THE APPLICATION OF OPTIMAL CONTROL THEORY TO DYNAMIC ROUTING IN DATA COMMUNICATION NETWORKS

by

FRANKLIN HOWARD MOSS

B.S.E., Princeton University
1971
S.M., Massachusetts Institute of Technology
1972

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1977

Signature of Author...
Department of Aeronautics and Astronautics
November 24, 1976

Certified by..... ....
Thesis Supervisor

Certified by...
Thesis Supervisor

Certified by...... ....
Thesis Supervisor

Accepted by...
Chairman, Departmental Graduate Committee

Archives
MASS. INST. TECH.
JUN 1 1977
LIBRARIES

THE APPLICATION OF OPTIMAL CONTROL THEORY
TO DYNAMIC ROUTING IN DATA COMMUNICATION NETWORKS

by

FRANKLIN HOWARD MOSS

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

ABSTRACT

The flow of messages in a message-switched data communication network is modeled in a continuous dynamical state space. The state variables represent message storage at the nodes and the control variables represent message flow rates along the links. A deterministic linear cost functional is defined which is the weighted total message delay in the network when we stipulate that all the message backlogs are emptied at the final time and the inputs are known. The desired minimization of the cost functional results in a linear optimal control problem with linear state and control variable inequality constraints.

The remainder of the thesis is devoted to finding the feedback solution to the optimal control problem when all the inputs are constant in time. First, the necessary conditions of optimality are derived and shown to be sufficient. The pointwise minimization in time is a linear program and the optimal control is seen to be of the bang-bang variety. Utilizing the necessary conditions it is shown that the feedback regions of interest are convex polyhedral cones in the state space. A method is then described for constructing these regions from a comprehensive set of optimal trajectories constructed backward in time from the final time. Techniques in linear programming are employed freely throughout the development, particularly the geometrical interpretation of linear programs and parametric linear programming.

There are several properties of the method which complicate its formulation as a compact algorithm for general network problems. However, in the case of problems involving networks with single destinations and all unity weightings in the cost functional it is shown that these properties do not apply. A computer implementable algorithm is then detailed for the construction of the feedback solution.

Thesis Supervisor: Adrian Segall
Title: Assistant Professor Electrical Engineering

Thesis Supervisor: Wallace E. Vander Velde
Title: Professor of Aeronautics and Astronautics

Thesis Supervisor: Alan S. Willsky
Title: Associate Professor of Electrical Engineering

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to the members of my Doctoral Committee for invaluable counsel and unfailing personal support. Professor Adrian Segall stimulated my interest in data communication networks, suggested the topic of this thesis, and generously contributed of his time as colleague and friend during this research. From him I have learned that perserverance is the most important part of research. Professor Wallace E. Vander Velde, chairman of the committee, has served as mentor and teacher throughout my doctoral program. His wise advice, with regard to this thesis and other matters, is greatly appreciated. My association with Professor Alan S. Willsky, both during thesis research and previously, has been most rewarding. His intellectual and personal energy, combined with a fine sense of humor, provide an atmosphere which I find both stimulating and enjoyable.

My research experience at the Electronic Systems Laboratory has been a very pleasant and productive one. I offer my heartfelt thanks to those faculty members, staff members and fellow students who have helped make it so. In particular: to Professor John M. Wozencraft, for graciously providing the administrative support which made this research possible; to Professor Sanjoy K. Mitter, for serving as a thesis reader and sharing of his expertise in optimal control theory; to Professor

Robert G. Gallager, for serving as a thesis reader and providing valuable comments on the message routing problem; to Dr. Stanley B. Gershwin, for many interesting discussions and suggestions; to Ms. Susanna M. Natti, for unselfishly forsaking her art career to apply her creative talents to the expert preparation of the manuscript; to Ms. Myra Sbarounis for efficiently and cheerfully administering to everyday affairs; and finally, to my officemates Mr. Martin Bello and Mr. James R. Yee (graduate students always come last) for maintaining a suitably chaotic atmosphere.

The support of my entire family during my graduate career, and during my education as a whole, has been my greatest incentive. I cannot adequately thank here my parents, Samuel and Rose, my brother Billy and sister Ivy for a lifetime of love and encouragement. It is certainly hopeless to describe in a few words what has been given me by Janie during four years of marriage - unwavering support, sacrifice, and love to be sure - but more than that, what it takes to make life sweet. I thank her every day. We are both ever thankful for our son Ilan, who always makes us smile. In appreciation, I dedicate this thesis to my family.

TABLE OF CONTENTS

Page

ABSTRACT 2

ACKNOWLEDGEMENTS 3

TABLE OF CONTENTS 5

Chapter 1 INTRODUCTION 9

1.1 Introduction to Data Communication Networks 9

1.2 Discussion of the Message Routing Problem 12

1.3 Thesis Overview 17

1.3.1 Objective of Thesis 17

1.3.2 Approach 18

1.3.3 Preview of Results 20

1.4 Synopsis of Thesis 23

Chapter 2 NEW DATA COMMUNICATION NETWORK MODEL AND OPTIMAL CONTROL PROBLEM FOR DYNAMIC ROUTING 26

2.1 Introduction 26

2.2 Basic Elements of the State Space Model 28

2.2.1 Topological Representation 28

2.2.2 Message Flow and Storage Representation 29

2.3 Dynamic Equations and Constraints 33

2.4 Performance Index and Optimal Control Problem Statement 35

Page

2.5 Discussion of the Optimal Control Problem 39

2.6 Previous Work on Optimal Control Problems with State Variable Inequality Constraints 41

2.7 Open Loop Solutions 45

2.7.1 Linear Programming Through Discretization 45

2.7.2 Penalty Function Method 47

Chapter 3 FEEDBACK SOLUTION TO THE OPTIMAL CONTROL PROBLEM WITH CONSTANT INPUTS 50

3.1 Introduction 50

3.2 Feedback Solution Fundamentals 52

3.2.1 Necessary and Sufficient Conditions 52

3.2.2 Controllability to Zero for Constant Inputs 67

3.2.3 Geometrical Characterization of the Feedback Space for Constant Inputs 72

3.3 Backward Construction of the Feedback Space for Constant Inputs 91

3.3.1 Introductory Examples of Backward Boundary Sequence Technique 91

3.3.2 Technique for the Backward Construction of the Feedback Space 113

3.3.2.1 Preliminaries 113

3.3.2.2 Description of the Algorithm 115

3.3.2.3 Discussion of the Algorithm 135

3.3.3 Basic Procedures of the Algorithm 142

Page

3.3.3.1 Solution of Constrained Optimization Problem 142

3.3.3.2 Determination of Leave-the-Boundary Costates, Subregions and Global Optimality 147

3.3.3.3 Construction of Non-Break Feedback Control Regions 168

3.3.3.4 Construction of Break Feedback Control Regions 175

3.3.4 Discussion of Fundamental Properties of the Algorithm 180

3.3.4.1 Non-Global Optimality of Certain Sequences 181

3.3.4.2 Non-Uniqueness of Leave-the-Boundary Costates 186

3.3.4.3 Subregions 190

3.3.4.4 Return of States to Boundary Backward in Time 196

3.4 Summary 202

Chapter 4 FEEDBACK ROUTING WITH CONSTANT INPUTS AND UNITY WEIGHTINGS 204

4.1 Introduction 204

4.2 Preliminaries 206

4.3 Special Properties of the Algorithm 216

4.4 Computational Algorithm 267

4.5 Extension to General Network Problems 278

Chapter 5 CONCLUSION 284

5.1 Discussion 284

Page

5.2 Summary of Results and Related Observations 288

5.2.1 Results 288

5.2.2 Related Observations 289

5.3 Contributions 292

5.4 Suggestions for Further Work 293

Appendix A FINDING ALL EXTREME POINT SOLUTIONS TO THE CONSTRAINED OPTIMIZATION PROBLEM 296

Appendix B EXAMPLE OF NON-BREAK AND BREAK FEEDBACK CONTROL REGIONS 299

REFERENCES 302

BIOGRAPHY 305

Chapter 1

INTRODUCTION

1.1 Introduction to Data Communication Networks

A data communication network is a facility which interconnects a number of data devices (such as computers and terminals) by communication channels for the purpose of transmission of data between them. Each device can use the network to access some or all of the resources available throughout the network. These resources consist primarily of computational power (CPU time), memory capacity, data bases and specialized hardware and software. With the rapidly expanding role being played by data processing in today's society (from calculating interplanetary trajectories to issuing electric bills) it is clear that the sharing of computer resources is a desirability. In fact, the distinguished futurist Herman Kahn of the Hudson Institute has forecast that the "marriage of the telephone and the computer" will be one of the most socially significant technological achievements of the next two hundred years.

Research in areas related to data communication networks began in the early 1960's and has blossomed into a sizable effort in the 1970's. The ARPANET (Advanced Research Projects Agency NETwork) was implemented in 1970 as an experimental network linking a variety of university, industrial and government research centers in the United States. Currently, the network connects about one hundred computers throughout the continental United States, Hawaii and Europe. The network has enjoyed considerable success and as a result several other major networks are presently being planned.

Following Kleinrock [1976], we now describe the basic components of a data communication network and their functions. Fundamentally, what is known as the communication subnetwork consists of a collection of nodes which exchange data with each other through a set of links. Each node essentially consists of a minicomputer which may have data storage capability and which serves the function of directing data which passes through the node. The links are data transmission channels of a given data rate capacity. The data devices which utilize the communication subnetwork, known as users, insert data into and receive data from the subnetwork through the nodes. See Figure 1.1. A more detailed description of the network in the context of the analysis of this thesis is presented in Chapter 2.

The data travelling along the links of the network is organized into messages, which are groups of bits which convey some information. One categorization of networks differentiates those which have message storage at the nodes from those which do not. Those with storage are known as store-and-forward networks. Another classification is made according to the manner in which the messages are sent through the network. In a circuit-switching network, one or more connected chains

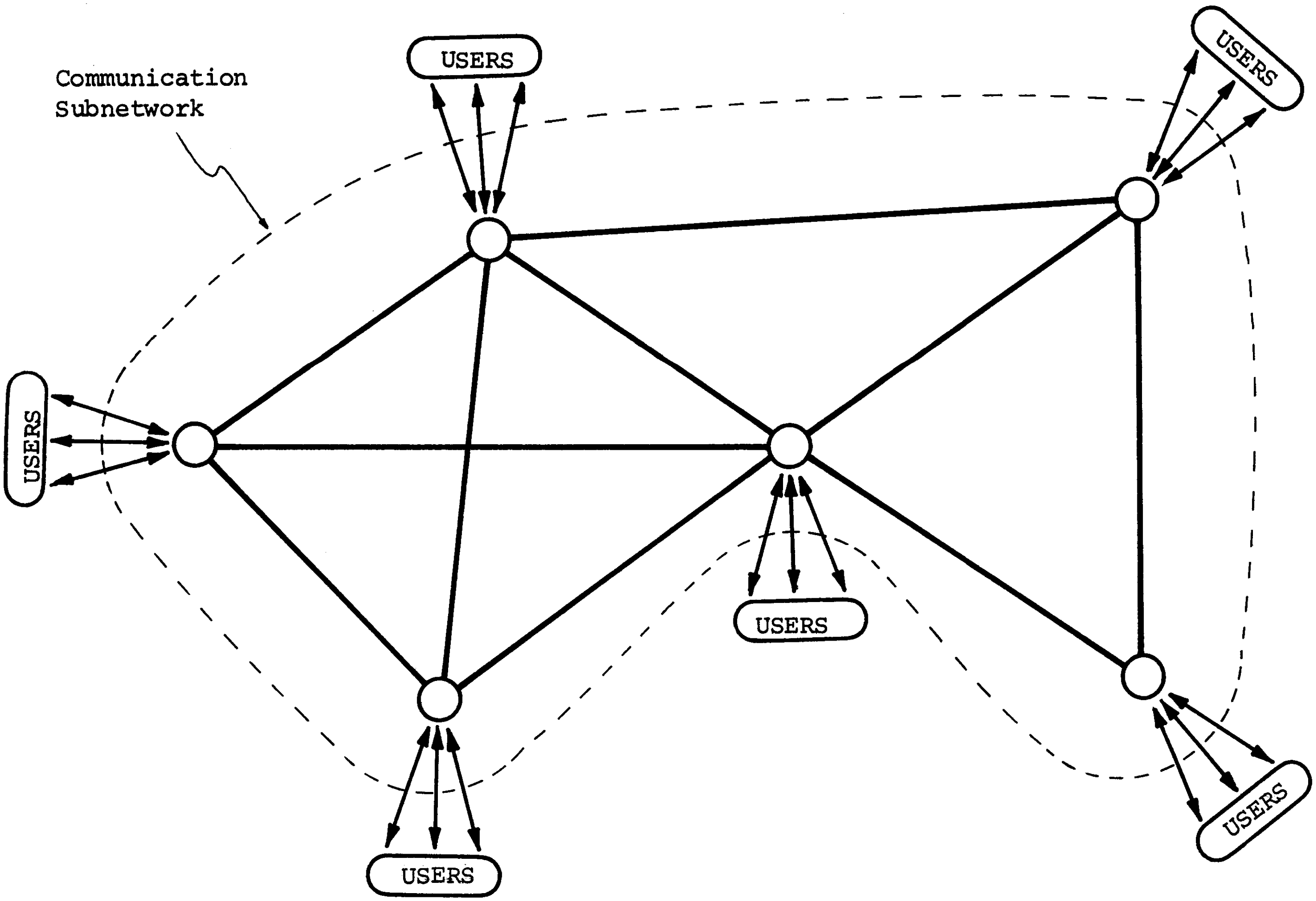


Figure 1.1 Data Communication Network

of links is set up from the node of origin to the destination node of of the message, and certain proportions of data traffic between the origin and destination are then transmitted along these chains. The other category includes both message switching and packet switching networks. In message-switching, only one link at a time is used for the transmission of a given message. Starting at the source node the message is stored in the node until its time comes to be transmitted on an outgoing link to a neighboring node. Having arrived at that node it is once again stored until being transmitted to the next node. The message continues to traverse links and wait at nodes until it finally reaches its destination. Packet-switching is fundamentally the same as message switching, except that a message is decomposed into smaller pieces of maximum length called packets. These packets are properly identified and work their way through the network in the fashion of message switching. Once all packets belonging to a given message arrive at the destination node, the message is reassembled and delivered to the user. In this thesis we shall be concerned with store-and-forward networks which employ message(packet)-switching.

1.2 Discussion of the Message Routing Problem

There is a myriad of challenging problems which must be confronted in the design and operation of data communication networks. Just to name a few, there are the problems of topological design (cost effective allocation of links and their rate capacities), detection of equipment failures and reconfiguration, and the routing of messages through the network from their nodes of origin to their nodes of destination. The latter problem is one of the fundamental issues involved in the operation of networks and as such has received considerable attention in the data-communication network literature. It is clear that the efficiency with which messages are sent to their destinations determines to a great extent the desirability of networking data devices. The subjective term "efficient" may be interpreted mathematically in many ways, depending on the specific goals of the network for which the routing procedure is being designed. For example, one may wish to minimize total message delay, maximize message throughput, cost. etc. In general, this issue is referred to as the <u>routing problem</u>. We shall be concerned with the minimum delay message routing problem in this thesis.

Routing procedures can be classified according to how dynamic they are. At one end of the scale we have purely <u>static</u> strategies in which fractions of the message traffic at a given node with a given destination are directed on each of the outgoing links, where the fractions do not change with time. On the other end of the scale we have <u>completely dynamic</u> strategies, which allow for continual changing of the routes as a function of time, and also as a function of message congestion and traffic requirements in the network. Static procedures are easier to implement than dynamic ones, but lack the important ability to cope with changing congestion and traffic requirements in the network possessed by dynamic strategies.

Although a large variety of routing procedures have been developed and implemented in existing networks (such as ARPANET) the lack of a basic model and theory able to accommodate various important aspects of the routing problem has made it necessary to base the procedures almost solely on intuition and heuristics. Of concern here is the fact that previous techniques have been addressed primarily to static routing strategies, which lack the previously discussed advantages of dynamic strategies. The best known approach to static message routing in store-and-forward data communications networks is due to Kleinrock [1964]. This approach is based upon queueing theory and we describe the principal elements here briefly for comparison with the dynamic method which we shall discuss subsequently:

(a) The messages arrive from users to the network according to independent constant rate Poisson processes and their lengths are assumed to be independent exponentially distributed and independent of their arrival times.

(b) At subsequent nodes along the paths of the messages, the lengths of the messages and their interarrival times become dependent, a fact which makes the analysis extremely difficult. To cope with this problem, the famous independence assumption is introduced, requiring the messages to "lose their identities" at each node and to be assigned new independent lengths.

(c) Once a message arrives at a node, it is first assigned to one of the outgoing links and waits in queue at that link until it is transmitted.

(d) Based on queueing analysis, the average delay in steady state experienced by messages in each link is calculated explicitly in terms of the various arrival rates, average message length and the capacity of the link.

(e) A routing procedure is then found to minimize the average delay over the entire network. See Cantor and Gerla [1974].

The routing procedure thus obtained is static, namely constant in time and a function only of the various average parameters of the system. In the parlance of control theory, which is the language which we shall be using for much of this thesis, such a strategy is referred to as open-loop.

Since Kleinrock's model was proposed in 1962, researchers in the area have repeatedly expressed the desire to find approaches to other aspects of the routing problem. In particular, the goal is to find an approach

(i) in which the independence assumption is not required,

(ii) that will be able to handle transients and dynamical situations and not only steady state, and

(iii) that can lead to optimal closed-loop control strategies, namely strategies that change according to the congestion in the network.

Requirement (i) is desirable since the independence assumption may be quite inappropriate in various situations. Furthermore, it is not easy to assess the validity of this assumption for a given network. The desirability of requirement (ii) has been discussed previously. Finally, perhaps the most important requirement is (iii), for it is a fundamental fact of optimal control theory that closed-loop strategies are much less sensitive than open-loop ones to perturbations in the parameters of the system. Hence, occurrences such as link and node failures or unexpected bursts of user input to the network are accommodated much better by closed-loop strategies.

It is pointed out in Segall [1976] that the traditional queueing theory approach can in principle be adapted to closed loop strategies, but that the number of states required would be immense. This approach looks at each message or packet as an entity, and therefore the state of the network is described by the number and destination of messages (packets) in each of the buffers. Therefore, in a network with N nodes, β outgoing links per node in the average and buffers of maximum capacity of M messages (packets), the number of states is approximately $(\beta NM)^{N-1}$, which is an extremely large number even for the smallest of networks.

Segall [1976] has introduced a model for message routing which is capable of achieving requirements (i)-(iii) above. His approach is to model the flow of messages in a message (packet)-switched store-and-forward data communication network in a <u>continuous</u> dynamical state space setting. The continuous nature of the state is justified by recognizing that any individual message contributes very little to the overall behavior of the network, so that it is not necessary to look individually at each of the messages and their lengths. In this vein, it makes more sense to regard the network in the more macroscopic fashion of Segall's model.

Having established the model, Segall expresses the minimum delay dynamic routing problem as a *linear optimal control problem with linear state and control variable inequality constraints* for which a *feedback* solution is sought. Should such a solution to this problem be obtained, the resulting strategy would be *dynamic* and *closed-loop*. The model and associated optimal control problem are discussed briefly in the next section and are presented in detail in Chapter 2.

