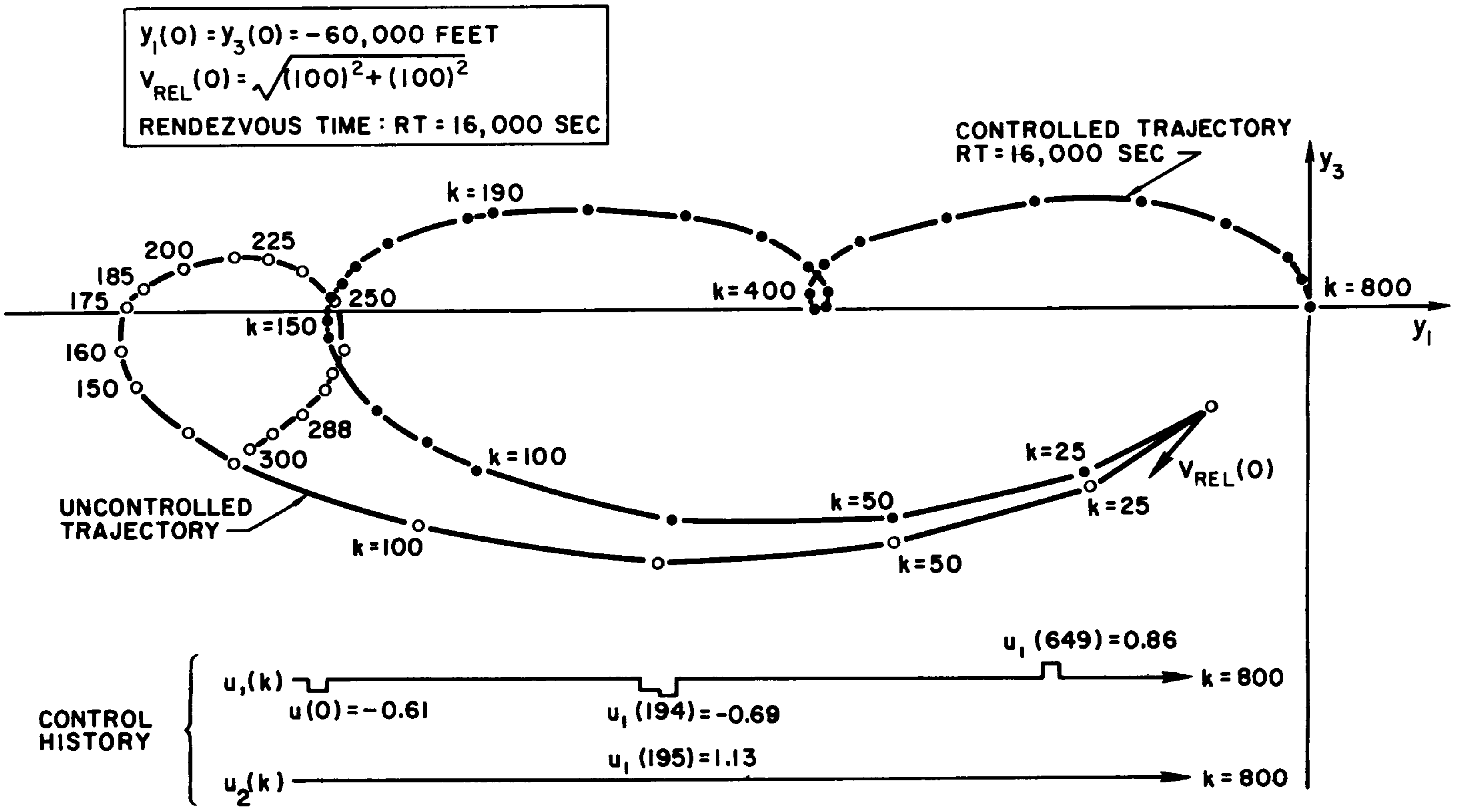
Figure 6-5 is included to indicate how performance index changes with allowed rendezvous time and initial relative velocity. It also serves to tabulate the series of trajectories successfully computed in the "unbounded" control case.

Bounded control computations were also made. These are graphed in Figure 6-7 along with the unbounded control histories. The characteristic bang-coast-bang nature of minimum fuel control is evident. As pointed out in Section 4.4, there is always an intermediate control pulse height at either end of the coast phase. This is characteristic of the discrete time solution. There is some reason to believe that, if this region of time were filled in with more LP "activity vectors," we could find the optimum switching time corresponding to the continuous time solution. This could be approached in the manner of Sakawa and Hayashi[5] who solved a minimum time problem using L. P. In the near future, Linear Programming codes will no doubt be available which will allow on-line insertion of such new "actvity" columns.