1.3 Thesis Overview

1.3.1 Objective of Thesis

The goal of this research is to obtain feedback solutions to the linear optimal control problem with linear state and control variable inequality constraints suggested by Segall for the message routing problem. Undoubtedly, the principal difficulty presented by this problem is the presence of state variable inequality constraints. This contention is supported by the observation that very few feedback solutions have been found for optimal control problems with this type of constraint. We clearly shall have to exploit the special properties of our problem (such as linearity and the structure of the dynamics and constraints) to develop new theory and techniques capable of producing feedback solutions.

1.3.2 Approach

We begin our discussion of approach by briefly describing Segall's state space model. In the model the state variables represent the quantity of messages stored at the nodes, distinguished according to node of current residence and node of ultimate destination. In reality, the measure of message storage at the nodes is a discrete variable (such as number of messages, packets, bits, etc.). However, in the spirit of viewing the network from a macroscopic point of view, we assume that the units of data traffic are such that after appropriate normalization the states may be approximated closely by continuous variables. The control variables represent the flow rate of traffic in the links, where each control represents that portion of a given link's rate capacity devoted to transmitting messages of a given destination. Finally, the inputs are the flow rates of messages entering the network from the users. In this thesis, we consider the inputs to be deterministic functions of time. The dynamical equation which represents the flow of messages in the network is

$$\dot{\underline{x}}(t) = \underline{B}\ \underline{u}(t) + \underline{a}(t) \tag{1.1}$$

where $\underline{x}(t)$, $\underline{u}(t)$ and $\underline{a}(t)$ are the vectors of state variables, control variables and inputs respectively. In the static flow literature the matrix $\underline{B}$ is known as the "incidence matrix", and it is composed solely of +1's, -1's and 0's which capture the node-link configuration of the network. We also have several essential constraints. The state variable inequality constraints are $\underline{x}(t) \geq \underline{0}$ for all t, since the message storage must always be non-negative. The control variable inequality constraints are $\underline{u}(t) \in \mathcal{U}$ for all t, where $\mathcal{U} = \{\underline{u}:\underline{u} \geq \underline{0} \text{ and } \underline{D}\ \underline{u} \leq \underline{C}\}$. Here $\underline{D}$ is a matrix of 0's and +1's and $\underline{C}$ is a vector of link rate capacities. These constraints represent flow non-negativity and link rate capacity constraints respectively.

We now associate with the dynamical state space model the linear cost functional

$$J = \int_{t_0}^{t_f} \underline{\alpha}^T \underline{x}(t)\,dt \tag{1.2}$$

where $\underline{x}(t_0) = \underline{x}_0$ is given, t_f is given implicitly by $\underline{x}(t_f) = \underline{0}$ and $\underline{\alpha}$ is a column vector of constant weighting factors. The implications of the stipulation that $\underline{x}(t_f) = 0$ are discussed in Section 2.4. We note for now that when $\underline{\alpha}$ is all 1's, then J is exactly the total delay experienced by all of the messages traveling through the network on $[t_0, t_f]$. By adjusting the values of the elements of $\underline{\alpha}$, J may be made to represent a desired form of weighted total delay. The optimal control problem which represents the dynamic feedback message routing problem is:

> Find the control $\underline{u}(t)$ as a function of time and state, $\underline{u}(t) \equiv \underline{u}(t, \underline{x})$, that will bring the state from $\underline{x}(t_0) = \underline{x}_0$ (given) to $\underline{x}(t_f) = \underline{0}$ while minimizing J subject to the dynamics and state and control variable inequality constraints.

Our approach to solving the above problem shall now be described briefly. We begin by deriving the necessary conditions of optimality associated with the optimal control problem and prove that they are also sufficient. Realizing that there is extremely little hope of obtaining at this time a feedback solution for the general deterministic input problem, we restrict the inputs to be constant in time. The necessary conditions indicate that with this assumption the optimal feedback control is regionwise constant over the state space. We then develop a procedure which utilizes the necessary conditions to construct all of these regions and identify their associated optimal controls. Although the procedure is not readily implementable on the computer for problems involving general multi-destination networks, we are able to utilize special structural properties of the problem to devise a computer algorithm for problems involving single destination networks with all unity weightings in the cost functional.

1.3.3 Preview of Results

In this section we elaborate on the discussion of approach of the previous section and simultaneously describe the highlights of the results which are obtained.

According to the necessary conditions which are derived for the problem, any optimal control must satisfy

$$\underline{u}^*(t) = \underset{\underline{u}(t)\epsilon\mathcal{U}}{\text{ARG MIN}} \quad (\underline{\lambda}^T(t)\underline{B}\ \underline{u}(t)) \tag{1.3}$$

pointwise in time. Here, $\underline{\lambda}(t)$ is the vector of costate variables corresponding to the state $\underline{x}(t)$ and is given by a linear differential equation with free terminal conditions. Of particular interest is the fact that a costate may exhibit discontinuities at times when its associated state is traveling on a boundary arc $x = 0$. We are also able to prove the significant result that the necessary conditions of optimality are also sufficient.

For given values of the costate vector, the pointwise minimization is a linear program. Owing to fundamental properties of linear programming we are able to deduce the following: (i) the optimal control always lies at boundary points of $\mathcal{U}$, and therefore is of the bang-bang variety; (ii) non-uniqueness of the optimal control is a possibility; (iii) the minimization need not be performed at every time, but may be solved parametrically in time to find the optimal bang-bang controls and switch times.

Note that all of the above results apply for general deterministic inputs. Henceforth we restrict ourselves to the special case in which the inputs are constant in time. In this case, the bang-bang nature of the optimal control implies that the slope of the optimal state trajectory is piecewise constant in time. We are now able to state and prove sufficiency conditions under which any initial state $\underline{x}(t_0) = \underline{x}_0$ is controllable to $\underline{x}(t_f) = \underline{0}$ for a given network with given set of constant inputs.

Using the necessary conditions, we are able to show that the optimal control is regionwise constant and that the regions are convex polyhedral cones in the state space. A procedure is then developed which employs the spirit of dynamic programming to construct these cones and their associated optimal controls by a sequence of backward optimal trajectories propagating from the final time t_f. When a sufficiently large variety of these trajectories is utilized to construct regions, the union of these regions fills up the state space with optimal controls, thus constituting the feedback solution.

In order to solve for the backward optimal trajectory, we must propagate the costates backward in time according to their differential equation and solve the minimization problem (1.3) for all solutions. However, there is a question regarding the appropriate values of the costates which realize a particular backward optimal trajectory through (1.1) and (1.3). The resolution to this problem is geometrical in nature in that it considers the Hamiltonian function associated with the necessary conditions to be a hyperplane which is continuously rotating about the constrained region of message flow while remaining tangent to it. As the costates are the coefficients of the hyperplane, their appropriate values are determined by orienting the hyperplane in a prescribed fashion with respect to the constraint region. This argument freely employs geometrical concepts in linear programming.

We are unable to devise a computer algorithm to implement the above procedure for general multi-destination network problems due to several complicating properties. However, we are able to show that for problems involving single destination networks with all unity weightings in the cost functional these complicating properties do not apply. A computational implementation of the procedure is then readily formulated and a computer example is performed. It turns out that one of the computational tasks of the algorithm is extremely inefficient. This task is involved with the problem of finding all of the non-unique extremal solutions to a linear program. As a consequence, the computational feasibility of the algorithm is contingent upon the development of a more efficient technique for solving this problem.

In evaluating the ultimate desirability of the feedback approach, we must take into account the fact that considerable computer storage may be required to implement the feedback solution on line. This results from the necessity of storing all of the linear inequalities which specify the convex polyhedral cones in the state space, and there may be many such cones. A tradeoff is therefore in order between the storage involved to implement closed-loop solutions and the computation involved in the implementation of open loop solutions.

1.4 Synopsis of Thesis

The purpose of this section is to provide a brief summary of the remaining chapters.

Chapter 2

The state space model and optimal control problem of Segall are described in detail. A brief discussion is devoted to two possible open-loop solutions: The first employs linear programming through discretization in time and the second is an adjoined penalty function technique. Note that although they are open-loop, these techniques are dynamic.

Chapter 3

In this chapter we present the procedure for the synthesis of a feedback solution to the optimal control problem with constant inputs. We begin by deriving the necessary conditions of optimality and prove that they are sufficient. Based upon these conditions we characterize the feedback regions of interest as convex polyhedral cones. After providing a few motivating examples, we present the algorithm for the backward construction of the feedback control. Finally, we isolate and describe in detail those properties of the algorithm which complicate its computational implementation.

Chapter 4

The geometrical linear programming interpretation is utilized to construct proofs that the complicating properties of the algorithm of Chapter 3 do not apply for problems involving single destination networks with all unity weightings in the cost functional. This fortuitous situation enables us to construct an algorithm which is implementable on the computer. An example is run for a five node network. We then present thoughts on how the techniques of this chapter may be applied to general multi-destination network problems with non-unity weightings in the cost functional.

Chapter 5

In this chapter we comment on various aspects of the approach in general and of the feedback control algorithm in particular. Some insight is provided into related topics not specifically discussed in preceding chapters. We then summarize our results, present the specific contributions and make suggestions for further work in this area.

Chapter 2

NEW DATA COMMUNICATION NETWORK MODEL AND OPTIMAL CONTROL PROBLEM FOR DYNAMIC ROUTING

2.1 Introduction

In this chapter, the flow and storage of messages in a message-(packet)-switched store-and-forward data communication network are expressed in a dynamical state space setting. This model for routing was introduced by Segall [1976]. The principal elements of the model - state variables, control variables, and inputs - are defined to represent mathematically the fundamentals of network operation: storage, traffic flow and message input respectively. Emerging from this characterization is an ordinary linear vector differential equation which dynamically describes the storage state of the network at every time. State variable positivity constraints and control variable capacity constraints, both linear, are imposed as an essential part of the model. Presently, we assume that the inputs are deterministic functions of time, representing a scheduled rate of demand.

Arising naturally out of the model as defined is a linear integral cost functional which is equal to the total delay experienced by all of the messages travelling in the network. Owing to the generality of our expression, we are also able to formulate a linear cost functional which corresponds to a measure of weighted message delay in the network. Combining the minimization of the cost functional with the dynamics and associated constraints, we obtain a linear optimal control problem with linear state and control variable inequality constraints for which we seek a feedback solution. We then discuss in some detail how the assumptions inherent in the optimal control problem formulation relate to the real world situation of data communication network message routing operation.

The advantages of this approach were discussed in Section 1.2. The principal disadvantage is associated with the difficulty of solving state variable inequality constrained optimal control problems. This is particularly true when a feedback solution is sought. The problem is placed in perspective by providing a summary of the previous work which has been performed in this area and pointing out the need for the development of additional theory for our particular formulation. This discussion sets the stage for the contributions to this problem area which are achieved in subsequent chapters.

Although the principal goal of the thesis is to obtain closed-loop solutions to the optimal control problem, there are several approaches to obtaining open-loop solutions which are conceptually straightforward and are therefore of some interest. Two such approaches are reported on briefly at the end of this chapter: linear programming through discretization and a penalty function method.

2.2 Basic Elements of the State Space Model

2.2.1 Topological Representation

We visualize data networks graphically to consist of a collection of nodes connected by a set of links between various pairs of the nodes. In Section 1.1 we presented a discussion of the general function of nodes and links in the network. In our model, we shall assume that the links are simplex, that is, carry messages in one direction only from the node at the input end to the node at the output end.

In a network consisting of N nodes we associate with each node an integer in the set $\{1, 2, \ldots, N\}$ and denote this collection of nodes by $\mathcal{N}$. The link connecting node i to node k is denoted by (i,k), and the collection of all links in the network is

Notation 2.1 $L \overset{\Delta}{=} \{(i,k)$, such that $i,k \in \mathcal{N}$ and there is a direct link connecting i to k$\}$.

We now denote

Notation 2.2 $C_{ik} \overset{\Delta}{=}$ capacity of link (i,k) in units of traffic/unit time, $(i,k) \in L$

and for every $i \in \mathcal{N}$ denote

Notation 2.3 $E(i) \overset{\Delta}{=}$ collection of nodes k such that $(i,k) \in L$,

$I(i) \overset{\Delta}{=}$ collection of nodes ℓ such that $(\ell,i) \in L$.

In Figure 2.1 we depict a graphical representation of such a data communication network.

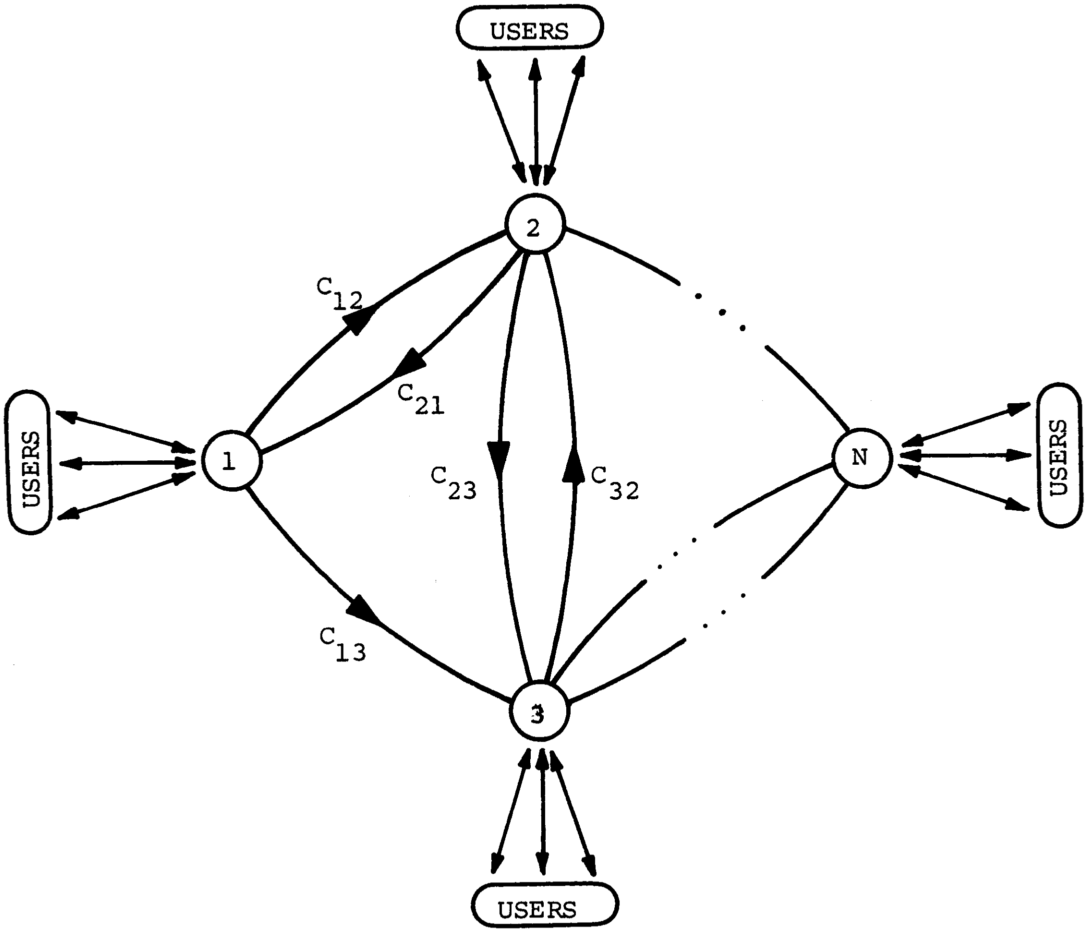


Figure 2.1 Data Communication Network Topology

2.2.2 Message Flow and Storage Representation

In accordance with the operational philosophy of the network, the users at each node in the network may input messages whose ultimate destinations are any of the other nodes in the network. We characterize this message traffic flow input to the network by:

Notation 2.4 $a_i^j(t) \triangleq$ rate of traffic with destination j arriving at node i (from associated users) at time t.

The message traffic from each user is first fed into the associated node and either immediately transmitted on an outgoing link or stored for eventual transmission. Also, each node may serve as an intermediate storage area for messages entering on incoming links enroute to their destinations. Once a message reaches its destination node, it is immediately forwarded to the appropriate user without further storage.

Hence, at each node $i \in \mathcal{N}$ of the network at any point in time we may have messages in residence whose destinations are all nodes other than i. Let us now imagine that at each node $i \in \mathcal{N}$ we have N-1 "boxes"; and that in each of these boxes we place all the traffic (messages, packets, bits, etc.) whose destination is a particular node, regardless of its origin. We do this for all possible destinations 1, 2, ... (i-1), (i+1), ...N. We now define the state variables of our model as

Notation 2.5 $x_i^j(t) \triangleq$ amount of traffic at node i at time t whose final destination is node j, where $i,j \in \mathcal{N}$, $i \neq j$.

The amount of traffic residing in each box at any time t is measured in some arbitrary unit (messages, packets, bits, etc.). Strictly speaking, the states are therefore discrete variables with quantization level determined by the particular unit of traffic selected. However, we shall assume that the units are such that after appropriate normalization the states x_i^j can be approximated by continuous variables. The rationale underlying this approximation is presented in Section 1.2.

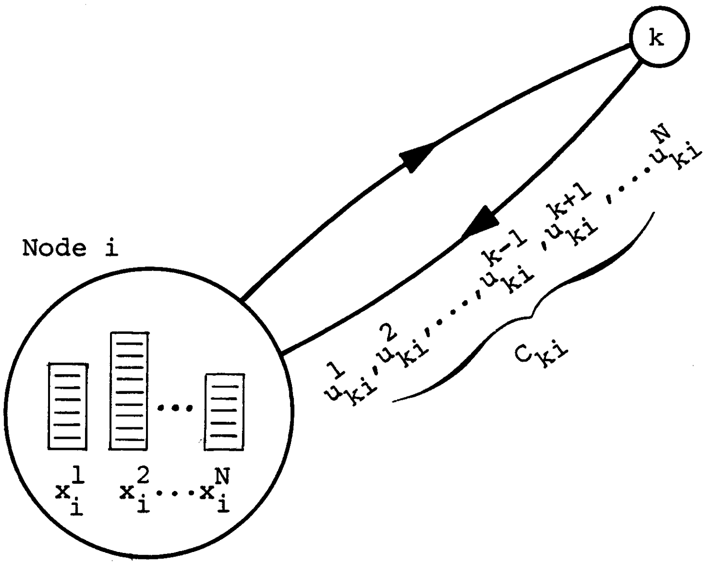
We simply repeat here that this macroscopic point of view regarding messages is justifiable in relation to the overall goals of desirable network operation. Note that in a network consisting of N nodes, the maximum number of state variables as defined above is N(N-1), which most certainly is reasonable when compared to the huge number of states associated with finite-state models (see Section 1.2).