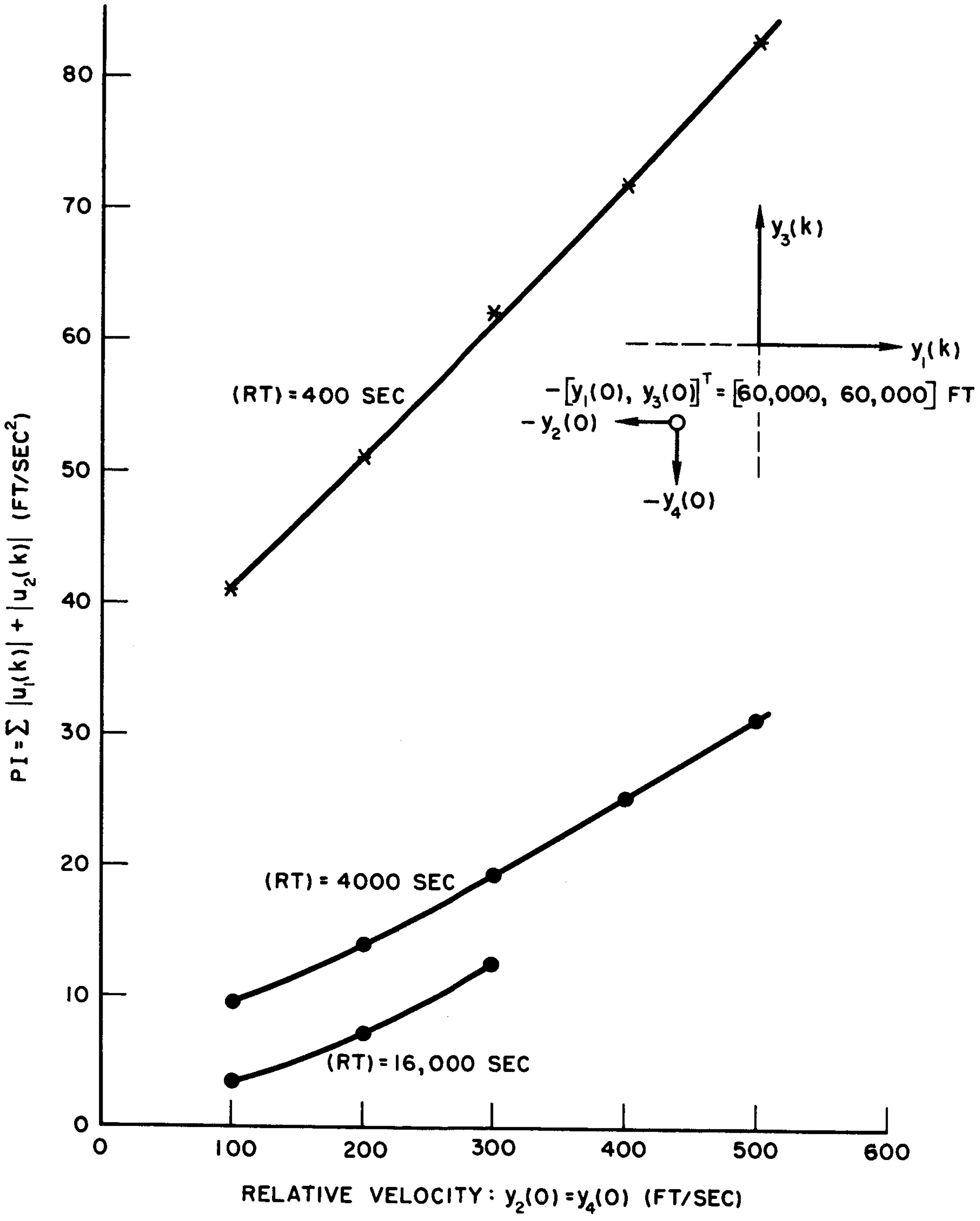
CONTROLLED AND UNCONTROLLED TRAJECTORIES I