There is a fundamental difference between the states x_i^j described here and the message queues of the traditional approach. In our model, when a message with destination j arrives at node i from either outside the network (i.e., from users) or from some adjacent node, it is classified as belonging in the "box" x_i^j, and as such is associated with the node i. When its time comes to be sent, the routing strategy to be developed in subsequent chapters assigns the message to be sent along some outgoing link $(i,k), k \in E(i)$. In previous models (e.g., Cantor and and Gerla [1974]) messages arriving at node i are immediately assigned to some outgoing link $(i,k), k \in E(i)$, by the routing strategy, there to await transmission. Hence, at least in an intuitive sense, the decision as to what direction to send a message is made in our model at a later time, thus enabling the strategy in force to make a more up-to-date decision. The ultimate performance of our strategy should benefit from this characteristic.

The final element of our message flow representation is the description of the allocation of link flow capacity. Each link emanating from a particular node i is shared by some or all of the up to (N-1) types of messages stored at time t at node i. This now gives rise to the definition of the control variables of our state space model:

Notation 2.6: $u_{ik}^{j}(t) \overset{\Delta}{=}$ portion of the rate capacity of the link (i,k) used at time t for traffic with final destination j.

So defined, the controls are the decision elements of the model available to be adjusted at the command of the routing strategy. The designation of the states corresponding to a particular node and controls corresponding to a particular link are illustrated in Figure 2.2.



k
Node i
x_i^1 $x_i^2 \cdots x_i^N$
$u_{ki}^1, u_{ki}^2, \ldots, u_{ki}^{k-1}, u_{ki}^{k+1}, \ldots u_{ki}^N$
C_{ki}

Figure 2.2 Illustration of State and Control Variables

2.3 Dynamic Equations and Constraints

We are now ready to state the dynamic relationship between the elements of our state space model. Assuming the inputs are deterministic functions of time, the time rate of change of the number of messages contained in each box is given by

$$\dot{x}_i^j(t) = a_i^j(t) - \sum_{k \varepsilon E(i)} u_{ik}^j(t) + \sum_{\substack{\ell \varepsilon I(i) \\ \ell \neq j}} u_{\ell i}^j(t) \qquad (2.1)$$

$$i,j \varepsilon N, \; j \neq i.$$

That is, the box corresponding to the state x_i^j is increased by the rate of messages arriving from users $(a_i^j(t))$ and messages arriving on incoming links $(u_{\ell i}^j(t), \ \ell \varepsilon I(i), \ \ell \neq j)$ and depleted by the rate of messages departing on outgoing links $(u_{ik}^j, \ k \varepsilon E(i))$. A pictorial depiction of the message flow situation corresponding to equation (2.1) is given in Figure 2.3.

The ordinary differential equation formulation of equation (2.1) is valid for deterministic inputs. When the inputs are stochastic, we must express the relationship in the incremental fashion of stochastic differential equations and properly define the nature of the input process. However, the important point here is that the state space description of message flow and storage presented in Section 2.2 is sufficiently general to accommodate a wide variety of input processes.

An essential feature of the model is the set of constraints we must impose on the state and control variables. In order for the

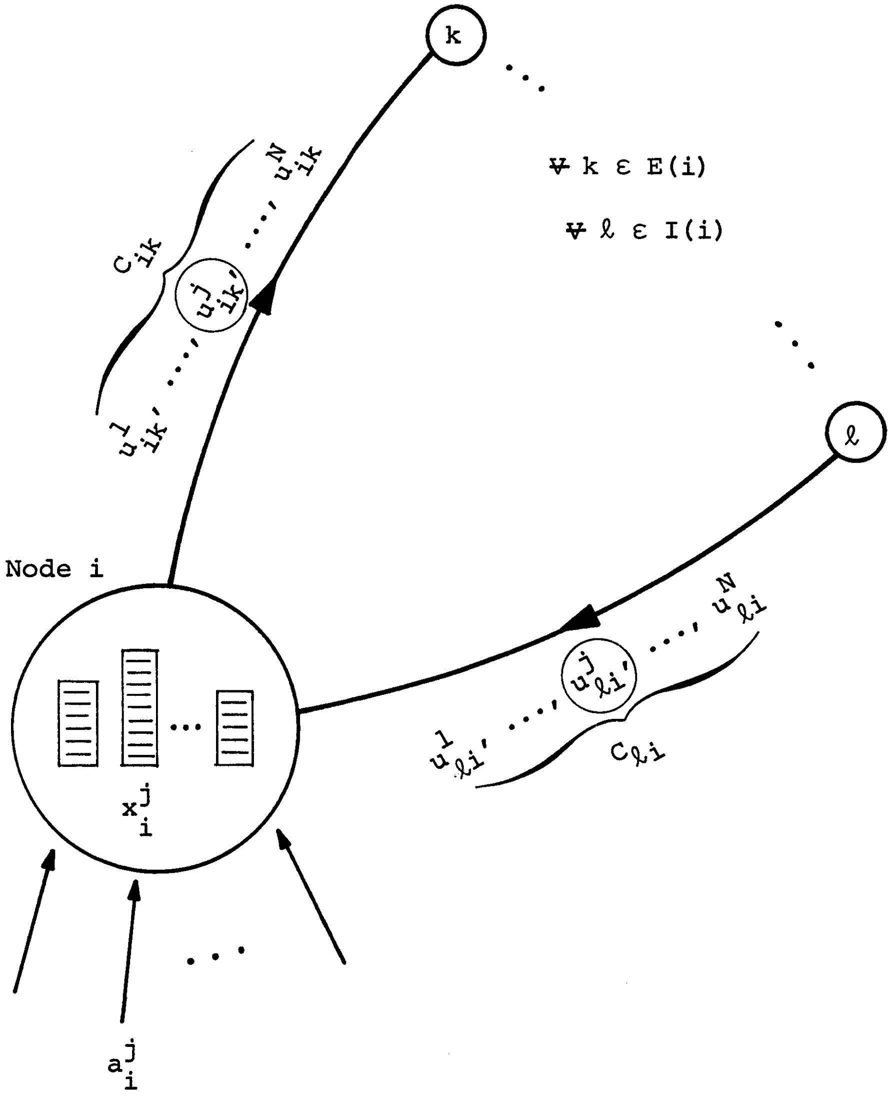

k
$\forall k \in E(i)$
$\forall \ell \in I(i)$
ℓ
C_{ik}
$u_{ik}^1, \ldots, u_{ik}^j, \ldots, u_{ik}^N$
Node i
x_i^j
$u_{\ell i}^1, \ldots, u_{\ell i}^j, \ldots, u_{\ell i}^N$
$C_{\ell i}$
a_i^j

Figure 2.3 Elements of Message Flow Dynamical Equation

mathematical statement of our problem to make physical sense, we must insist on non-negativity of the message storage state variables and of the flow control variables:

$$x_i^j(t) \geq 0 \qquad \forall\, t \tag{2.2}$$

and

$$u_{ik}^j(t) \geq 0 \qquad \forall\, t. \tag{2.3}$$

The rate capacity constraints on each transmission link is expressed by

$$\sum_{j \varepsilon N} u_{ik}^j(t) \leq C_{ik} \qquad \begin{array}{l} \forall\, t \\ \forall (i,k) \;\varepsilon\; L,\; j \neq i\,. \end{array} \tag{2.4}$$

Constraints (2.2)-(2.4) are the only ones which shall be dealt with explicitly in this thesis. The assumption is therefore made that the storage areas containing the messages corresponding to the state variables are infinite in capacity. In practice, of course, these areas will be limited in size, so that we may wish to insist on upper bounding state variable constraints. Possible forms of these constraints are $x_i^j \leq S_i^j$ or $\sum_{j \neq i} x_i^j \leq S_i$ depending on the actual assignment of message storage in a node.

2.4 Performance Index and Optimal Control Problem Statement

As indicated in Section 1.2, one of the major problems associated with routing strategies predicated upon queueing models is the requirement of the independence assumption in order to derive the closed-form expression for the total message delay in the network. This assumption may be at great variance with the physical realities of the situation. On the other hand, optimal control oriented formulations such as ours require only a functional expression for the quantity of interest in terms of the state and control variables of the model. By design, the method in which we have defined the state variables gives rise to possible performance indices which are most appropriate in this application. For example, observe that if $x(t)$ is the amount of message traffic residing in some box at time t, then the quantity

$$\int_{t_0}^{t_f} x(t)\,dt \tag{2.5}$$

gives the total time spent in this box by the traffic passing through it during the time period of interest $[t_0, t_f]$, when t_f is such that $x(t_f) = 0$. Consequently, expression (2.5) is exactly the total delay in the box experienced by the class of messages represented by $x(t)$. Hence, the <u>total delay</u> experienced by all the messages as they travel through the network during $[t_0, t_f]$ is given by

$$D = \int_{t_0}^{t_f} \left[\sum_{i,j} x_i^j(t) \right] dt \tag{2.6}$$

$$i,j \in N,\ j \neq i$$

where t_f is defined as the time at which all the message storage state variables x_i^j go to zero. <u>Priorities</u> can be accommodated in the cost functional (2.6) by associating non-equal weightings α_i^j to the appropriate state variables, so that we have

$$J = \int_{t_0}^{t_f} \left[\sum_{i,j} \alpha_i^j x_i^j(t) \right] dt \tag{2.7}$$

$$i,j \in N,\ j \neq i$$

with t_f defined as above. A logical fashion in which to assign priorities is by destination, in which case $\alpha_k^j = \alpha_\ell^j \quad \forall\, k,\ell \neq j$. Cost functional (2.7) is then a measure of total delay weighted by destination. We now have all the elements needed to state our optimal control problem. In words, the <u>data communication network dynamic message routing problem is</u>:

> At every time t, given the knowledge of the traffic congestion in the network ($x_i^j(t)$, $i,j \in N$, $j \neq i$), dynamically decide what portion of each link capacity to use for each type of traffic (i.e., assign $u_{ik}^j(t)$, $(i,k) \in L$, $j \in N$), so as to minimize the specified cost functional (i.e., total delay if $\alpha_i^j = 1 \quad \forall\, i,j,\ i \neq j$) while bringing the traffic from a specified initial level to zero at the final time.

To facilitate the expression of this problem in compact mathematical form, we define the five column vectors - $\underline{a}$, $\underline{x}$, $\underline{u}$, $\underline{C}$ and $\underline{\alpha}$ - which are respectively concatenations of the inputs, state variables, control variables, link capacities and cost functional weightings. Denote

n = dimension ($\underline{a}$) = dimension ($\underline{x}$) = dimension ($\underline{\alpha}$), m = dimension ($\underline{u}$) and r = dimension ($\underline{C}$) = card (L). For a given network topology we define the $n \times m$ matrix $\underline{B}$ as follows: associated with every state variable x_i^j is a row $\underline{b}^T$ of $\underline{B}$ such that

$$\underline{b}^T\underline{u} = -\sum_{k\varepsilon E(i)} u_{ik}^j + \sum_{\substack{\ell\varepsilon I(i)\\ \ell\neq j}} u_{\ell i}^j \tag{2.8}$$

$$i,j \ \varepsilon\ N,\ j \neq i.$$

The matrix $\underline{B}$ is analogous to the incidence matrix which describes flow in a static network. However, a fundamental distinction from the static flow situation is that we do not require conservation of flow at nodes as we have the capability of message storage in nodes. Note that $\underline{B}$ is composed entirely of +1's, -1's and 0's and that every column of $\underline{B}$ has at most two non-zero elements. If a particular column has exactly one non-zero entry then it is -1.

Similarly, we define the $r \times m$ matrix $\underline{D}$: associated with every link (i,k) is a row $\underline{d}^T$ of $\underline{D}$ such that

$$\underline{d}^T\underline{u} \leq C_{ik} \tag{2.9}$$

represents the constraint (2.4). The elements of $\underline{D}$ are 0's and +1's only, and each column has precisely one +1.

We may now compactly express the <u>linear optimal control problem with linear state and control variable inequality constraints</u> which represents the data communication network dynamic message routing problem previously stated:

Find the set of controls $\underline{u}$ as a function of time and state

$$\underline{u}(t) \equiv \underline{u}(t, \underline{x}) \qquad t \in [t_0, t_f] \tag{2.10}$$

that will bring the state from a given initial condition $\underline{x}(t_0) = \underline{x}_0$ to $\underline{x}(t_f) = \underline{0}$ and minimizes the cost functional

$$J = \int_{t_0}^{t_f} [\underline{\alpha}^T \underline{x}(t)]dt \tag{2.11}$$

subject to the state dynamics

$$\dot{\underline{x}}(t) = \underline{B}\ \underline{u}(t) + \underline{a}(t) \tag{2.12}$$

and constraints on the state and control variables

$$\underline{x}(t) \geq 0 \qquad \forall\ t \in [t_0, t_f] \tag{2.13}$$

$$U \begin{cases} \underline{D}\ \underline{u} \leq \underline{C} \\ \underline{u} \geq \underline{0}. \end{cases} \tag{2.14}$$

Note that $\underline{a}(t)$ must be such that the state is controllable to zero with the available controls. Conditions under which this is true are given for a special case in Section 3.2.2.

2.5 Discussion of the Optimal Control Problem

Before engaging in the details of the solution to (2.10)-(2.14), some discussion regarding the validity of the problem statement is appropriate. To begin, we have stipulated inputs which are known explicitly as a function of time, whereas computers certainly operate in a stochastic user demand environment. Secondly, the requirement that all the storage states go to zero at some final time is not consistent with a network continually receiving input and storing messages in a steady fashion. Also, as pointed out in Section 2.3, we have ignored upper bounds on message storage capacity. Finally, by specifying a feedback function of the type (2.10) we are assuming that total information regarding storage is available throughout the entire network for controller decisions. In practice, one may wish to consider schemes which allow for the control decision to be made on the basis of local information only, so called distributed control schemes.

With the above drawbacks in mind, we now provide justification for our approach. We begin by pointing out that none of the assumptions made thus far are inherent in a basic state space model. These have simply been invoked to provide a problem formulation for which there exists some hope of obtaining a reasonable solution at this early stage of experience with the model. Also, they may not be as limiting as they first appear. For instance, one possible approximation for the situation with stochastic inputs is to take into account only the ensemble average rates of the inputs. We then design the routing strategy by solving (2.10)-(2.14) with these averages serving as the deterministic inputs $\underline{a}(t)$, and employ the controls thus obtained in the operation of the network. Such a strategy may prove to be reasonably successful if the variances and higher moments of the distributions of the inputs are small compared to the means.

Next, the requirement that all states go to zero at the final time may correspond to a situation in which one wishes to dispose of message backlogs at the nodes for the purpose of temporarily relieving congestion in the network locally in time. This procedure may represent that portion of an overall scheme during which inputs are appropriately regulated or no longer forthcoming. In the latter case we may refer to the resulting operation as a "minimum delay close down procedure".

Elimination of the state variable upper bounds is not always limiting, as we shall discover for a class of single destination network problems studied in Chapter 4. In this case, the optimal routing strategy never requires any state to exceed its initially specified value, which certainly must be within the available storage capacity.

Finally, the assumption of a centralized controller may well be valid in the case of a small network. An example of this is the IBM/440 network. See Rustin [1972]. At any rate, obtaining the routing strategy under this assumption could prove extremely useful in the determination of the suboptimality of certain decentralized schemes.

2.6 Previous Work on Optimal Control Problems with State Variable Inequality Constraints

Pioneering research into the problem of the optimal control of dynamical systems involving inequality constraints on the control variables was performed by Valentine [1937] and McShane (1939]. Problems also involving inequality constraints on functions of the state variables alone were not treated until the early 1960's, probably motivated by the emerging aerospace problems of that time. Pontryagin [1959] provided the setting for the new approaches with the introduction of his famous maximum principle, a landmark work in the field of optimal control. Gamkrelidze [1960] studied the situation in which the first time derivative of the state constraint function is an explicit function of the control variable, or so called "first order state variable inequality constraints." Note from equations (2.12) and (2.13) that this is our situation. This work was devoted to finding necessary conditions in the form of multiplier rules which must be satisfied by extremal trajectories. Berkovitz [1962] and Dreyfus [1962] derive similar results from the points of view of the calculus of variations and dynamic programming respectively.

Subsequent works involved with necessary conditions have been devoted primarily to unravelling the technical difficulties which arise when a time derivative higher than the first is required to involve the control variable explicitly (e.g. Bryson, Denham and Dreyfus [1963], Speyer [1968] and Jacobson, Lele and Speyer [1971]). The literature is quite often at variance with regard to the necessary conditions associated with this problem, although the later work cited appears to present a satisfactory resolution. At any rate, we shall not be concerned with these differences as our problem involves only first order state constraints.

Computational aspects of the problem are dealt with in Ho and Bretani [1963] and Denham and Bryson [1964]. Both works present iterative numerical algorithms for the solution of the open loop problem, the procedure discussed in the latter being essentially an extension of the steepest-ascent method commonly used in control problems unconstrained in the state. As such, it appears doubtful that this algorithm would exhibit acceptable convergence properties for our linear cost problem, particularly in the vicinity of the optimum. Ho and Bretani [1963] report that this is also true of their algorithm.

Little theoretical and computationally oriented attention has been paid to the class of control problems with state variable inequality constraints and control appearing linearly in the dynamics and performance index. In this case, the control is of the bang-bang type and the costates may be characterized by a high degree of nonuniqueness. Maurer [1975] examines the necessary conditions associated with this problem when the control and state constraint are both scalars, and presents an interesting analogy between the junction conditions associated with state boundary arcs and singular control arcs. However, no computational algorithm is reported.

Perhaps the most interesting computational approach presented for the all linear problem is the mathematical programming oriented cutting plane algorithm of Kapur and Van Slyke [1970]. The basic algorithm consists of solving a sequence of succeedingly higher dimensional optimal control problems without state space constraints. Under certain hypotheses they are able to prove strong convergence of the control to the optimum. The drawbacks to this approach are that the state of the augmented problem may grow unreasonably large, and that even unconstrained state linear optimal control problems may be difficult to solve efficiently. In the same paper Kapur and Van Slyke [1970] suggest formulating the problem as a large linear program through discretization of all constraints, a more or less brute force approach which is discussed briefly in Section 2.7.1.

Common to all of the approaches described above is that none broach the difficult problem of obtaining feedback solutions to the state constrained optimal control problem. In fact, the application of necessary conditions to arrive at feedback solutions is not a common occurrence even for unconstrained state problems. Notable exceptions are the linear time-optimal and linear quadratic problem, for which the feedback solutions are well known.

In light of these facts, in order to solve problem (2.10)-(2.14), we must first develop and understand the necessary conditions associated with optimal solutions and creatively apply them to obtain a feedback solution. Certainly, we must fully exploit the total linearity of our problem, which has not been done heretofore. As is usually the case, we shall be forced into making even further assumptions about the problem in order to achieve our goal. This overall effort constitutes the primary mathematical contribution of this work, and is the subject of Chapters 3-5.

2.7 Open Loop Solutions

Although our foremost goal is the development of feedback solutions to the optimal control problem of Section 2.4., several open loop solutions have received consideration. The principal advantage of these approaches is that they apply to inputs represented by deterministic functions of time of arbitrary form. Also, at least in principle, open-loop solutions may be implemented as feedback schemes by continually recalculating them in time with the current state taken as the initial condition for each problem. As such, open-loop solutions are worthy of brief mention at this time, but we shall not pursue these particular approaches further in this thesis.