FIGURE 6-3



CONTROLLED AND UNCONTROLLED TRAJECTORIES II

FIGURE 6-4

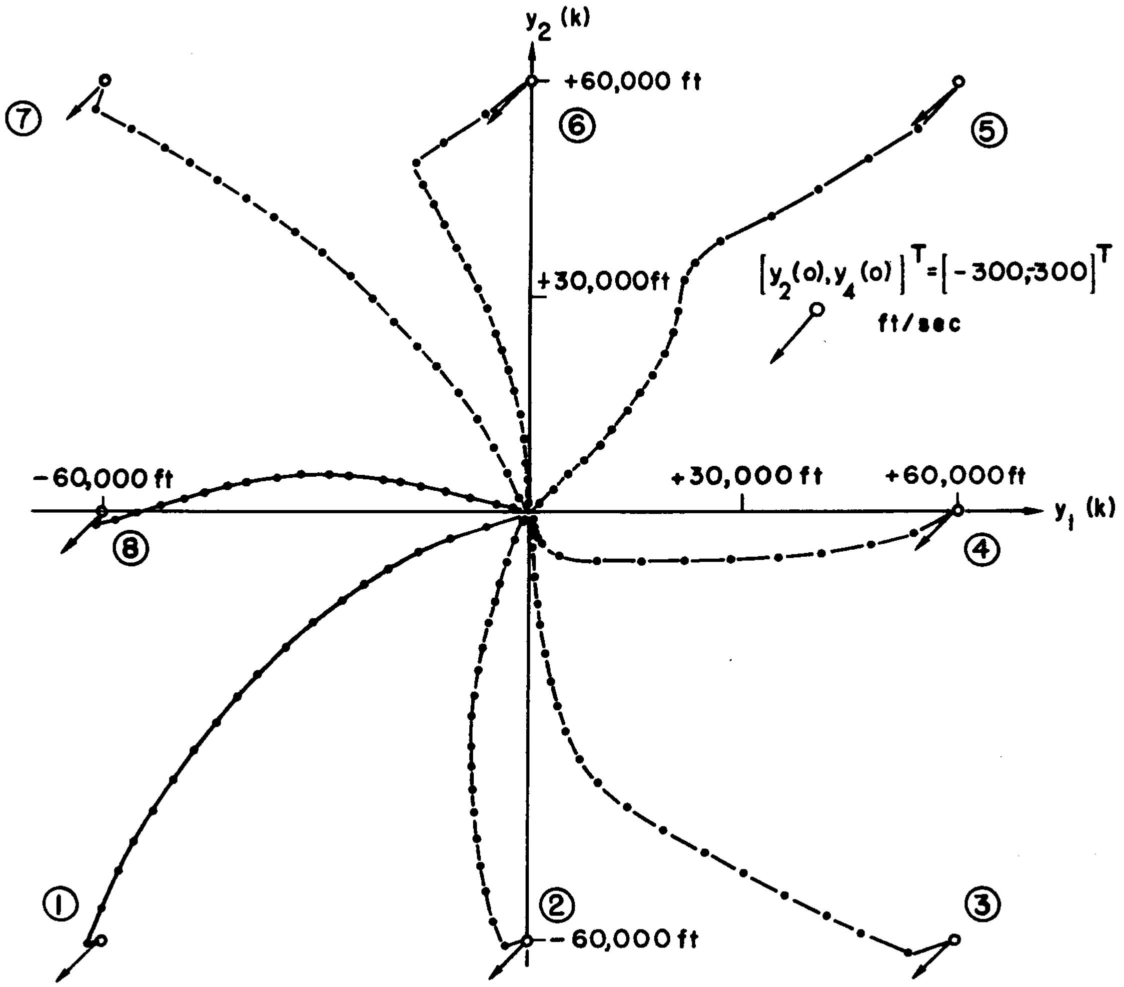


CHANGE OF PERFORMANCE INDEX WITH RELATIVE VELOCITY AND RENDEZVOUS TIME

FIGURE 6-5

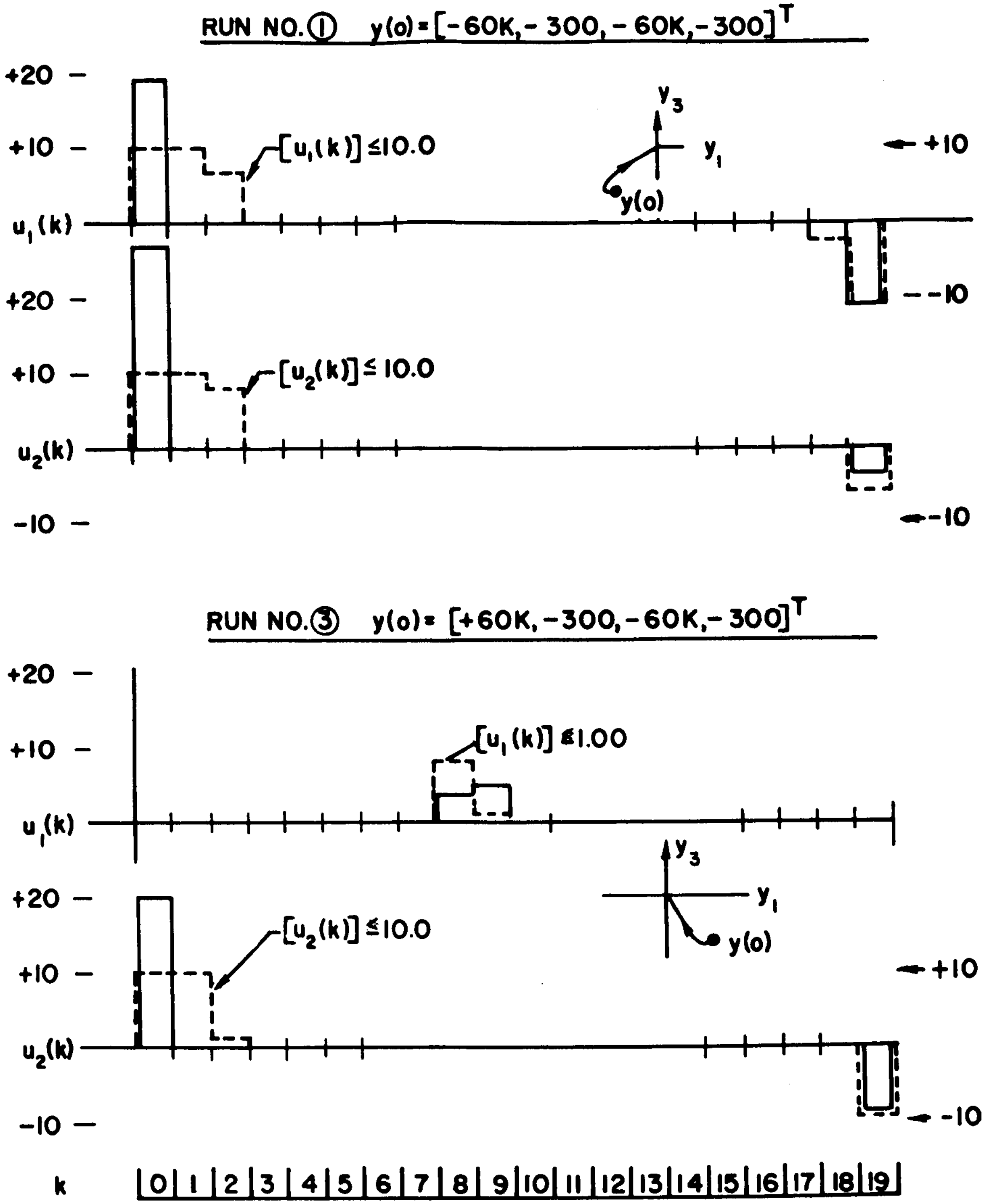
Figure 6-6 portrays the nature of the trajectories resulting from controls listed in Figure 6-7 and Computation Summary VI. Only the unbounded control trajectories were plotted in Figure 6-6. The bounded control trajectories are very similar, but with some of the corners of the plotted results more rounded.