2.7.1 Linear Programming Through Discretization

This technique is a rather standard approach for linear optimal control problems. See, for example, Kapur and Van Slyke [1970]. We first begin with the assumption that the inputs are such that the state can be driven to zero with the available controls. Next, we select a time T which is sufficiently large to insure that $T \geq t_f$, where t_f is such that $\underline{x}(t_f) = \underline{0}$. To discretize, we divide the time interval [0, T] into P parts, each of length $\Delta t = t_p - t_{p-1}$, $p \in [1, 2, \ldots, P]$. We then make the Cauchy-Euler approximation to the dynamics

$$\dot{\underline{x}}(t_p) = [\underline{x}(t_{p+1}) - \underline{x}(t_p)]/\Delta t. \tag{2.15}$$

With this approximation, equation (2.12) may be written

$$\underline{x}(t_{p+1}) = \underline{x}(t_p) + \Delta t[\underline{a}(t_p) + \underline{B}\,\underline{u}(t_p)], \tag{2.16}$$

$$\forall\, p \in [0, 1, \ldots, P-1]$$

and the performance index may be approximated to first order by the discrete expression

$$J_d = \sum_{p=0}^{P} \underline{\alpha}^T \underline{x}(t_p). \tag{2.17}$$

In this format we have the following constraints:

$$\underline{D}\,\underline{u}(t_p) \leq \underline{C} \tag{2.18a}$$

$$\underline{u}(t_p) \geq \underline{0} \tag{2.18b}$$

$$\underline{x}(t_p) \geq \underline{0} \tag{2.18c}$$

$$\forall\, p \in [0, 1, \ldots, P].$$

We now consider the $\underline{x}(t_p)$ and $\underline{u}(t_p)$ $\forall\, p \in [0, 1, \ldots, P]$ to be the decision variables of a linear programming problem. The total number of such variables is (n+m)P. The non-negativity constraints (2.18b) and (2.18c) are consistent with standard linear programming format.

Since (2.17) is a linear function of the decision variables, we have a linear programming problem with nP equality constraints (2.16) and rP inequality constraints (2.18a). The general format of the program is known as the "staircase structure", a form which has received considerable attention in the programming literature. Such works are devoted to exploiting the special structure of the problem in order to reduce computation through application of decomposition techniques. See, for example, Manne and Ho [1974]. Cullum [1969] proves that as the number of discretization points P goes to infinity, the solution to the discrete problem approaches the optimal solution of the original problem. However, for a reasonable size network and for P sufficiently large to insure good quality of the approximation, the size of the linear program (both in terms of the number of variables and the number of constraints) becomes prohibitively large for practical application. For this reason, this technique has been applied only for the purpose of obtaining sample solutions to provide insight into the properties of optimal solutions.

2.7.2 Penalty Function Method

Optimal control problems with inequality constraints on the state (and/or) control have frequently been solved in an open-loop fashion by converting them to a sequence of problems without inequality constraints by means of penalty functions. One such technique is presented by Lasdon, Warren and Rice [1967]. The penalty function detailed in that paper works from inside the constraint, the penalty increasing as the boundary is approached. Applying this technique to our situation, the state variable inequality constrained problem (2.10)-(2.14) is converted to a problem without state constraints by augmenting the performance index with a penalty function as follows:

$$J_a = \int_0^T \sum_{i,j} (x_i^j(t) + \varepsilon_i^j/x_i^j(t))\, dt \qquad (2.19)$$

$$\varepsilon_i^j > 0,$$

$$i,j \in N,\ i \neq j.$$

The modified problem is then to minimize (2.19) subject to the dynamics (2.12) and the control constraints (2.14).

Since the penalty function term

$$\int_0^T \sum_{i,j} \left[(\varepsilon_i^j/x_i^j(t))\,dt\right] \quad i,j \in N,\ i \neq j \qquad (2.20)$$

approaches infinity as any x_i^j approaches its boundary $x_i^j = 0$, we would speculate that the minimizing solution remains within the constrained region $\underline{x} \geq \underline{0}$. In Lasdon, Warren and Rice [1967] it is shown that this conjecture is true, and further that the minimizing control as a function of time and the minimizing cost approach those for the constrained problem as $\max_{i,j} (\varepsilon_i^j) \to 0 \quad i,j \in N \quad i \neq j$. Also, since we are approaching the constraint boundary from the interior, any solution for the unconstrained problem is also feasible for the constrained problem.

In order to implement this technique, we need to solve the unconstrained problem by any appropriate numerical technique for successively decreasing values of ε_i^j. Gershwin [1976] has created a program for the solution of the penalty function approach to this problem in which he utilizes a modified form of differential dynamic programming for each unconstrained minimization. The computational efficiency of the algorithm is greatly enhanced by the exploitation of parametric linear programming techniques. Whether or not this scheme encounters numerical difficulties as ε_i^j grows very small remains to be determined.

Chapter 3

FEEDBACK SOLUTION TO THE OPTIMAL CONTROL PROBLEM WITH CONSTANT INPUTS

3.1 Introduction

We have formulated the data communication network dynamic message routing problem as the linear optimal control problem of Section 2.4. In this chapter we develop the fundamental theory underlying a novel approach to the synthesis of a feedback solution to that problem. The technique is predicated upon the assumption that all inputs to the network are constant in time. In this section we present a brief review of the development.

We begin by presenting the necessary conditions for the general deterministic problem (inputs not constrained to be constant) and discuss these conditions in some detail. We then show that these conditions are also sufficient, a most fortuitous situation since it guarantees the optimality of trajectories which we eventually shall construct using these conditions.

The subsequent discussion is restricted to the constant inputs case. Based exclusively on the necessary and sufficiency conditions, a geometrical characterization of the feedback space for constant inputs is presented: the regions of $\underline{x}$-space over which the same controls and sequence of states on and off boundary arcs are optimal are convex polyhedral cones in the state space. It is this special property (a result of the total linearity of the problem) which embodies the adaptability of the constant input formulation to the synthesis of a feedback solution. What one needs is an algorithm to specify these regions and the appropriate controls.

Next, we present several simple examples to illustrate how these regions may be constructed. This is accomplished by producing a certain set of optimal trajectories backward in time from the final state $\underline{x} = \underline{0}$. Each portion of these trajectories generates one of these convex polyhedral conical regions, thereby providing the optimal control in that region. If a sufficiently complete set of trajectories is calculated, we manage to find enough regions to fill out the entire space.

The remainder of the chapter is devoted to the extension of this concept to the general network optimal control problem with constant inputs. Taking the lead from the examples performed, an algorithm is presented for the construction of the conical regions from a comprehensive set of backward optimal trajectories. This algorithm is in general form, and several specific issues are raised regarding its execution. An issue of central importance is the determination of the appropriate set of costates required by the algorithm. The resolution of this question is essentially geometrical in nature, relying upon the interpretation of the Hamiltonian as a continuously rotating hyperplane. In fact, many of the arguments are geometrical, so that we frequently use two and three dimensional examples for illustration of the ideas.

Unfortunately, there are several characteristics associated with the backward construction method which complicate its formulation as a compact computational algorithm for the general network model. These properties are introduced by way of examples in the Section 3.3.4. Looking ahead, however, we are able to show in Chapter 4 that these bothersome properties are non-existent in the case of problems involving single destination networks with all unity weightings in the cost functional.

3.2 Feedback Solution Fundamentals

3.2.1 Necessary and Sufficiency Conditions

The necessary and sufficiency conditions to be presented are valid for arbitrary deterministic inputs. For ease of reference, we restate our problem here.

Minimization

$$\min_{\underline{u}(t)\varepsilon\mathcal{U}} J(\underline{x}(t)) \tag{3.1}$$

Performance Index

$$J(\underline{x}(t)) = \int_{t_0}^{t_f} [\underline{\alpha}^T\underline{x}(t)]dt \tag{3.2}$$

Dynamics

$$\dot{\underline{x}}(t) = \underline{B}\ \underline{u}(t) + \underline{a}(t) \tag{3.3}$$

Boundary Conditions

$$\underline{x}(t_0) = \underline{x}_0 \qquad \underline{x}(t_f) = \underline{0} \tag{3.4}$$

State Constraints

$$\underline{x}(t) \geq 0 \quad \forall\ t\ \varepsilon\ [t_0, t_f] \tag{3.5}$$

Control Constraints $\quad \mathcal{U} \begin{cases} \underline{D}\,\underline{u}(t) \le \underline{C} \\ \underline{u}(t) \ge \underline{0} \end{cases} \quad \forall\, t \in [t_0, t_f] \qquad (3.6)$

Theorem 3.1 (Necessary Conditions)

Let the scalar functional h be defined as follows

$$h(\underline{u}(t), \underline{\lambda}(t)) \overset{\Delta}{=} \underline{\lambda}^T(t)\dot{\underline{x}}(t) = \underline{\lambda}^T(t)[\underline{B}\,\underline{u}(t) + \underline{a}(t)]. \tag{3.7}$$

A necessary condition for the control law $\underline{u}^*(\cdot) \in \mathcal{U}$ to be optimal for problem (3.1)-(3.6) is that it minimize h pointwise in time, namely

$$\underline{\lambda}^T(t)\underline{B}\,\underline{u}^*(t) \le \underline{\lambda}^T(t)\underline{B}\,\underline{u}(t) \tag{3.8}$$

$$\forall\, \underline{u}(t) \in \mathcal{U} \qquad \forall\, t \in [t_0, t_f].$$

The costate $\underline{\lambda}(t)$ is possibly a discontinuous function which satisfies the following differential equation

$$-d\underline{\lambda}(t) = \underline{\alpha}dt + d\underline{\eta}(t), \quad t \in [t_0, t_f] \tag{3.9}$$

where componentwise $d\underline{\eta}(\tau)$ satisfies the following complementary slackness condition

$$\left.\begin{aligned} x_i^j(t)d\eta_i^j(t) &= 0 \quad (3.10) \\ d\eta_i^j(t) &\le 0 \quad (3.11) \end{aligned}\right\} \begin{aligned} &\forall\, t \in [t_0, t_f] \\ &i,j \in \mathcal{N},\ j \ne i. \end{aligned}$$

The terminal boundary condition for the costate differential equation is

$$\underline{\lambda}(t_f) = \underline{\nu} \quad \text{free} \tag{3.12}$$

and the transversality condition is

$$\underline{\lambda}^T(t_f)\underline{\dot{x}}(t_f) = \underline{0}. \tag{3.13}$$

Finally, the function h is everywhere continuous, i.e.

$$h(\underline{u}(t^-), \underline{\lambda}(t^-)) = h(\underline{u}(t^+), \underline{\lambda}(t^+)) \tag{3.14}$$

$$\forall\, t \,\varepsilon\, [t_0, t_f].$$

<u>Proof:</u>

Jacobson, Lele and Speyer [1971] present a generalized Kuhn-Tucker theorem in a Banach space for the minimization of a differentiable function subject to inequality constraints. For our problem, it calls for the formation of the Lagrangian

$$\bar{J} = \int_{t_0}^{t_f} \underline{\alpha}^T\underline{x}(\tau)d\tau + \int_{t_0}^{t_f} \underline{\lambda}^T(\tau)[\underline{B}\ \underline{u}(\tau) + \underline{a}(\tau) - \underline{\dot{x}}(\tau)]d\tau$$

$$+ \int_{t_0}^{t_f} d\underline{\eta}(\tau)^T\underline{x}(\tau) + \underline{\nu}^T\underline{x}(t_f) \tag{3.15}$$

where $\underline{\eta}$ is an n × 1 vector adjoining the state constraints which satisfies the complementary slackness condition

$$\int_{t_0}^{t_f} d\underline{\eta}^T(\tau)\underline{x}(\tau) = 0 \tag{3.16}$$

$$d\underline{\eta}(\tau) \leq 0 \tag{3.17}$$

$$\forall\, \tau \,\varepsilon\, [t_0, t_f].$$

The vector $\underline{\nu}$ which adjoins the final condition is an n × 1 vector of arbitrary constants.

For $\underline{u}^*(\cdot)$ to be optimal, $\bar{J}$ must be minimized at $\underline{u}^*(\cdot)$ where $\underline{x}(\cdot)$, $\underline{x}(t_f)$ and t_f are unconstrained and $\underline{u}$ is constrained by $\underline{u} \in \mathcal{U}$. Taking the differential of $\bar{J}$ with respect to arbitrary variations of $\underline{x}(\cdot)$, $\underline{x}(t_f)$ and t_f we obtain

$$\begin{aligned} d\bar{J} = & \int_{t_0}^{t_f} \underline{\alpha}^T \delta\underline{x}(\tau)d\tau + \underline{\alpha}^T \underline{x}(t_f)dt_f \\ & - \int_{t_0}^{t_f} \underline{\lambda}^T(\tau)\delta\dot{\underline{x}}(\tau)d\tau + \int_{t_0}^{t_f} d\underline{\eta}^T(\tau)\delta\underline{x}(\tau) \\ & + \underline{\nu}^T d\underline{x}(t_f) \end{aligned} \tag{3.18}$$

where $\delta\underline{x}$ is the variation in $\underline{x}$ for time held fixed and

$$d\underline{x}(t_f) = \delta\underline{x}(t_f) + \dot{\underline{x}}(t_f)dt_f \tag{3.19}$$

is the total differential of $x(t_f)$. We next integrate the third term of (3.18) by parts and rearrange to obtain

$$\begin{aligned} d\bar{J} = & \int_{t_0}^{t_f} \delta\underline{x}^T(\tau)\,[\underline{\alpha}d\tau + d\underline{\eta}(\tau) + d\underline{\lambda}(\tau)] \\ & - \underline{\lambda}^T(t_f)\delta\underline{x}(t_f) + \underline{\lambda}^T(t_0)\underline{\delta x}(t_0) \\ & + \underline{\alpha}^T\underline{x}(t_f)dt_f + \underline{\nu}^T d\underline{x}(t_f)\,. \end{aligned} \tag{3.20}$$

Since the initial conditions are fixed we have $\delta\underline{x}(t_0) = \underline{0}.$ Substituting for $d\underline{x}(t_f)$ from (3.19):

$$d\bar{J} = \int_{t_0}^{t_f} \delta\underline{x}^T(\tau)\,[\underline{\alpha}d\tau + d\underline{\eta}(\tau) + d\underline{\lambda}(\tau)] \; +$$

$$+ [\underline{\nu}^T - \underline{\lambda}^T(t_f)]\delta\underline{x}(t_f)$$

$$+ [\underline{\alpha}^T\underline{x}(t_f) + \nu^T\dot{\underline{x}}(t_f)]dt_f. \qquad (3.21)$$

Now, in order for $\bar{J}$ to be stationary with respect to the free variations $\delta\underline{x}(\tau)$, $\delta\underline{x}(t_f)$ and dt_f we must have

$$\underline{\alpha}d\tau + d\underline{\eta}(\tau) + d\underline{\lambda}(\tau) = 0 \qquad (3.22)$$

$$\underline{\lambda}(t_f) = \underline{\nu} \qquad \underline{\nu} \text{ free} \qquad (3.23)$$

$$\underline{\nu}^T\dot{\underline{x}}(t_f) = \underline{\lambda}^T(t_f)\dot{\underline{x}}(t_f) = -\underline{\alpha}^T\underline{x}(t_f). \qquad (3.24)$$

Equations (3.16) and (3.17) together with the constraint $\underline{x} > 0$ imply

$$d\eta_i^j(t)x_i^j(t) = 0 \qquad \forall\, \tau \in [t_0, t_f] \qquad (3.25)$$
$$i,j \in N,\ j \neq i.$$

If we integrate the term $\int_{t_0}^{t_f} \underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau)d\tau$ by parts in equation (3.15) and substitute equations (3.16) and (3.22)-(3.24) into (3.15) we obtain

$$\bar{J} = \int_{t_0}^{t_f} \underline{\lambda}^T(\tau)[\underline{B}\ \underline{u}(\tau) + \underline{a}(\tau)]d\tau. \qquad (3.26)$$

In order for $\bar{J}$ to be minimized with respect to $\underline{u}(\cdot)\varepsilon\,\mathcal{U}$, the term $\underline{\lambda}^T(\tau)\ \underline{B}\ \underline{u}(\tau)$ must clearly be minimized pointwise in time, that is

$$\underline{\lambda}^T(\tau)\underline{B}\ \underline{u}^*(\tau) \leq \underline{\lambda}^T(\tau)\underline{B}\ \underline{u}(\tau) \qquad (3.27)$$

$$\forall\, \underline{u}(\tau) \ \varepsilon\ \mathcal{U}\ ,\ t\ \varepsilon\ [t_0, t_f].$$

Thus, we have accounted for equation (3.8), leaving only (3.14) to be proven.

To this end, let us assume that we have an optimal state trajectory $\underline{x}^*(t)$ and associated costate trajectory $\underline{\lambda}(t)$, $t \in [t_0, t_f]$. Then by the principle of optimality, for any fixed $\tau \leq t_f$, the functions $\underline{x}^*(t)$ and $\underline{\lambda}(t)$, $t \in [t_0, \tau]$, are optimal state and costate trajectories which carry the state from $\underline{x}_0$ to $\underline{x}(t_f) = \underline{x}(\tau)$. Hence, all of our previous conditions apply on $[t_0, \tau]$ with $\underline{x}(t_f) = x(\tau)$. Applying the transversality condition (3.24) at $t_f = \tau$, we obtain

$$\underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau) = -\underline{\alpha}^T\underline{x}(\tau). \tag{3.28}$$

Since equation (3.28) holds for all $\tau \in [t_0, t_f]$ and $x(\tau)$ is everywhere continuous, then $\underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau)$ must be everywhere continuous. This proves equation (3.14). ■ Theorem 3.1

All of the necessary conditions which refer to the functional h defined in (3.7) also apply to the more general (and familiar) functional called the <u>Hamiltonian</u> defined as

$$H(\underline{x}(t), \underline{u}(t), \underline{\lambda}(t), \underline{\mu}(t)) = \underline{\alpha}^T\underline{x}(t) + \underline{\lambda}^T(t)\dot{\underline{x}}(t) + \underline{\mu}^T(t)\underline{x}(t). \tag{3.29}$$

The vector of multiplier functions $\underline{\mu}(t)$ is defined componentwise as

$$\mu_i^j(t) = \frac{d\eta_i^j(t)}{dt} \tag{3.30}$$

and therefore μ_i^j is defined only at points for which η_i^j is absolutely continuous. We need not be concerned with this technicality when we use h, which still embodies the important properties of H for our problem.