To assist in any future comparisons between the LP solutions and those obtained by the maximum principle, the dual solutions of runs ① through ⑧ are listed in Computation Summary VIII. These may be used as initial conditions on the adjoint equations in the manner demonstrated in Section 5.2.



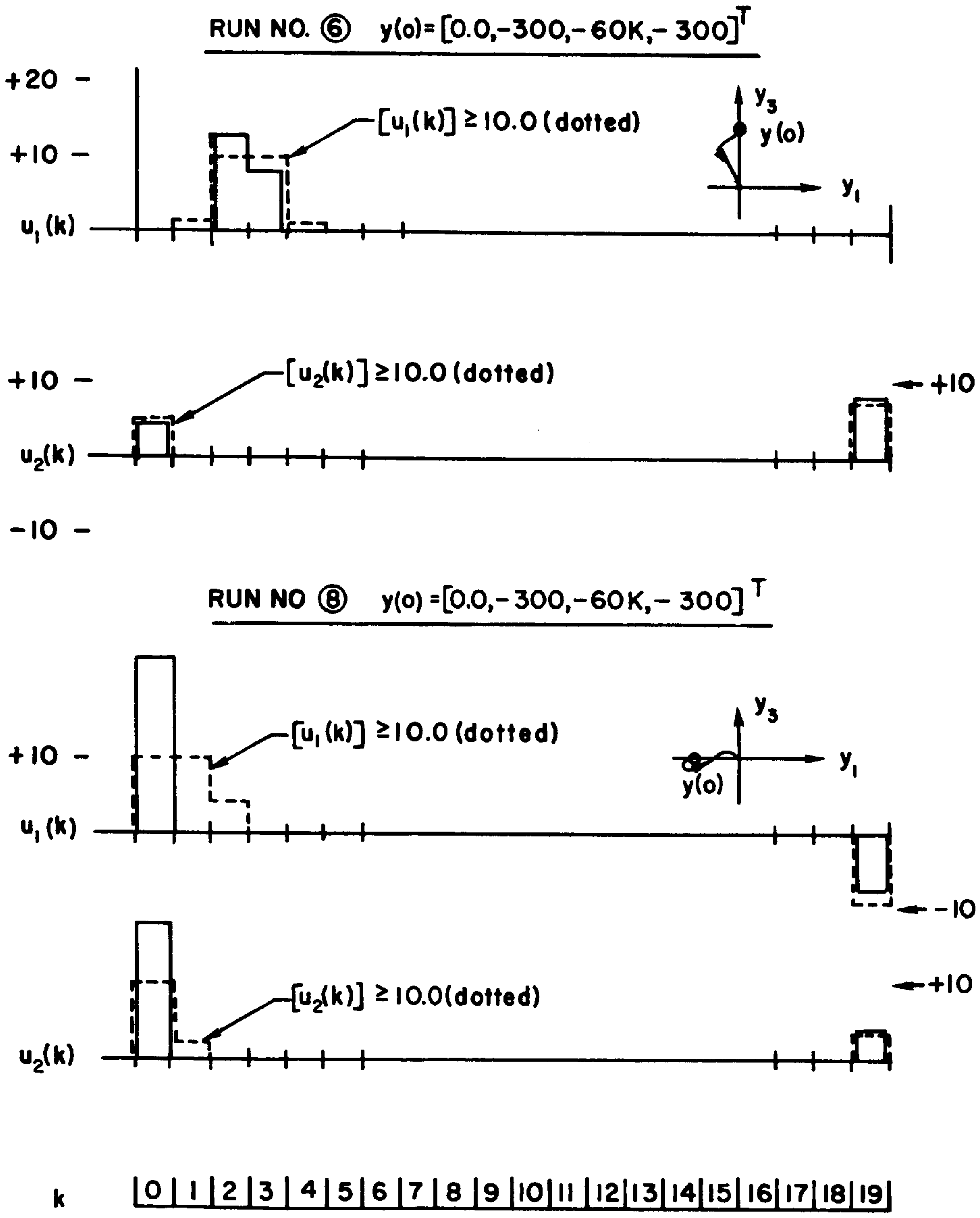
RENDEZVOUS TRAJECTORIES RENDEZ TIME 400 SECS. $V(0)_{REL} = \sqrt{(300)^2 + (300)^2}$

FIGURE 6-6



TYPICAL CONTROL-TIME HISTORY (400 SEC RENDEZ. TIME)

FIGURE 6-7



TYPICAL CONTROL-TIME HISTORY (400 SEC RENDEZ. TIME)
(Continued)

FIGURE 6-7

COMPUTATION SUMMARY VI

RESULTS OF CONTROL COMPUTATIONS: RENDEZVOUS TIME 400 SECS $[y_2(0), y_4(0)] = [-300, -300]$ ft/sec

RUN NO. ① $[y_1(0), y_3(0)] = [-60,000, -60,000]$ ft.

k →	1	2	3		19	20	COST (4)
(1) $u_1(k)$	19.56	0.0	0.0	—	0.0	-11.14	61.81584
$u_2(k)$	27.29	0.0	0.0	—	0.0	- 3.82	
(2) $u_1(k)$	19.72	0.0	0.0	—	0.0	-11.29	62.92885
$u_2(k)$	20.00	7.74	0.0	—	0.0	- 4.18	
(3) $u_1(k)$	10.00	10.00	0.79	—	-2.37	-10.00	66.73568
$u_2(k)$	10.00	10.00	8.68	—	0.0	- 4.90	

RUN NO. ② $[y_1(0), y_3(0)] = [0.0, -60,000]$ ft.

k →	1	2	3			20	COST
$u_1(k)$	12.11	0.0	0.0	—	—	3.68	46.90393
$u_2(k)$	24.14	0.0	0.0	—	—	- 6.98	
$u_1(k)$	12.19	0.0	0.0	—	—	- 3.77	47.53778
$u_2(k)$	20.00	4.39	0.0	—	—	- 7.18	
$u_1(k)$	10.00	2.64	0.0	—	—	- 4.21	49.99312
$u_2(k)$	10.00	10.00	5.27	—	—	- 7.87	

RUN NO. ③ $[y_1(0), y_3(0)] = [60,000, -60,000]$ ft.

k →	1	2	3	9	10	20	COST
$u_1(k)$	0.0	0.0	0.0	3.91	4.51	0.0	37.79935
$u_2(k)$	20.11	0.0	0.0	0.0	0.0	- 9.26	
$u_1(k)$	0.0	0.0	0.0	3.56	4.47	0.0	37.81221
$u_2(k)$	20.0	0.12	0.0	0.0	0.0	- 9.27	
$u_1(k)$	0.0	0.0	0.0	8.15	0.27	0.0	39.02481
$u_2(k)$	10.0	10.0	0.79	0.0	0.0	- 9.80	

COMPUTATION SUMMARY VI (Continued)

RUN NO. ④ $[y_1(0)\ ,\ y_3(0)] = [60{,}000\ ,\ 0.0]$ ft.

k →	1	2	9	10	11	20	COST
$u_1(k)$	0.0	0.0	3.15	11.85	0.0	0.0	27.33107
$u_2(k)$	10.55	0.0	0.0	0.0	0.0	-1.78	
$u_1(k)$	0.0	0.0	3.15	11.85	0.0	0.0	27.33107
$u_2(k)$	10.55	0.0	0.0	0.0	0.0	-1.78	
$u_1(k)$	0.0	0.0	4.18	10.0	0.82	0.0	27.39711
$u_2(k)$	10.0	0.59	0.0	0.0	0.0	-1.81	

RUN NO. ⑤ $[y_1(0)\ ,\ y_3(0)] = [60{,}000\ ,\ 60{,}000]$ ft.

k →	1	5	6	—	—	20	COST
$u_1(k)$	0.0	0.0	15.50	—	—	6.08	28.23317
$u_2(k)$	1.96	0.0	0.0	—	—	4.70	
$u_1(k)$	0.0	0.0	15.50	—	—	6.08	28.23317
$u_2(k)$	1.96	0.0	0.0	—	—	4.70	
$u_1(k)$	0.0	5.13	10.0	—	—	6.44	28.23357
$u_2(k)$	2.05	0.0	0.0	—	—	4.61	

RUN NO. ⑥ $[y_1(0)\ ,\ y_3(0)] = [0.0\ ,\ 60{,}000]$ ft.

k →	1	2	3	4	5	20	COST
$u_1(k)$	0.0	0.0	13.11	8.47	0.0	0.0	34.52447
$u_2(k)$	5.38	0.0	0.0	0.0	0.0	7.57	
$u_1(k)$	0.0	0.0	13.11	8.47	0.0	0.0	34.52447
$u_2(k)$	5.38	0.0	0.0	0.0	0.0	7.57	
$u_1(k)$	0.0	1.57	10.0	10.0	0.006	0.0	34.52520
$u_2(k)$	5.38	0.0	0.0	0.0	0.0	7.57	