For emphasis, we shall discuss next the nature of the costates as functions of the corresponding states as determined by equations (3.9)-(3.10). At points of absolute continuity of η_i^j, the associated costate differential equation may be expressed in the familiar form

$$-\dot{\lambda}_i^j(\tau) = \frac{\partial H}{\partial x_i^j} = \alpha_i^j + \mu_i^j(\tau) \ . \tag{3.31}$$

When the state $x_i^j > 0$ (x_i^j is said to be on an interior arc) then equation (3.10) implies $d\eta_i^j = 0$, and equation (3.30) gives $\mu_i^j = 0$. Therefore, equation (3.31) reduces to

$$-\dot{\lambda}_i^j(\tau) = \alpha_i^j \quad \text{when } x_i^j(\tau) > 0. \tag{3.32}$$

When the state $x_i^j = 0$ (x_i^j is said to be on a boundary arc), its costate is possibly discontinuous, depending on the nature of η_i^j. At points for which η_i^j is absolutely continuous, we have from equations (3.10) and (3.11) that $d\eta_i^j \leq 0$ (since $x_i^j = 0$), and therefore equation (3.30) implies $\mu_i^j(t) \leq 0$. Hence,

$$\left.\begin{aligned} -\dot{\lambda}_i^j(\tau) &= \alpha_i^j + \mu_i^j(\tau) && (3.33) \\ \mu_i^j(\tau) &\leq 0 && (3.34) \end{aligned}\right\} \text{ when } x_i^j(\tau) = 0.$$

On the other hand at times τ_p when η_i^j experiences jumps of magnitude $\Delta\eta_i^j(\tau_p)$, equation (3.11) indicates

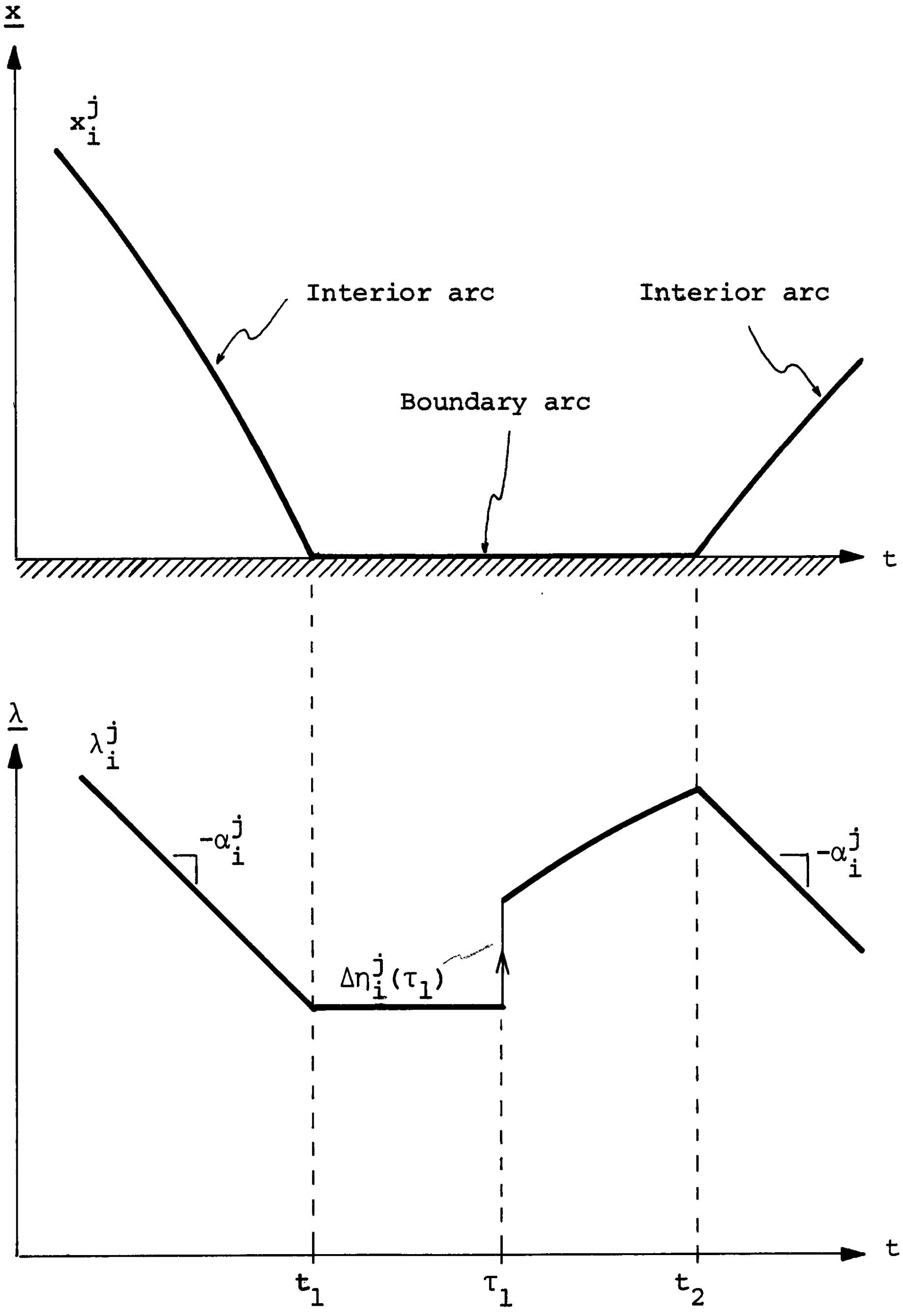


Figure 3.1 Example of State-Costate Trajectory Pair

$$\Delta\eta_i^j(\tau_p) \leq 0 \tag{3.35}$$

and from equation (3.9) we have

$$\Delta\lambda_i^j(\tau_p) = \lambda_i^j(\tau_p^+) - \lambda_i^j(\tau_p^-) = -\Delta\eta_i^j(\tau_p) \geq 0. \tag{3.36}$$

That is, jumps in the costate are always positive. The above situations are depicted in Figure 3.1 for a possible state-costate trajectory pair.

The times at which the state enters or exits from a boundary arc are called <u>boundary junctions</u>. Note that the minimum value which the slope of the costate may attain on or off a boundary arc is $-\alpha_i^j$. Aside from jumps, a characteristic of the costate which distinguishes the state constrained problem is the possible non-uniqueness of the costate for a given optimal trajectory. As such, <u>it is clear that the costates may not be interpreted as the partial derivative of the optimal cost with respect to the state, commonly known as the influence function</u>. Furthermore, the non-uniqueness of the costate presents special implications for the feedback scheme to be developed. An example of this behavior and a discussion of its significance is presented in Section 3.3.4.2.

In general, any trajectory obtained from a set of necessary conditions is an extremal, and as such is merely a candidate for an optimal trajectory. Fortunately, in our problem it turns out that any such extremal trajectory is actually optimal, as is shown in the following theorem.

Theorem 3.2

The necessary conditions of Theorem 3.1 are sufficient.

Proof:

Let $\underline{x}^*(t)$, $\underline{u}^*(t)$, $\underline{\lambda}(t)$, $\underline{\eta}(t)$ satisfy the equations (3.2)-(3.6) and the necessary conditions (3.7)-(3.14) of Theorem 3.1. Also, let $\underline{x}(t)$, $\underline{u}(t)$ be any state and costate trajectory satisfying (3.2)-(3.6). Then

$$\delta J = J(\underline{x}) - J(\underline{x}^*)$$

$$= \int_{t_0}^{t_f} \underline{\alpha}^T(\tau)(\underline{x}(\tau) - \underline{x}^*(\tau))d\tau$$

$$= \int_{t_0}^{t_f} (-d\underline{\lambda}^T(\tau) - d\underline{\eta}^T(\tau))(\underline{x}(\tau) - \underline{x}^*(\tau))$$

by substituting from (3.9) and expanding obtain

$$= \int_{t_0}^{t_f} (-\underline{x}^T(\tau)d\underline{\lambda}(\tau) - \underline{x}^T(\tau)d\underline{\eta}(\tau) + \underline{x}^{*T}(\tau)d\underline{\lambda}(\tau) + \underline{x}^{*T}(\tau)d\underline{\eta}(\tau)).$$

From equation (3.16), $\int_{t_0}^{t_f} \underline{x}^{*T}(\tau)d\underline{\eta}(\tau) = 0$. We now integrate the first and third term in δJ by parts:

$$\int_{t_0}^{t_f} \underline{x}^T(\tau)d\underline{\lambda}(\tau) = \underline{\lambda}^T(t_f)\underline{x}(t_f) - \underline{\lambda}^T(t_0)\underline{x}(t_0)$$

$$- \int_{t_0}^{t_f} \underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau)d\tau$$

$$\int_{t_0}^{t_f} \underline{x}^{*T}(\tau)d\underline{\lambda}(\tau) = \underline{\lambda}^T(t_f)\underline{x}^*(t_f) - \underline{\lambda}^T(t_0)\underline{x}^*(t_0)$$

$$- \int_{t_0}^{t_f} \underline{\lambda}^T(\tau)\dot{\underline{x}}^*(\tau)d\tau.$$

Now, by (3.4)

$$\underline{x}(t_0) = \underline{x}^*(t_0) = \underline{x}_0 \quad \text{and}$$

$$\underline{x}(t_f) = \underline{x}^*(t_f) = \underline{0}.$$

Substituting these expressions we obtain

$$\delta J = \int_{t_0}^{t_f} (\underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau) - \underline{\lambda}^T(\tau)\dot{\underline{x}}^*(\tau))d\tau$$

$$- \int_{t_0}^{t_f} \underline{x}^T(\tau)d\underline{\eta}(\tau)$$

$$= \int_{t_0}^{t_f} \underline{\lambda}^T(\tau)\underline{B}(\underline{u}(\tau) - \underline{u}^*(\tau))d\tau$$

$$- \int_{t_0}^{t_f} \underline{x}^T(\tau)d\underline{\eta}(\tau).$$

But by (3.8)

$$\int_{t_0}^{t_f} \underline{\lambda}^T(\tau)\underline{B}(\underline{u}(\tau) - \underline{u}^*(\tau))d\tau \geq 0$$

and since $\underline{x}(\tau) \geq 0$ and $d\underline{\eta}(\tau) \leq 0$ we have

$$\int_{t_0}^{t_f} \underline{x}^T(\tau)d\underline{\eta}(\tau) \leq 0.$$

Therefore $\delta J \geq 0 \quad \forall\, \underline{u}(\cdot) \in \mathcal{U},\ \underline{x}(\cdot) \geq 0.$ ■ Theorem 3.2

Although it is not immediately apparent how one may utilize the necessary conditions to obtain optimal solutions (this if most often the case), certain fundamental characteristics of optimal solutions may be deduced immediately from the form of these conditions. We begin this discussion by noting that finding the optimal control function $\underline{u}^*(\cdot)$ reduces from (3.8) to solving at every time $\tau \in [t_0, t_f]$ the linear program with decision vector $\underline{u}(t)$:

$$\underline{u}^*(\tau) = \underset{\underline{u}(\tau)\in\mathcal{U}}{\text{ARG MIN}} \; [\underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau)]$$

$$= \underset{\underline{u}(\tau)\in\mathcal{U}}{\text{ARG MIN}} \; [\underline{\lambda}^T(\tau)\underline{B}\ \underline{u}(\tau)] \tag{3.37}$$

$$\mathcal{U}\begin{cases} \underline{D}\ \underline{u} \leq \underline{C} \\ \underline{u} \geq 0 \end{cases}$$

This is a fortuitous situation, since much is known about characterizing and finding solutions of linear programs. We know, for instance, that optimal solutions always lie on the boundary of the convex polyhedral constraint region $\mathcal{U}$.

We now proceed to represent the solution to the linear program (3.37) for the specific form of the matrices $\underline{B}$ and $\underline{D}$ which correspond to our network problem. The minimization can actually be performed on <u>one link at a time</u>. Consider the link (i,k) and a possible set of associated controls

$$u_{ik}^{1},\ u_{ik}^{2},\ \ldots,\ u_{ik}^{i-1},\ u_{ik}^{i+1},\ \ldots,\ u_{ik}^{N}.$$

A given control may appear in one of the two following ways:

1) u_{ik}^{j} enters into exactly two state equations:

$$\begin{aligned} \dot{x}_i^j(t) &= -u_{ik}^j(t) + \ldots + a_i^j \\ \dot{x}_k^j(t) &= +u_{ik}^j(t) + \ldots + a_k^j \end{aligned} \tag{3.38}$$

2) u_{ik}^{k} enters into exactly one state equation:

$$\dot{x}_i^k(t) = -u_{ik}^k + \ldots + a_i^k \tag{3.39}$$

Hence, all controls on link (i,k) contribute the following terms to $\underline{\lambda}^T\underline{B}\ \underline{u}$:

$$\begin{aligned} & (\lambda_k^1(t) - \lambda_i^1(t))u_{ik}^1(t) \\ + \quad & (\lambda_k^2(t) - \lambda_i^2(t))u_{ik}^2(t) \\ + \ldots + \ & (\lambda_k^{i-1}(t)-\lambda_i^{i-1}(t))u_{ik}^{i-1}(t) \\ + \ & (\lambda_k^{i+1}(t)-\lambda_i^{i+1}(t))u_{ik}^{i+1}(t) \\ + \ldots + \ & (\lambda_k^N(t) - \lambda_i^N(t))u_{ik}^N(t) \end{aligned} \tag{3.40}$$

where $\lambda_k^k(t) = 0$. Equations (2.4), (3.37) and (3.40) determine

The optimal control law at time t on (i,k):

- $u_{ik}^{\ell}(t) = C_{ik}$ and $u_{ik}^{j}(t) = 0 \quad \forall\, j \neq \ell$ (3.41)

  if $(\lambda_k^{\ell}(t) - \lambda_i^{\ell}(t)) < (\lambda_k^{j}(t) - \lambda_i^{j}(t)) \quad \forall\, j \neq \ell$

  and $(\lambda_k^{\ell}(t) - \lambda_i^{\ell}(t)) < 0$.

- $u_{ik}^{\ell}(t) + u_{ik}^{\ell+1}(t) + \ldots + u_{ik}^{m}(t) = C_{ik}$ (3.42)

  $u_{ik}^{j}(t) = 0 \quad \forall\, j \not\subset [\ell,\ell+1,\ldots,m]$

  if $(\lambda_k^{\ell}(t) - \lambda_i^{\ell}(t)) = (\lambda_k^{\ell+1}(t) - \lambda_i^{\ell+1}(t)) = \ldots =$

  $= (\lambda_k^{m}(t) - \lambda_i^{m}(t)) < (\lambda_k^{j}(t) - \lambda_i^{j}(t))$

  $\forall\, j \not\subset [\ell, \ell+1,\ldots, m]$

  and $(\lambda_k^{\ell}(t) - \lambda_i^{\ell}(t)) = (\lambda_k^{\ell+1}(t)-\lambda_i^{\ell+1}(t)) = \ldots =$

  $= (\lambda_k^{m}(t) - \lambda_i^{m}(t)) < 0$.

- $u_{ik}^{\ell}(t) + u_{ik}^{\ell+1}(t) + \ldots + u_{ik}^{m}(t) \leq C_{ik}$ (3.43)

  $u_{ik}^{j}(t) = 0 \quad \forall\, j \not\subset [\ell,\ell+1,\ldots,m]$

  if $(\lambda_k^{\ell}(t)-\lambda_i^{\ell}(t)) = (\lambda_k^{\ell+1}(t) - \lambda_i^{\ell+1}(t)) = \ldots =$

  $= (\lambda_k^{m}(t) - \lambda_i^{m}(t)) = 0$

  and $(\lambda_k^{j}(t) - \lambda_i^{j}(t)) > 0 \quad \forall\, j \not\subset [\ell, \ell+1, \ldots, m]$.

The actual computation of the optimal control at time t requires knowledge of $\underline{\lambda}(t)$, which in turn requires knowledge of the optimal state for time greater than or equal to t. This is the central difficulty in the application of necessary conditions in the determination of a feedback solution. The remainder of this thesis is essentially devoted to demonstrating how this difficulty may be surmounted for our problem.

However, we may immediately deduce several properties of the optimal control from examining the general form of (3.41)-(3.43). It is evident that the quantities which determine the optimal controls are the coefficients of the form $(\lambda_k^j(t) - \lambda_i^j(t))$ which multiply the control $u_{ik}^j(t)$. The only situation under which it is ever optimal to have $u_{ik}^j(t)$ strictly positive is if $(\lambda_k^j(t) - \lambda_i^j(t)) \leq 0$. In words, this condition says that it is optimal to send messages with destination j from node i to node k at time t only if the costate associated with $x_i^j(t)$ is greater than or equal to that associated with $x_k^j(t)$. This fact gives rise to a very interesting analogy: The flow of messages in a data communications network can be roughly visualized as the frictionless flow of fluids in a network of pipes with storage at tanks in the nodes. We may then relate the costates $\lambda_k^j(t)$ and $\lambda_i^j(t)$ to the pressures at the storage areas x_k^j and x_i^j respectively, and the difference $(\lambda_k^j(t) - \lambda_i^j(t))$ to the associated pressure difference. In this vein, we shall refer to $(\lambda_k^j(t) - \lambda_i^j(t))$ as the costate difference which exists at time t between node i and node k and is associated with the traffic going to destination j. It is therefore optimal to send messages of a given destination from one node to an adjacent node only if the costate difference is negative or zero.

According to equation (3.41), if the costate difference associated with destination j on link (i,k) is strictly negative and less than the remaining costate differences on this link, then the optimal control is $u_{ik}^{j}(t) = C_{ik}$ and all other controls are zero. However, equations (3.42) and (3.43) reveal that when two or more costate differences on the same link are non-positive and equal, the associated optimal control will not be uniquely determined. In these situations, the optimal solution set is in fact infinitely large. Such non-uniqueness is a fundamental property of linear programs.

3.2.2 Controllability to Zero for Constant Inputs

Henceforth we shall be considering only the situation in which all the inputs are constant functions of time over the interval of interest $t \in [t_0, t_f]$. The utility of this assumption for the representation of data network operation is discussed in Section 5.1. In this section, we present a simple theorem which characterizes all those inputs which allow the state to be driven to zero under given link capacity constraints. We begin with a definition:

Definition 3.1: We denote by

$$\dot{\mathcal{X}} = \{\dot{\underline{x}} \,|\, -\dot{\underline{x}} = \underline{B}\ \underline{u} \text{ and } \underline{u} \in \mathcal{U}\} \subset R^n \tag{3.44}$$

the *set of feasible flows* attainable through the available controls.

The reason for the negative sign in the definition of $\dot{\mathscr{X}}$ is basically one of notational convenience. We shall also refer to $\dot{\mathscr{X}}$ as the <u>$\dot{x}$-constraint figure.</u> Note that since $\mathcal{U}$ is a bounded convex polyhedron in $\mathcal{R}^m$ and $\underline{B}$ is a linear mapping from $\mathcal{R}^m$ to $\mathcal{R}^n$, then $\dot{\mathscr{X}}$ is a bounded convex polyhderon in $\mathcal{R}^n$.

<u>Theorem 3.3</u> (Controllability to zero, constant inputs)

All initial conditions of the system (3.3)-(3.6) are controllable to zero under constant inputs if and only if

$$\underline{a} \in \text{Int}(\dot{\mathscr{X}}) \qquad (\underline{a} \in \mathcal{R}^n, \dot{\mathscr{X}} \subset \mathcal{R}^n)$$

where Int$(\dot{\mathscr{X}})$ denotes the interior of the $\dot{x}$-constraint figure.