COMPUTATION SUMMARY VI (Continued)

RUN NO. ⑦ $[y_1(0)\ ,\ y_3(0)] = [-\ 60{,}000\ ,\ 60{,}000]$ ft.

k →	1	2	3	19	—	20	COST
$u_1(k)$	26.41	0.0	0.0	0.0	—	-4.84	50.56013
$u_2(k)$	9.54	0.0	0.0	0.0	—	9.77	
$u_1(k)$	20.00	6.75	0.0	0.0	—	-5.18	51.22686
$u_2(k)$	9.40	0.0	0.0	0.0	—	9.90	
$u_1(k)$	10.00	10.00	7.70	0.0	—	-6.12	53.09934
$u_2(k)$	9.00	0.0	0.0	0.28	—	10.00	

RUN NO. ⑧ $[y_1(0)\ ,\ y_3(0)] = [-\ 60{,}000\ ,\ 0.0]$ ft.

k →	1	2	3	—	—	20	COST
$u_1(k)$	22.99	0.0	0.0	—	—	-7.99	52.36392
$u_2(k)$	18.42	0.0	0.0	—	—	2.97	
$u_1(k)$	20.00	3.15	0.0	—	—	-8.15	52.67456
$u_2(k)$	18.35	0.0	0.0	—	—	3.04	
$u_1(k)$	10.00	10.00	4.07	—	—	-9.07	54.61029
$u_2(k)$	10.00	8.53	0.0	—	—	2.94	

NOTES:

(1) $|u_1(k)|$ UNBOUNDED

(2) $|u_1(k)| \leq 20.0$

(3) $|u_1(k)| \leq 10.0$

(4) COST IS MEASURED IN T/M ft/sec^2

END COMPUTATION SUMMARY VI

COMPUTATION SUMMARY VII

INITIAL CONDITIONS FOR ADJOINT EQUATIONS

RUN NO.	$\pi_1(0) \times 10^3$	$\pi_2(0) \times 10^3$	$\pi_3(0) \times 10^3$	$\pi_4(0) \times 10^3$
①[1]	- 0.2485	- 51.3626	- 0.2566	- 53.6599
1[2]	- 0.2566	- 51.2738	- 0.2792	- 61.5178
1[3]	- 0.3039	- 58.5945	- 0.3066	- 71.0030
②	- 0.2485	- 51.3626	- 0.2566	- 53.6599
2	- 0.2566	- 51.2738	- 0.2792	- 61.5178
2	- 0.2723	- 53.9673	- 0.3013	- 70.3208
③	- 0.0107	- 40.6038	- 0.1745	- 52.6374
3	- 0.0171	- 40.0494	- 0.1924	- 58.4185
3	- 0.0243	- 39.4203	- 0.2127	- 64.9781
④	- 0.0107	- 40.6038	- 0.1745	- 52.6374
4	- 0.0107	- 40.6038	- 0.1745	- 52.6374
4	- 0.0114	- 39.0462	- 0.1919	- 58.3447
⑤	- 0.1045	- 49.7313	+ 0.0747	- 50.3314
5	- 0.1045	- 49.7313	+ 0.0747	- 50.3314
5	- 0.1045	- 49.7362	+ 0.0747	- 50.3315
⑥	- 0.1064	- 49.8932	+ 0.0742	- 50.3398
6	- 0.1064	- 49.8932	+ 0.0742	- 50.3398
6	- 0.1065	- 49.9179	+ 0.0743	- 50.3399
⑦	- 0.3535	- 52.4296	- 0.0301	- 51.4024
7	- 0.3665	- 57.7567	- 0.0241	- 51.4579
7	- 0.3847	- 63.9741	- 0.0104	- 51.4544
⑧	- 0.3535	- 52.4296	- 0.0301	- 51.4024
8	- 0.3665	- 57.7567	- 0.0241	- 51.4579
8	- 0.3840	- 63.6374	- 0.0264	- 54.6149

[1] $|u_i(k)|$ UNBOUNDED

[2] $|u_i(k)| \leq 20.0$

[3] $|u_i(k)| \leq 10.0$

END COMPUTATION SUMMARY VII

CHAPTER VII
SUMMARY AND EXTENSIONS

7.0 Summary

This investigation has resulted in defining the character of the solution to the minimum fuel optimal control problem obtained by means of linear programming. Preliminary data on the application of L. P. to orbital rendezvous has been obtained. The major difficulty in solving this type of application problem in its most general (nonlinear) form, lies in the fact that for the finite thrust case it reduces to a nonlinear two point boundary value problem. In other investigations, iterative gradient techniques have been successfully employed to obtain solutions for certain classes of this problem. However, convergence to the optimum is usually slow, particularly when the initial guess at a solution is a poor one. The results of the work reported here can be applied toward orbital rendezvous and station-keeping in at least the following four ways:

(a) As a means of rapidly obtaining a good initial solution to begin a more conventional gradient technique.

(b) As a means of performing sensitivity studies through the use of parametric linear programming.

(c) As a method of designing guidance equations for real time solution of orbital rendezvous and station-keeping by space-borne digital computers.