<u>Proof:</u> <u>Suppose $\underline{a} \in$ Int$(\dot{\mathscr{X}})$.</u>

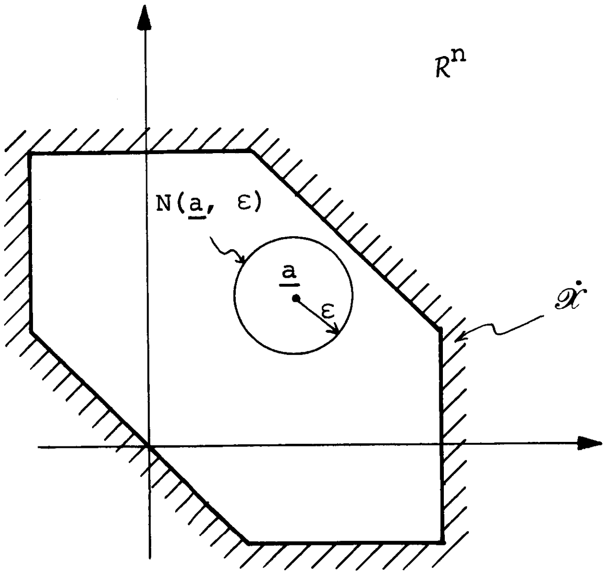


Figure 3.2 $\dot{\underline{x}}$-Constraint Figure

We shall proceed by showing that it is possible to pick a control corresponding to a feasible flow which simultaneously brings all components of any initial state $\underline{x}(0)$ to zero.

There is a neighborhood $N(\underline{a}, \varepsilon)$ which lies entirely in $\mathrm{Int}(\dot{\mathscr{X}})$ for $\varepsilon > 0$ sufficiently small. See Figure 3.2. Consequently, we may pick a constant control $\underline{u}'$ such that for all x_i^j

$$-\underline{b}_i^j \underline{u}' = a_i^j + \varepsilon \frac{x_i^j(0)}{||\underline{x}(0)||}$$

where $\underline{b}_i^j$ is the row of $\underline{B}$ corresponding to x_i^j, and $x_i^j(0)$ is the corresponding component of $\underline{x}(0)$. The norm is in the Euclidean sense. Applying the control $\underline{u}'$ we obtain

$$\dot{x}_i^j(t) = \underline{b}_i^j \underline{u}' + a_i^j = -\varepsilon \frac{x_i^j(0)}{||\underline{x}(0)||}$$

Since $\varepsilon > 0$, all components of $\underline{x}(0)$ are brought to zero in the time

$$T = \frac{x_i^j(0)}{\dot{x}_i^j} = \frac{||\underline{x}(0)||}{\varepsilon} .$$

On the other hand, suppose that $\underline{a \notin \mathrm{Int}(\dot{\mathscr{X}})}$. Then $\underline{a}$ falls either on a boundary of $\dot{\mathscr{X}}$ or in the complement of $\dot{\mathscr{X}}$ with respect to R^n. It now must be shown that in either case there exists some initial condition which is not controllable to zero.

We begin with the case in which $\underline{a}$ falls at a boundary point of $\dot{\mathscr{X}}$; that is, $\underline{a}$ falls on a face of the convex polyhedron $\dot{\mathscr{X}}$. As a preliminary, we recognize the fact that since all of the components of $\underline{a}$ are non-negative, then the face of $\dot{\mathscr{X}}$ on which it falls lies at least partially in the non-negative orthant of R^n. It is not difficult to see that any face of $\dot{\mathscr{X}}$ which lies in the non-negative orthant of R^n has the following properties:

(i) Consider the hyperplane which contains the face. Then the coefficients of the equation of the hyperplane are all non-negative; that is, the unit normal vector to the face has all non-negative components.

(ii) At least one of the components of the unit normal vector is strictly positive.

Let $\underline{n}$ denote the unit normal vector to the particular face of $\dot{\mathscr{X}}$ on which $\underline{a}$ lies. See Figure 3.3.

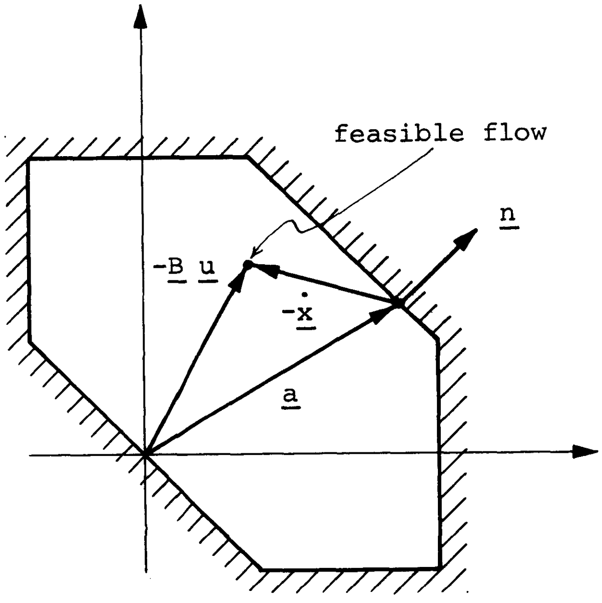


Figure 3.3 Attainable Flows in the Presence of Constant Inputs

Now, since $-\dot{\underline{x}} = -\underline{B}\ \underline{u} - \underline{a}$ it is evident from Figure 3.3 that a given flow $-\dot{\underline{x}}$ can be realized by a feasible flow $-\underline{B}\ \underline{u}$ with inputs $\underline{a}$ if and only if

$$-\dot{\underline{x}} \cdot \underline{n} \leq 0. \tag{3.45}$$

We now proceed to show that if $\underline{a}$ lies at a boundary point of $\dot{\mathscr{X}}$, then we can always find some initial condition which cannot be brought to zero with the available controls. To this end, consider the following subset of R^n:

$$\psi = \{\underline{x} \in R^n | x_i^j = 0 \quad \text{if } n_i^j > 0\}$$

where n_i^j is the component of $\underline{n}$ corresponding to x_i^j. Now, choose <u>any</u> initial condition such that $\underline{x}(t_0) \not\in \psi$. Clearly such a point always exists. We now shall show by contradiction that any such initial condition can never be brought to the final state $\underline{x}(t_f) = \underline{0}$. Suppose that the state can be brought to zero at some time t_f. Then the state must hit ψ at some time $\tau' \leq t_f$. Therefore, at some time $\tau'' < \tau'$ we must have $\dot{x}_k^\ell(\tau'') < 0$ for at least one component x_k^ℓ of the state vector such that $n_k^\ell > 0$. The existence of such a component is guaranteed by property (ii) above. Also, all of the other components of $\underline{x}$ associated with positive components of $\underline{n}$ must be such that they have non-positive time derivatives at τ''. Hence, since by (i) above <u>all</u> components of $\underline{n}$ are non-negative we have

$$-\dot{\underline{x}}(\tau'') \cdot \underline{n} = \sum_{\substack{i,j \in N \\ j \neq i}} -\dot{x}_i^j(\tau'') n_i^j > 0.$$

The above inequality contradicts (3.45) and therefore any $x(t_0) \not\subset \psi$ cannot be brought to zero.

Finally, consider the case in which $\underline{\underline{a} \in R^n/\dot{\mathscr{X}}.}$ Then it is easily seen that there is at least one component of the state which increases without bound for any $\underline{u} \in \mathcal{U}$. Therefore, no initial condition is controllable to zero in this case and the theorem is proved.

■ Theorem 3.3.

Controllability to zero is certainly a necessary condition for the existence of an optimal solution to (3.1)-(3.6), although not in itself sufficient. However, as our feedback synthesis technique will be based upon trajectories which satisfy the necessary conditions, we are guaranteed of their optimality by the sufficiency Theorem 3.2. We shall therefore have no need of an explicit proof of the existence of an optimizing control.

3.2.3 Geometrical Characterization of the Feedback Space for Constant Inputs

In what follows, we shall assume that the controllability to zero condition of Theorem 3.3 is satisfied. The following is a consequence of the optimal control characterization (3.41)-(3.43), and as such is a corollary to Theorem 3.1.

Corollary 3.1 If $\underline{a}$ = constant, then there always exists a solution to (3.1)-(3.6) with controls piecewise constant in time and trajectory with piecewise constant slopes.

Proof:

For a given costate trajectory $\underline{\lambda}(t)$, $t \in [t_0, t_f]$, the optimal control is given by (3.41)-(3.43). For those controls with unique optimal values, the determining factor is the sign and relative magnitude of the associated costate difference. Over periods of time for which these factors remain the same, the optimal control is constant and is given by $u_{ik}^{\ell} = C_{ik}^{\ell}$ or $u_{ik}^{\ell} = 0$. Those controls not uniquely determined over certain periods of time (as in situations (3.42) and (3.43)) may achieve any values which satisfy certain constraints, such as $\sum_{j=\ell}^{m} u_{ik}^{j} = C_{ik}$ or $\sum_{j=\ell}^{m} u_{ik}^{j} \leq C_{ik}$. The only additional requirement on these controls may be that they maintain certain states on boundary arcs over certain periods of time. Since the inputs are constant, any such requirement may always be satisfied by constant valued controls over the appropriate time period. As the controls have been shown to be piecewise constant and the inputs are constant, it follows immediately from equation (3.3) that the associated optimal trajectories have piecewise constant slopes. ■ Corollary 3.1

We note in passing that the above corollary also holds for piecewise constant inputs.

The solution to the constant input problem is of the bang-bang variety in that the optimal control switches intermittently among boundary points of $\mathcal{U}$. Also, in situations when one or more costate differences are zero or several are negative and equal, the control is termed <u>singular</u>. Under such circumstances, the optimal control is not determined uniquely from the laws (3.41)-(3.43). In the solution technique to be presented, this non-uniqueness will play a major role.

Owing to the bang-bang nature of the control, every optimal trajectory may be characterized by a finite number of parameters. We now present a compact set of notation for specifying these parameters.

<u>Definition 3.2</u>

$$U(\underline{x}) \overset{\Delta}{=} \{\underline{u}_0, \underline{u}_1, \underline{u}_2, \ldots, \underline{u}_{f-1}\} \tag{3.46}$$

$$\text{and} \qquad T(\underline{x}) \overset{\Delta}{=} \{t_0, t_1, \ldots, t_f\}$$

are a <u>sequence of optimal controls</u> and associated <u>control switch time sequence</u> which bring the state $\underline{x}$ optimally to $\underline{0}$ on $t \in [t_0, t_f]$, where $\underline{u}_p$ is the optimal control on $t \in [t_p, t_{p+1})$, $p \in [0, 1, \ldots, f-1]$.

When dealing with the necessary conditions, an additional property of a given trajectory which shall be of interest is which states travel on boundary arcs and over what periods of time. This information is summarized in the following definitions:

<u>Definition 3.3</u>

$$\mathcal{B}_p \overset{\Delta}{=} \{x_i^j \mid x_i^j(\tau) = 0 \quad \forall\, \tau \in [t_p, t_{p+1})\} \tag{3.47}$$

is the set of state variables travelling on boundary arcs during the application of $\underline{u}_p$.

Definition 3.4

$$\mathcal{B}(\underline{x}) = \{\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_{f-1}\} \tag{3.48}$$

is the sequence of sets $\mathcal{B}_p$ corresponding to the application of $U(\underline{x})$ on $T(\underline{x})$. $\mathcal{B}(\underline{x})$ is referred to as the boundary sequence.

In the forthcoming development, we shall be interested in regions of the state space which have the following property: when we consider every point of the region to be an initial condition of the optimal control problem, a common optimal control sequence and a common associated boundary sequence apply to all points. Formally, we define the following subset of R^n:

Definition 3.5

$$\mathscr{C}(U, \mathcal{B}) \triangleq \{\underline{x} \in R^n | U(\underline{x}) = U \text{ and } \mathcal{B}(\underline{x}) = \mathcal{B}\}. \tag{3.49}$$

Note that since any two distinct members of $\mathscr{C}(U, \mathcal{B})$ have associated with them the same sequence of optimal controls, the number of switch times in their respective sequences is the same. However, the values of the particular switch times will in general be different. Suppose $\underline{x}_1 \in \mathscr{C}(U, \mathcal{B})$ and $\underline{x}_2 \in \mathscr{C}(U, \mathcal{B})$. Then the switching time sequences are distinguished with the following notation:

Notation 3.1

$$\begin{aligned} T(\underline{x}_1) &= \{t_0', t_1', \ldots, t_f'\} \\ T(\underline{x}_2) &= \{t_0'', t_1'', \ldots, t_f''\}. \end{aligned} \tag{3.50}$$

The above definitions are illustrated pictorially in Figure 3.4.

It may appear at first that the regions $\mathcal{C}(U, \mathcal{B})$ defined above are too special to be of any utility. However, we shall soon demonstrate that such regions are related to the basic building blocks of the feedback space. With this as motivation, we begin by providing a fundamental geometric property of these regions.

Theorem 3.4 (Geometrical characterization)

Let the inputs $\underline{a}$ = constant. Let U and $\mathcal{B}$ be control and boundary sequences respectively as defined in (3.46)-(3.48). Also, let $\mathcal{C}(U, \mathcal{B})$ be the subset of $\mathcal{R}^n$ as defined in (3.49). Then $\mathcal{C}(U, \mathcal{B})$ is a convex polyhedral cone in $\mathcal{R}^n$.

Proof: The basic elements are embodied in the following lemmas:

Lemma 3.1. Let the inputs $\underline{a}$ = constant. Suppose $\underline{x}_1 \epsilon \mathcal{R}^n$ and $\underline{x}_2 \epsilon \mathcal{R}^n$. If $\underline{x}_2 = \gamma \underline{x}_1$, where γ is a scalar $\gamma > 0$, then

$$U(\underline{x}_2) = U(\underline{x}_1)$$

$$\text{and} \quad \mathcal{B}(\underline{x}_2) = \mathcal{B}(\underline{x}_1).$$

Proof of Lemma 3.1:

For the purpose of this proof, we shall consider time to be propagating backward from t_f.

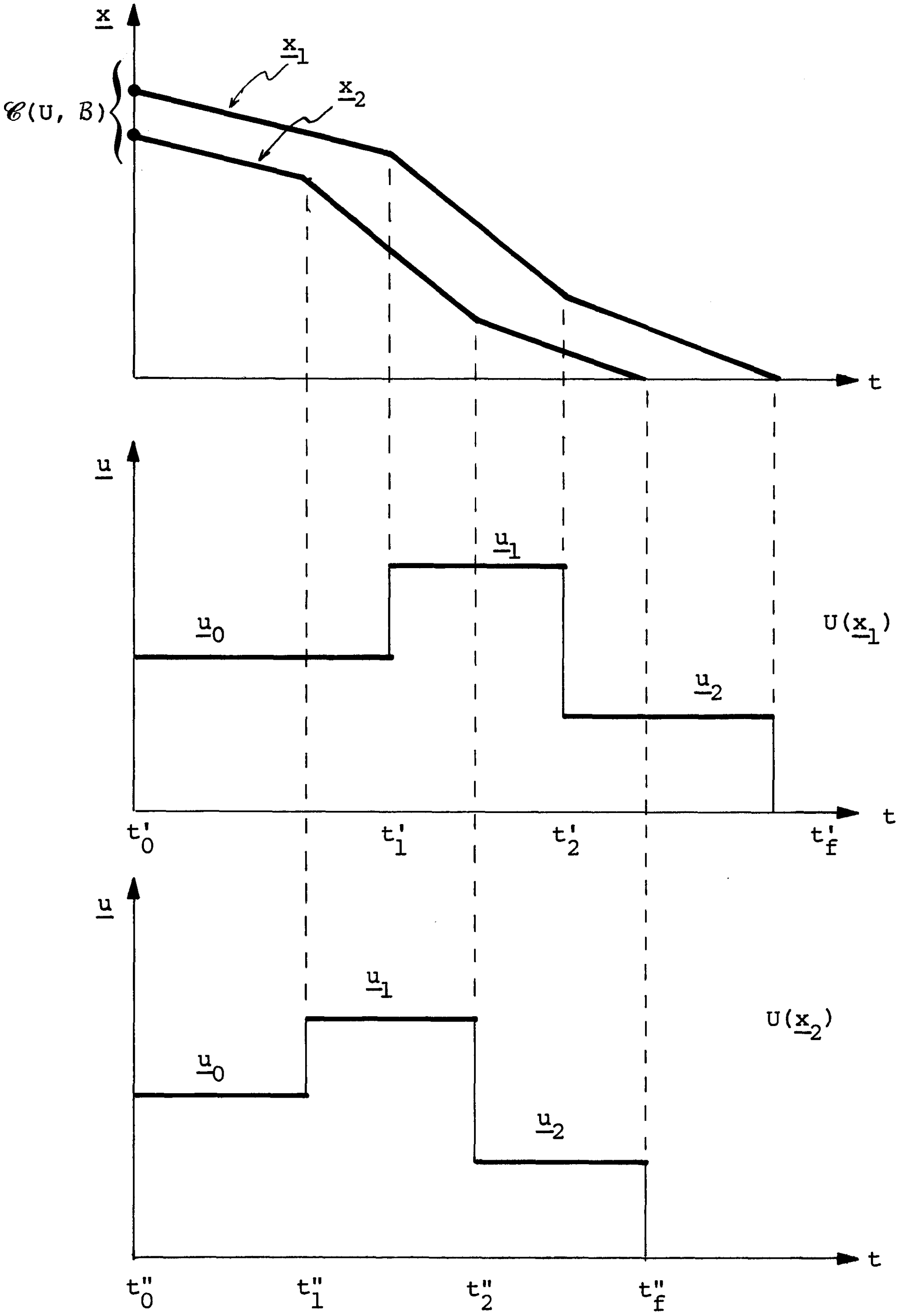


Figure 3.4 Illustration of $U(\underline{x})$, $T(x)$, $\mathscr{C}(U, \mathcal{B})$

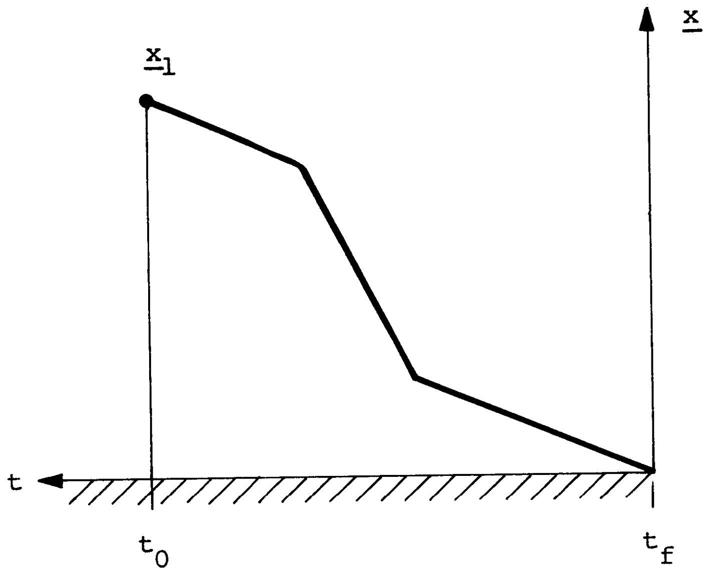


Figure 3.5 Time Convention for Proof of Lemma 3.1

Let the optimal solution which brings the state $\underline{x}(t_0) = \underline{x}_1$ to $\underline{x}(t_f) = \underline{0}$ be characterized by the control and switch time sequences:

$$U(\underline{x}_1) = \{\underline{u}_0, \underline{u}_1, \ldots, \underline{u}_{f-1}\}$$
$$T(\underline{x}_1) = \{t'_0, t'_1, \ldots, t'_f\}. \tag{3.51}$$

Associated with this optimal solution is the costate vector $\underline{\lambda}_1$ evolving in backward time according to

$$d\underline{\lambda}_1(\tau) = \underline{\alpha}d\tau + d\underline{\eta}_1(\tau)$$
$$\underline{\lambda}_1(t_f) = \underline{\nu}_1 \tag{3.52}$$
$$\tau \in [t'_f, t'_0].$$

<u>The proof shall consist of producing a switching time set</u> $T(\underline{x}_2)$ <u>and a costate history</u> $\underline{\lambda}_2(\tau)$ <u>for which</u> $U(\underline{x}_1)$ <u>is an optimal control sequence</u>

which brings $\underline{x}_2 = \gamma\underline{x}_1$ *to* $\underline{0}$. These sets are

$$T(\underline{x}_2) = \{t''_0, t''_1, \ldots, t''_f\} \tag{3.53}$$

$$\text{where} \quad t''_p = \gamma t'_p \tag{3.54}$$

$$p \in [0, 1, \ldots f]$$

$$d\underline{\lambda}_2(\tau) = \underline{\alpha} d\tau + d\underline{\eta}_2(\tau) \tag{3.55}$$

$$\text{where } \underline{\lambda}_2(t''_f) = \gamma\underline{\lambda}_1(t'_f) \tag{3.56}$$

$$\text{and} \quad d\underline{\eta}_2(\tau) = \gamma d\underline{\eta}_1(\tau/\gamma). \tag{3.57}$$

To show that indeed this set optimally brings $\underline{x}_2$ to $\underline{0}$ we have to show the following assertions:

(i) The application of $U(\underline{x}_1)$ on $T(\underline{x}_2)$ drives the state from $\underline{0}$ to $\underline{x}_2 = \gamma\underline{x}_1$ backward in time.