(d) As a means of solving the nonlinear problem resulting from a comprehensive mathematical model of the situation (which may include the higher harmonics of the earths gravity field). This would require the use of generalized programming which was introduced but not fully developed in Sections 2. 6, 3. 4, and 5. 4.

Another result of this investigation is the demonstration of the connection between dual variables of L. P. and the adjoint variables of the maximum principle. This should find use in easing the computational difficulties of applying the discrete maximum principle. The fact, demonstrated in Sections 5. 2, 5. 3, and 5. 4, that solutions can be relatively easily obtained by manual computation is worthy of note. A great amount of computer time can be saved by solving simple problems by hand, prior to machine computation of larger scale problems.

The use of the functional analysis approach to optimal control described by Kranc and Sarachik[22] has been hampered by the lack of easy computing methods in the case of L_1 and L_∞ norms. As shown in Section 5. 3, the L. P. approach relieves this problem considerably.

A new method of computing feedback coefficients has been proposed and demonstrated in a preliminary manner in Section 4. 5. This "closed loop" control synthesis technique could be applied to the orbital rendezvous guidance equation problem in cases where the full six dimensional problem can be decoupled into three two dimensional state equations. More work must be accomplished before this technique can be shown to offer any advantage.

It is hoped that this study will encourage further work toward making more efficient use of L. P. algorithms and concepts in optimal control theory applications. At some future time, the direction of effort should be in the development of machine codes specifically designed for optimal control theory usage. Instead of relying on existing codes, an algorithm which will take advantage of the specific structure of dynamic control theory problems should result in greater speed, accuracy and flexibility. Certain potential extensions will be discussed in the next section.

7.1 Extensions

Several areas of potentially profitable work using the concepts and methods of nonlinear and linear programming were identified in the course of this investigation. These will be described in brief under headings indicative of their connection with more well known approaches.

(a) Nonlinear Computations:

The generalized programming method seems to offer an attractive method for solving many nonlinear control problems previously approached only by iterative gradient techniques. As a follow-on to the present work, the author proposes that the generalized programming algorithm be tested for convergence and accuracy on the full nonlinear rendezvous problem.

(b) Differential Gaming:

As noted at the end of Chapter III, the column vector Q in the generalized programming technique can be considered as a representation of a larger convex set. This set could be thought of as those reachable states available to an evader. With some modification, the pursuit-evasion game can be computed using the technique of Section 3.4. The role played by the dual variables (pricing vector) in this application requires clarification.

(c) Game Theory:

It is known that symmetric matrix games can be formulated as linear programs and conversely. If this connection can be established between the linear programming formulation of the minimum fuel problem and an equivalent game matrix, useful ideas from both disciplines can be applied.

(d) Sensitivity Analysis:

Once an optimal L. P. solution has been obtained, the additional computation required to obtain the new optimal solution resulting from changes in cost coefficients right hand sides, or "A matrix" constants is relatively minor. Parametric studies can thus be readily performed. One such study would be to determine the change in fuel cost versus sampling time for the bounded control case.

(e) Upper and Lower Control Bounds:

For the rendezvous problem with spacecraft engines which are throttleable, the L. P. approach has one advantage not enjoyed by currently available methods. Throttleable engines usually have a threshold below which throttling cannot occur. Thus the allowable control region is between a lower as well as an upper bound (a region around zero being excluded). The allowable control range is consequently not convex. The L. P. formulation is readily adaptable to cope with this case.

(f) Stochastic Control:

Stochastic features of the rendezvous problem can be approached via "a two stage linear program under uncertainty" formulation. The work of Dantzig and Madansky[37] is referenced here. A two stage program contains constraints

$Ax + By = b, x \geq 0$ where A and B are known matrices;

b is a random vector (corresponding to the rendezvous initial conditions). It is desired to minimize with respect to x the function

$$E \min_{y} (cx + fy)$$

where c and f are known and E is an expectation operator taken over the distribution of b. At a given point in an orbit, the state vector b would be estimated and the solution to $Ax = b$ would be obtained and applied to the spacecraft orbit correction system. Because of errors in the initial estimation, and the imperfect nature of the control acceleration, the solution vector x must be corrected by the vector y to satisfy the constraints $Ax + By = b$ where b is the new state vector obtained only after application of the solution x. In addition to this approach, other work in stochastic linear programming has been started. To this date no codes have been developed to exploit the preliminary work which now exists.

REFERENCES

1. Athans, M. "The status of optimal control theory and applications for deterministic systems." IEEE Trans. Automatic Control, AC-11:580-596, July 1964.

2. Lee, Robert C. Optimal estimation, identification, and control. Cambridge, Massachusetts, The M.I.T. Press, 1964.

3. Whalen, Barry Hugh. Linear programming for optimal control. Ph.D. in Electrical Engineering, University of California, Berkeley, August 1962.

4. Torng, H.C. "Optimization of discrete control systems through Linear Programming." Journal of the Franklin Institute, 278:28-44, July 1964.

5. Sakawa, Y. and C. Hayashi. "Application of the Linear-Programming technique to the numerical solution of the optimal control problem." System Engineering for Control System Design. Proceedings. International Federation of Automatic Control, August 25-28, 1965, Tokyo, Science Council of Japan, 1965, p 40-49.