(ii) $\mathcal{B}(\underline{x}_2) = \mathcal{B}(\underline{x}_1)$.

(iii) $\underline{\eta}_2$ and $\underline{\lambda}_2$ satisfy all of the necessary conditions.

(iv) $U(\underline{x}_1)$ is optimal on $T(\underline{x}_2)$.

Step (i).

Since our system is time invariant, for convenience we set $t'_f = t''_f = t_f = 0$ as the initial time in the backward sense. Then if we denote by $x_1(\cdot)$ and $x_2(\cdot)$ the state trajectories resulting from the application of $U(\underline{x}_1)$ on $T(\underline{x}_1)$ and $T(\underline{x}_2)$ respectively, integration of the state dynamics backward in time yields

$$
\begin{aligned}
\underline{x}_2(\gamma t) &= -\int_0^{\gamma t} \dot{\underline{x}}_2(\tau)\,d\tau \\
&= -\int_0^{\gamma t} \dot{\underline{x}}_1(\tau/\gamma)\,d\tau \\
&= -\int_0^{t} \gamma\, \dot{\underline{x}}_1(s)\,ds, \quad s = \tau/\gamma, \\
&= \gamma \underline{x}_1(t) \qquad t \in [t_f, t_0'].
\end{aligned}
\tag{3.58}
$$

In particular, we have

$$\underline{x}_2(t_0'') = \underline{x}_2 = \gamma \underline{x}_1 = \gamma \underline{x}_1(t_0') . \tag{3.59}$$

Step (ii).

From (3.58) we see that if any component of $\underline{x}_1$ is zero on $[t_p', t_{p+1}')$, then the corresponding component of $\underline{x}_2$ is zero on $[t_p'', t_{p+1}'')$. Likewise, if any component of $\underline{x}_1$ is greater than zero on $[t_p', t_{p+1}')$, then the corresponding component of $\underline{x}_2$ is greater than zero on $[t_p'', t_{p+1}'')$.

Step (iii).

By virtue of the fact that $\mathcal{B}(\underline{x}_2) = \mathcal{B}(\underline{x}_1)$, it is easily verified that $d\underline{\eta}_2$ as defined in (3.57) satisfies the complementary slackness condition (3.10). Since $\underline{\lambda}_2$ is defined to satisfy a differential equation of the required form, and $d\underline{\eta}_2$ satisfies complementary slackness, then $\underline{\lambda}_2(\tau)$, $\tau \in [t_f, t_0']$, is a legitimate costate trajectory.

Step (iv).

The relationship between $\underline{\lambda}_1$ and $\underline{\lambda}_2$ backward in time is

$$\underline{\lambda}_2(\gamma t) = \int_{t_f}^{\gamma t} \underline{\alpha}\, d\tau + \int_{t_f}^{\gamma t} d\underline{\eta}_2(\tau) + \gamma\underline{\nu}_1$$

by (3.55) and (3.56)

$$= \int_{t_f}^{\gamma t} \underline{\alpha}\, d\tau + \gamma \int_{t_f}^{\gamma t} d\underline{\eta}_1(\tau/\gamma) + \gamma\underline{\nu}_1$$

by (3.57)

$$= \gamma\int_{t_f}^{t} \underline{\alpha}\, ds + \gamma \int_{t_f}^{t} d\underline{\eta}_1(s) + \gamma\underline{\nu}_1$$

where $s = \tau/\gamma$

$$= \gamma\underline{\lambda}_1(t) \qquad \forall\ t \in [t_f, t_0']. \tag{3.60}$$

Definitions (3.51) imply

$$\underline{u}_p^* = \underset{\underline{u}\in U}{\text{ARG MIN}}\ (\underline{\lambda}_1^T(\tau)\ \underline{B}\ \underline{u})$$

$$\tau \in [t_p', t_{p+1}') \qquad p \in [0, 1, \ldots, f-1]$$

$$= \underset{\underline{u}\in U}{\text{ARG MIN}}\ (\gamma\underline{\lambda}_1^T(\tau)\underline{B}\ \underline{u})$$

since γ positive scalar

$$= \underset{\underline{u}\in U}{\text{ARG MIN}}\ (\underline{\lambda}_2^T(\gamma\tau)\underline{B}\ \underline{u}). \tag{3.61}$$

The above is equivalent to

$$\underline{u}_p^* = \text{ARG} \min_{\underline{u} \in \mathcal{U}} \ (\underline{\lambda}_2^T(\tau) \underline{B}\ \underline{u}) \tag{3.62}$$

$$\tau \in [t''_p, t''_{p+1}), \ p \in [0, 1, \ldots, f-1].$$

Therefore, our supposition regarding the optimality of $U(\underline{x}_1)$ on $T(\underline{x}_2)$ is verified. Hence, the application of $U(\underline{x}_1)$ on $T(\underline{x}_2) = \{t''_0, t''_1, \ldots, t''_f\}$ forward in time brings the state from $\underline{x}_2(t''_0) = \gamma \underline{x}_1$ to $\underline{x}_2(t''_f) = \underline{0}$.

■ Lemma 3.1

Lemma 3.2. Let the inputs $\underline{a}$ = constant. Suppose for some $\underline{x}_1$, $\underline{x}_2 \in \mathcal{R}^n$, $\underline{x}_1 \geq 0$, $\underline{x}_2 \geq 0$, that

$$U(\underline{x}_1) = U(\underline{x}_2) = U$$

and

$$\mathcal{B}(\underline{x}_1) = \mathcal{B}(\underline{x}_2) = \mathcal{B}.$$

If
$$\underline{x}_3 = \gamma \underline{x}_1 + (1-\gamma)\underline{x}_2, \quad 0 \leq \gamma \leq 1,$$

then

$$U(\underline{x}_3) = U$$

and

$$\mathcal{B}(\underline{x}_3) = \mathcal{B}.$$

Proof of Lemma 3.2:

As in the proof of Lemma 3.1, we assume that time propagates backward from the final time. Let the optimal solution which brings the state from $\underline{x}_1$ to $\underline{0}$ be characterized by the control sequence and switch time set

$$U(\underline{x}_1) = U = \{\underline{u}_0, \underline{u}_1, \ldots, \underline{u}_{f-1}\}$$

$$T(\underline{x}_1) = \{t'_0, t'_1, \ldots, t'_f\} \tag{3.63}$$

respectively and similarly for $\underline{x}_2$:

$$U(\underline{x}_2) = U = \{\underline{u}_0, \underline{u}_1, \ldots, \underline{u}_{f-1}\}$$

$$T(\underline{x}_2) = \{t''_0, t''_1, \ldots, t''_f\}. \tag{3.64}$$

Associated with each of the above optimal solutions are costate vectors $\underline{\lambda}_1$ and $\underline{\lambda}_2$ which evolve backward in time according to

$$d\underline{\lambda}_1(\tau) = \underline{\alpha}d\tau + d\underline{\eta}_1(\tau) \qquad \underline{\lambda}_1(t'_f) = \underline{\nu}_1 \tag{3.65}$$

$$\tau \in [t'_f, t'_0]$$

$$d\underline{\lambda}_2(\tau) = \underline{\alpha}d\tau + d\underline{\eta}_2(\tau) \qquad \underline{\lambda}_2(t''_f) = \underline{\nu}_2 \tag{3.66}$$

$$\tau \in [t''_f, t''_0].$$

The proof proceeds along similar lines to the proof of the previous lemma. We shall produce a switching time set $T(\underline{x}_3)$ and a costate history $\underline{\lambda}_3(\tau)$ for which U is an optimal control sequence which brings $\underline{x}_2 = \gamma\underline{x}_1 + (1-\gamma)\underline{x}_2$ to $\underline{0}$. These sets are

$$T(\underline{x}_3) = \{t'''_0, t'''_1, \ldots, t'''_f\} \tag{3.67}$$

$$\text{where} \quad t'''_p = \gamma t'_p + (1-\gamma)t''_p \tag{3.68}$$

$$p \in [0, 1, \ldots, f]$$

$$d\underline{\lambda}_3(\tau) = \underline{\alpha}d\tau + d\underline{\eta}_3(\tau) \tag{3.69}$$

$$\text{where} \quad \underline{\lambda}_3(t'''_f) = \gamma\underline{\lambda}_1(t'_f) + (1-\gamma)\underline{\lambda}_2(t''_f) \tag{3.70}$$

and

$$d\underline{\eta}_3(t'''_p + \tau) = \gamma \, d\underline{\eta}_1(t'_p + \frac{\Delta t'_p}{\Delta t'''_p}\tau)$$

$$+ (1-\gamma) \, d\underline{\eta}_2(t''_p + \frac{\Delta t''_p}{\Delta t'''_p}\tau) \qquad (3.71)$$

$$\tau \in [0, \Delta t'''_p), \; p \in [0, 1, \ldots, f)$$

where we denote

<u>Notation 3.2</u>

$$\left.\begin{aligned} \Delta t'_p &= t'_p - t'_{p-1} \\ \Delta t''_p &= t''_p - t''_{p-1} \\ \Delta t'''_p &= t'''_p - t'''_{p-1} \end{aligned}\right\} \forall\, p \in [0,1,\ldots,f]. \qquad (3.72)$$

From equations (3.72) and (3.68) we obtain

$$\Delta t'''_p = \gamma \Delta t'_p + (1-\gamma)\Delta t''_p. \qquad (3.73)$$

To show that indeed the above set optimally brings $\underline{x}_3$ to $\underline{0}$ we have to show the following assertions:

(i) The application of U on $T(\underline{x}_3)$ drives the state from $\underline{0}$ to $\underline{x}_3 = \gamma\underline{x}_1 + (1-\gamma)\underline{x}_2$ backward in time.

(ii) $\mathcal{B}(\underline{x}_3) = \mathcal{B}$.

(iii) $\underline{\eta}_3$ and $\underline{\lambda}_3$ satisfy all of the nececessary conditions.

(iv) U is optimal on $T(\underline{x}_3)$.

Step (i).

Since our system is time invariant, we set $t'_f = t''_f = t'''_f = t_f$ as the initial time in the backward sense. We now denote by $\underline{x}_1(\cdot)$, $\underline{x}_2(\cdot)$ and $\underline{x}_3(\cdot)$ the state trajectories resulting from the application of U on $T(\underline{x}_1)$, $T(\underline{x}_2)$ and $T(\underline{x}_3)$ respectively and designate $\dot{\underline{x}}_q = \underline{B}\ \underline{u}_q + \underline{a}$. Integration of the state dynamics backward in time yields

$$\underline{x}_3(t'''_p + t) = -\sum_{q=f-1}^{p+1} \dot{\underline{x}}_{q-1}\Delta t'''_q - \dot{\underline{x}}_{p-1}t$$

$$= -\sum_{q=f-1}^{p+1} \dot{\underline{x}}_{q-1}(\gamma\Delta t'_q+(1-\gamma)\Delta t''_q)-\dot{\underline{x}}_{p-1}t$$

by substituting (3.73)

$$= \gamma\underline{x}_1(t'_p) + (1-\gamma)\underline{x}_2(t''_p) - \dot{\underline{x}}_{p-1}t$$

$$t \in [0, \Delta t'''_p).$$

Also from (3.73) we have

$$\gamma \frac{\Delta t'_p}{\Delta t'''_p} + (1-\gamma) \frac{\Delta t''_p}{\Delta t'''_p} = 1$$

so that the above expression now becomes

$$\underline{x}_3(t'''_p + t) = \gamma\underline{x}_1(t'_p) + (1-\gamma)\underline{x}_2(t''_p)$$

$$- \dot{\underline{x}}_{p-1}(\gamma \frac{\Delta t'_p}{\Delta t'''_p} + (1-\gamma) \frac{\Delta t''_p}{\Delta t'''_p}) t$$

$$= \gamma\underline{x}_1(t'_p + \frac{\Delta t'_p}{\Delta t'''_p} t) + (1-\gamma)\underline{x}_2(t''_p + \frac{\Delta t''_p}{\Delta t'''_p} t)$$

$$t \in [0, \Delta t'''_p). \qquad (3.74)$$

In particular, we obtain the desired value at t'''_0 :

$$\underline{x}_3(t'''_0) = \underline{x}_3 = \gamma \underline{x}_1 + (1-\gamma)\underline{x}_2 = \gamma \underline{x}_1(t'_0) + (1-\gamma)\underline{x}_2(t''_0). \qquad (3.75)$$

<u>Step (ii).</u>

Since $\mathcal{B}(\underline{x}_1) = \mathcal{B}(\underline{x}_2) = \mathcal{B}$, if any component of $\underline{x}_1$ is zero on $[t'_p, t'_{p+1})$, then the corresponding component of $\underline{x}_2$ is zero on $[t''_p, t''_{p+1})$, and vice versa; also, from (3.74) we see that under this circumstance the corresponding component of $\underline{x}_3$ is zero on $[t'''_p, t'''_{p+1})$. Likewise for components strictly greater than zero. Hence, $\mathcal{B}(\underline{x}_3) = \mathcal{B}$.

<u>Step (iii).</u>

By virtue of the fact that $\mathcal{B}(\underline{x}_3) = \mathcal{B}$ it is easily verified that $d\underline{\eta}_3$ as defined in (3.71) satisfies the complementary slackness condition (3.10). Since $\underline{\lambda}_3$ is defined to satisfy a differential equation of the required form and $d\underline{\lambda}_3$ satisfies complementary slackness, then $\underline{\lambda}_3(\tau)$, $\tau \in [t_f, t'''_0]$, is a legitimate costate trajectory.

<u>Step (iv).</u>

Integrating the costate equation (3.69) backward in time on $t \in [0, \Delta t'''_p)$ we obtain

$$\underline{\lambda}_3(t'''_p + t) = \underline{\lambda}_3(t'''_p) + \int_{t'''_p}^{t} \underline{\alpha}\, d\tau + \int_{t'''_p}^{t} d\underline{\eta}_3\,(t_p + \tau)$$

$$= \underline{\lambda}_3(t'''_p) + \int_{t'''_p}^{t} \underline{\alpha}\, d\tau + \gamma \int_{t'''_p}^{t} d\underline{\eta}_1(t'_p + \frac{\Delta t'_p}{\Delta t'''_p}\tau) +$$

$$+ (1-\gamma) \int_{t'''_p}^{t} d\underline{\eta}_2 \left(t''_p + \frac{\Delta t''_p}{\Delta t'''_p} \tau\right).$$

Now perform the change of variables

$$s_1 = \frac{\Delta t'_p}{\Delta t'''_p} \tau$$

$$s_2 = \frac{\Delta t''_p}{\Delta t'''_p} \tau$$

and note that $\tau = \gamma s_1 + (1-\gamma) s_2$ by (3.73). Then

$$\underline{\lambda}_3(t'''_p + t) = \underline{\lambda}_3(t'''_p) + \int_{t'''_p}^{\tau=t} \underline{\alpha}\, d(\gamma s_1 + (1-\gamma) s_2)$$

$$+ \gamma \int_{t'''_p}^{s_1=\frac{\Delta t'_p}{\Delta t'''_p} t} d\underline{\eta}_1(t'_p + s_1) + (1-\gamma) \int_{t'''_p}^{s_2=\frac{\Delta t''_p}{\Delta t'''_p} t} d\underline{\eta}_2(t''_p + s_2)$$

$$= \underline{\lambda}_3(t'''_p) + \gamma \left[\int_{t'''_p}^{s_1=\frac{\Delta t'_p}{\Delta t'''_p} t} (\underline{\alpha}\, ds_1 + d\underline{\eta}_1(t'_p + s_1))\right]$$

$$+ (1-\gamma) \left[\int_{t'''_p}^{s_2=\frac{\Delta t''_p}{\Delta t'''_p} t} (\underline{\alpha}\, ds_2 + d\underline{\eta}_2(t''_p + s_2))\right] \qquad (3.76)$$

$$t \in [0, \Delta t'''_p].$$

From (3.65) and (3.66) we obtain by backward integration

$$\underline{\lambda}_1\left(t'_p + \frac{\Delta t'_p}{\Delta t'''_p} t\right) = \underline{\lambda}_1(t'_p) + \left[\int_{t'''_p}^{s_1=\frac{\Delta t'_p}{\Delta t'''_p} t} (\underline{\alpha}\, ds_1 + d\underline{\eta}_1(t'_p + s_1))\right]$$

$$(3.77)$$

and

$$\underline{\lambda}_2(t''_p + \frac{\Delta t''_p}{\Delta t'''_p} t) = \underline{\lambda}_2(t''_p) + \left[\int_{t'''_p}^{s_2 = \frac{\Delta t''_p}{\Delta t'''_p} t} (\underline{\alpha}\, ds_2 + d\underline{\eta}_2(t''_p + s_2)) \right]. \tag{3.78}$$

$$\tau \; \varepsilon \; [0, \Delta t'''_p).$$

Substituting (3.77) and (3.78) into (3.76) obtain

$$\begin{aligned} \underline{\lambda}_3(t'''_p + t) = {} & \underline{\lambda}_3(t'''_p) + \gamma\underline{\lambda}_1(t'_p + \frac{\Delta t'_p}{\Delta t'''_p} t) - \gamma\underline{\lambda}_1(t'_p) \\ & + (1-\gamma)\underline{\lambda}_2(t''_p + \frac{\Delta t''_p}{\Delta t'''_p} t) - (1-\gamma)\underline{\lambda}_2(t''_p) \end{aligned}$$

$$\tau \; \varepsilon \; [0, \Delta t'''_p). \tag{3.79}$$

Upon substitution of the end condition (3.70) at $t_f = t'_f = t''_f = t'''_f$, the costate propagates in the following fashion backward in time:

$$\underline{\lambda}_3(t'''_p + t) = \gamma\underline{\lambda}_1(t'_p + \frac{\Delta t'_p}{\Delta t'''_p} t) + (1-\gamma)\underline{\lambda}_2(t''_p + \frac{\Delta t''_p}{\Delta t'''_p} t) \tag{3.80}$$

$$\forall\, p \; \varepsilon \; [0, 1, \ldots, f], \quad t \; \varepsilon \; [0, \Delta t'''_p).$$