6. Dantzig, George B. Linear programming and extensions. Princeton, New Jersey, Princeton University Press, 1963.

7. Kuhn, H.W. and A.W. Tucker. "Nonlinear Programming." Second Berkeley Symposium on Mathematical Statistics and Probability. Proceedings. University of California, 1962, Berkeley, California, University of California Press, 1951. p 481-492.

8. Wolfe, P. "Recent developments in nonlinear programming." RAND Corporation, Report, R-401-PR, May 1962.

9. Everett, Hugh. "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources." Operations Research, 11:399-417, May-June 1963.

10. Brooks, R. and A. Geoffrion. "Finding Everett' s Lagrange multipliers by linear programming." RAND Corporation, Paper, P-3315, February 1966.

11. Neustadt, L.W. "Minimum effort control systems." Journal of SIAM, Series A on Control, 1:16-31, 1962.

12. Pontryagin, L.S., V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko. The mathematical theory of optimal processes. New York, Wiley, 1962.

13. Krasovskii, N.N. "On the theory of optimum control." Applied Mathematics and Mechanics, 23:899-919, 1959.

14. Dantzig, G. B. "Linear control processes and mathematical programming." SIAM Journal on Control, 4:56-60, February 1966.

15. Bellman, R.I. and O. Gross. "On the Bang-Bang control problem." Quarterly of Applied Mathematics, 13:321-324, 1956.

16. Meditch, J.S. "On minimal fuel satellite attitude controls." Fourth Joint Automatic Control Conference. Proceedings. University of Minnesota, June 19-21, 1963, Minneapolis, Minnesota, American Institute of Chemical Engineers, 1963. pp 558-563.

17. Meditch, J.S. and L.W. Neustadt. "An application of optimal control to midcourse guidance." Aerospace Corporation, Technical Report, A62-1732.2-5, July 1962.

18. Leondes, Cornelius T. Advances in control theory. New York, Academic Press, 1965. Vol. II, Chapter IV.

19. Paiewonsky, B.H. and others. "A study of synthesis techniques for optimal controllers." Flight Dynamics Laboratory Wright-Patterson Air Force Base, Technical Report, ASD-TDR-63-239, June 1964.

20. Athanassiades, M. "Optimal control for linear time-invariant plants with time, fuel, and energy constraints." IEEE Trans. Applied Industry, 81:321-325, January 1963.

21. Roxin, E. "A geometric interpretation of Pontryagin's Maximum Principle." Air Force Office of Scientific Research, Report, AFOSR 1751, December 1961.

22. Kranc, G.M. and P.E. Sarachik. "An application of functional analysis to the optimal control problem." Journal of Basic Engineering, Tans. Am. Soc. Mech. Engr., 85:143-150, June 1963.

23. Neustadt, L.W. "Optimization, a moment problem, and nonlinear programming." Journal of SIAM, Series A on Control, 2:33-53, 1964.

24. Akhiezer, N. and M. Krein. Some questions in the theory of moments. Providence, Rhode Island, American Mathematical Society, 1962.

25. Swiger, James Morris. The application of the theory of minimum normed operators to optimum control system problems. Ph. D. in Engineering, University of California, Los Angeles, January 1965.

26. Van Slyke, Richard Maurice. Mathematical programming and optimal control. Ph. D. in Engineering Science, University of California, Berkeley, 1965.

27. Gass, Saul I. Linear programming, methods and applications. New York, McGraw-Hill, 1964.

28. Lindorff, David P. Theory of sampled-data control systems. New York, Wiley, 1965.

29. Kleinman, D. L. and M. Athans. "The discrete minimum principle with application to the linear regulator problem." MIT Electronic Systems Laboratory, Technical Report, ESL-R-260, February 1966.

30. Jordan, B. W. and E. Polak. "Theory of a class of discrete optimal control systems." Journal of Electronics and Control, 17:697-711, December 1964.

31. Jackson, R. and F. Horn. "On discrete analogues of Pontryagin' s Maximum Principle." International Journal of Control, 4:389-396, 1965.

32. Kopp, R. E. and H. G. Moyer. "Necessary conditions for singular extremals." AIAA Journal, 3:1439-1444, August 1965.

33. Kolmogorov, A. N. and S. V. Fomin. Elements of the theory of functions and functional analysis. Rochester, New York, Graylock Press, 1957.

34. Clohessy, W. H. and R. S. Wiltshire. "Terminal System for Satellite Rendezvous." Journal of the Aerospace Science, 27:653-658, September 1960.

35. Harrison, Eugene. Minimum energy space rendezvous. Ph. D. in Mechanical Engineering, Michigan State University, East Lansing, June 1962.

36. "LP/90 Usage Manual" C-E-I-R, Inc. Farragut Square South, Washington, D. C. 2nd Edition, July 31, 1963.

37. Dantzig, George B. and Albert Madansky. "On the solution of two stage linear programs under uncertainty." Fourth Berkeley Symposium on Mathematical Statistics and Probability. Proceedings. University of California, Berkeley, pp 165-176, 1961.