We now verify that for $\underline{\lambda}_3$ given by the differential equation (3.69) with end condition (3.70), U is an optimal control sequence on $T(\underline{x}_3)$. By the definitions of U, $T(\underline{x}_1)$ and $T(\underline{x}_2)$ we have

$$\underline{u}^*_p = \underset{\underline{u}\varepsilon\mathcal{U}}{\text{ARG MIN}} \; \underline{\lambda}_1^T(t'_p + t)\underline{B}\,\underline{u}, \quad t \; \varepsilon \; [0, \Delta t'_p) \tag{3.81}$$

and

$$\underline{u}^*_p = \underset{\underline{u}\varepsilon\mathcal{U}}{\text{ARG MIN}} \; \underline{\lambda}_2^T(t''_p + t)\underline{B}\,\underline{u}, \quad t \; \varepsilon \; [0, \Delta t''_p). \tag{3.82}$$

Equations (3.81) and (3.82) imply

$$\gamma \underline{\lambda}_1^T(t_p' + t)\underline{B}\,\underline{u}_p^* \leq \gamma \underline{\lambda}_1^T(t_p' + t)\underline{B}\,\underline{u}_p \qquad (3.83)$$

$$\forall\, \underline{u}_p \in \mathcal{U},\ t \in [0,\ \Delta t_p')$$

and

$$(1-\gamma)\underline{\lambda}_2^T(t_p'' + t)\underline{B}\,\underline{u}_p^* \leq (1-\gamma)\underline{\lambda}_2^T(t_p'' + t)\underline{B}\,\underline{u}_p \qquad (3.84)$$

$$\forall\, \underline{u}_p \in \mathcal{U},\ t \in [0,\ \Delta t_p'').$$

Equations (3.80), (3.83) and (3.84) give

$$\underline{\lambda}_3^T(t_p''' + t)\underline{B}\,\underline{u}_p^* \leq \underline{\lambda}_3^T(t_p''' + t)\underline{B}\,\underline{u}_p \qquad (3.85)$$

$$\forall\, \underline{u}_p \in \mathcal{U},\ t \in [0,\ \Delta t_p'''),$$

that is

$$\underline{u}_p^* = \text{ARG} \min_{\underline{u} \in \mathcal{U}} \underline{\lambda}_3^T(t_p''' + t),\quad t \in [0,\ \Delta t_p''') \qquad (3.86)$$

$$\forall\, p \in [0,\ 1,\ \ldots,\ f].$$

Therefore, with $\underline{\lambda}_3$ as defined, U is an optimal control sequence which when applied forward in time on $T(\underline{x}_3) = \{t_0''', t_1''', \ldots, t_f'''\}$ brings the state from $\underline{x}_3(t_0''') = \underline{x}_3$ to $\underline{x}_3(t_f''') = \underline{0}$. ■ Lemma 3.2

Lemma 3.1 implies that the regions $\mathscr{C}(U, \mathcal{B})$ are cones and Lemma 3.2 implies that these cones are convex. As each region is convex, the interface between any two adjacent regions must be a portion of a hyperplane through the origin which belongs to one region or the other (we assume for technical reasons that the number of regions is finite).

Therefore, we have established the convex polyhedral conical nature of the regions. ■ Theorem 3.4

In preparation for development of the feedback technique, we present the following corollary to Theorem 3.1 which narrows down the freedom of the costates at the final time.

Corollary 3.2 If any state, say x_i^k, is strictly positive on the final time interval $[t_{f-1}, t_f)$ of an optimal trajectory, then $\lambda_i^k(t_f) = 0$.

Proof:

Consider a specific state x_i^k satisfying the hypothesis. Then since $\dot{x}_i^k(\tau)$, $\tau \in [t_{f-1}, t_f]$, is constant by Corollary 3.1, we must have

$$\dot{x}_i^k(t_f) < 0.$$

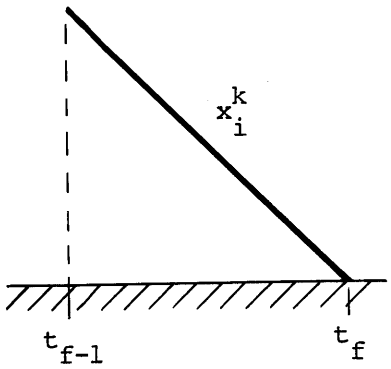


Figure 3.6 Typical State Variable Approaching Boundary at t_f

Then there must exist a directed chain of links from node i to node k (arbitrarily denote them by $\{(i, i+1), (i+1, i+2), \ldots, (k-1,k)\}$)

carrying some messages with destination k, that is

$$u^k_{i,i+1}(t_f) > 0,\ u^k_{i+1,i+2}(t_f) > 0,\ \ldots,\ u^k_{k-1,k}(t_f) > 0.$$

In order for the above control to be optimal, we must have (by (3.41)-(3.43))

$$\lambda^k_{i+1}(t_f) - \lambda^k_i(t_f) \leq 0$$

$$\lambda^k_{i+2}(t_f) - \lambda^k_{i+1}(t_f) \leq 0$$

$$\vdots$$

$$-\lambda^k_{k-1}(t_f) \leq 0.$$

From the above we conclude

$$\lambda^k_i(t_f) \geq \lambda^k_{k-1}(t_f) \geq 0.$$

We now proceed to show by contradiction that $\lambda^k_i(t_f) = 0$. Suppose $\lambda^k_i(t_f) > 0$. Then the transversality condition $\sum\limits_{i,j} \lambda^j_i(t_f)\dot{x}^j_i(t_f) = 0$ implies that there must be at least one $\dot{x}^\ell_i(t_f) < 0$ such that $\lambda^\ell_i(t_f) < 0$. But the above reasoning applied to x^ℓ_i implies that $\lambda^\ell_i(t_f) \geq 0$. Hence, a contradiction. ■ Corollary 3.2

3.3 Backward Construction of the Feedback Space for Constant Inputs

3.3.1 Introductory Examples of Backward Boundary Sequence Technique

The construction of regions of the type $\mathscr{C}(U, \mathcal{B})$ will play an important role in the synthesis of the feedback space. A basic observation with regard to these regions is that they are functions of the entire future sequence of controls which carry any member state optimally to zero. This general dependence of the current policy upon the future is the basic dilemma in computing optimal controls. This problem is often accommodated by the application of the principle of dynamic programming, which seeks to determine the optimal control as a function of the state by working backward from the final time. The algorithm to be developed employs the spirit of dynamic programming to enable construction of regions of the type $\mathscr{C}(U, \mathcal{B})$ from the appropriate set of optimal trajectories run backward in time. These trajectories are fashioned to satisfy the necessary and sufficient conditions of Theorem 3.1, as well as the costate boundary condition at t_f given in Corollary 3.2.

We motivate the backward construction technique with several two dimensional examples which introduce the basic principles involved. In fact, throughout the thesis important concepts will be illustrated with the simplest possible examples whenever the interests of clarity can be served. The reader is encouraged to study these examples carefully, as they comprise an essential part of the exposition. Also, one is able to grasp the general geometrical notions more readily with a simple two or three dimensional picture in mind.

Example 3.1

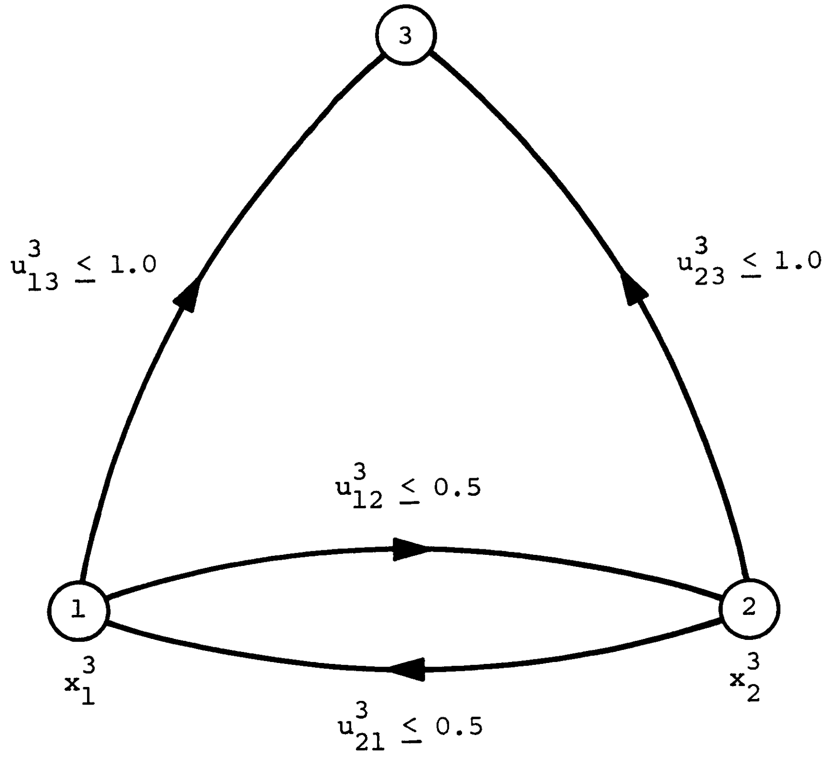


Figure 3.7 Simple Single Destination Network

The network as pictured in Figure 3.7 has a single destination, node 3; hence, we can omit the destination superscript "3" from the state and control variables without confusion. For simplicity, we assume that the inputs to the network are zero, so that the dynamics are:

$$\begin{aligned} \dot{x}_1(t) &= -u_{13}(t) - u_{12}(t) + u_{21}(t) \\ \dot{x}_2(t) &= -u_{23}(t) + u_{12}(t) - u_{21}(t) \end{aligned} \tag{3.87}$$

with control constraints as indicated in Figure 3.7. The cost function is defined as

$$D = \int_{t_0}^{t_f} \{x_1(t) + x_2(t)\}dt. \tag{3.88}$$

Let the vector notation be

$$\underline{x} \equiv \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \qquad \underline{u} \equiv \begin{pmatrix} u_{12} \\ u_{21} \\ u_{13} \\ u_{23} \end{pmatrix}.$$

We wish to find the optimal control which drives any state $\underline{x}(t_0) \geq \underline{0}$ to $\underline{x}(t_f) = \underline{0}$ while minimizing D.

As our intent is to work backward from the final time, we list all possible situations which may occur over the final time interval $[t_{f-1}, t_f]$ with respect to the states x_1 and x_2. Recall from Corollary 3.1 that the state rates $\dot{x}_1$ and $\dot{x}_2$ are constant on all intervals.

(i) $\dot{x}_1 < 0$

$x_2 = \dot{x}_2 = 0$

(ii) $\dot{x}_2 < 0$

$x_1 = \dot{x}_1 = 0$

(iii) $\dot{x}_1 < 0$

$\dot{x}_2 < 0.$

These are certainly the only situations which may occur on the final time interval, although we have no way of knowing a priori which are consistent with the optimality conditions. We do know by virtue of sufficiency that any trajectory which is optimal has associated with it costates which satisfy the necessary conditions; and that any trajectory which satisfies the necessary conditions is indeed optimal. Hence, our first task in the analysis of each of these cases shall be to attempt to produce a set of costates for which the control which solves the appropriate form of (3.37) gives rise to the specified trajectory on some non-zero interval of time preceding the final time. We now consider these situations one at a time:

(i) $\dot{x}_1(\tau) < 0,\ x_2(\tau) = \dot{x}_2(\tau) = 0,\quad \tau \in [t_{f-1}, t_f]$.

This situation is depicted as part of Figure 3.8 . We begin by considering the time period $[t_{f-1}, t_f]$ in a general sense without actually fixing the switching time t_{f-1}. This is simply the time period corresponding to the final bang-bang optimal control which brings the state to zero with $\dot{x}_1 < 0$ and $x_2 = \dot{x}_2 = 0$. We now set out to find if there is a costate satisfying the necessary conditions for which this situation is optimal; and if so, to find the value of the optimal control. The linear program to be solved on $\tau \in [t_{f-1}, t_f]$ is

$$\begin{aligned}\underline{u}^*(\tau) &= \text{ARG} \min_{\underline{u}\in U} [\lambda_1(\tau)\dot{x}_1(\tau) + \lambda_2(\tau)\dot{x}_2(\tau)] \\ &= \text{ARG} \min_{\underline{u}\in U} [(\lambda_2(\tau)-\lambda_1(\tau))u_{12}(\tau) + (\lambda_1(\tau)-\lambda_2(\tau))u_{21}(\tau) \\ &\qquad -\lambda_1(\tau)u_{13}(\tau)-\lambda_2(\tau)u_{23}(\tau)]. \qquad (3.89)\end{aligned}$$

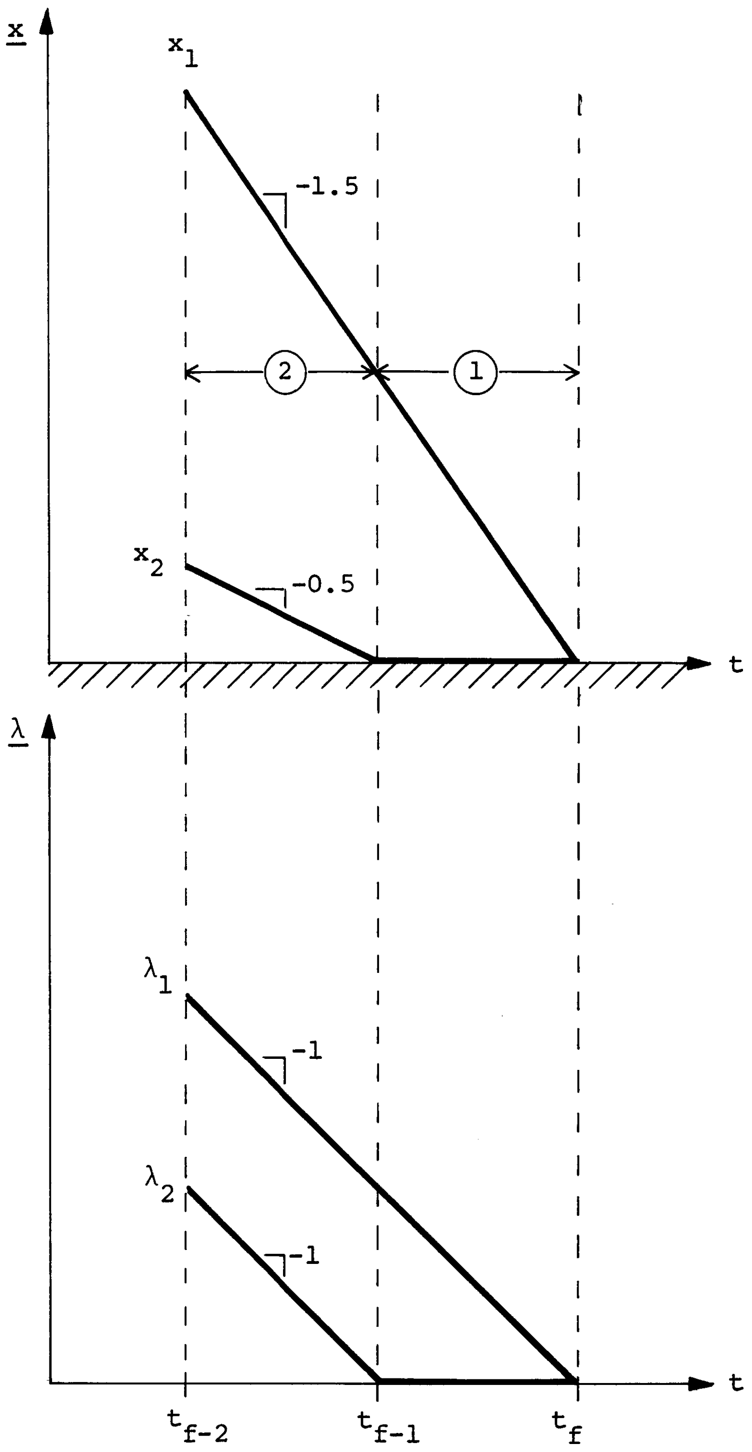


Figure 3.8 State-Costate Trajectory Pair for Example 3.1, Case (i)

Now, the stipulation $\dot{x}_1 < 0$ tells us from Corollary 3.2 that

$$\lambda_1(t_f) = 0 \tag{3.90}$$

and since x_1 is on an interior arc, equation (3.32) gives

$$\dot{\lambda}_1(\tau) = -1 \qquad \tau \in [t_{f-1}, t_f]. \tag{3.91}$$

This is shown in Figure 3.8 . Now, since we specify $x_2 = 0$ on this interval, its costate equation is

$$\begin{aligned} -d\lambda_2(\tau) &= 1\, d\tau + d\eta_2(\tau) \\ d\eta_2(\tau) &\leq 0 \\ \lambda_2(t_f) &= \nu_2 \quad \text{free} \qquad \tau \in [t_{f-1}, t_f] \end{aligned} \tag{3.92}$$

where η_2 is a possibly discontinuous function. We now submit that the costate value $\lambda_2(\tau) = \dot{\lambda}_2(\tau) = 0$, $\tau \in [t_{f-1}, t_f]$ is such that there exists a solution to (3.89) for which $\dot{x}_1 < 0$ and $\dot{x}_2 = 0$. The reader in fact may verify that $\lambda_2(\tau) = 0$, $\tau \in [t_{f-1}, t_f]$, is the only possible value since $\lambda_2(\tau) > 0$ or $\lambda_2(\tau) < 0$ imply by (3.89) and (3.87) that $\dot{x}_2 < 0$ and $\dot{x}_2 > 0$ respectively. Before presenting the specific solution we verify that $\lambda_2(\tau) = \dot{\lambda}_2(\tau) = 0$, $\tau \in [t_{f-1}, t_f]$, is a costate value which is consistent with equation (3.92). First, the required final condition $\lambda_2(t_f) = 0$ is acceptable since the necessary conditions leave it entirely free. Also, the differential equation $\dot{\lambda}_2(\tau) = 0$ may be realized with $d\eta_2(\tau) = -d\tau$, which is consistent with (3.92). This costate trajectory is pictured as part of Figure 3.8 . Now, with the costates so determined, one solution to (3.89) is

$$\underline{u}(\tau) = (0.5,\ 0.0,\ 1.0,\ 0.5)^T \tag{3.93}$$

$$\dot{x}_1(\tau) = -1.5 \qquad \dot{x}_2(\tau) = 0 \tag{3.94}$$

$$\tau \in [t_{f-1},\ t_f].$$

We emphasize that the above solution is only one among an infinite set of solutions to (3.89). However, it is the solution which we are seeking. We now make an important observation regarding this solution. Since $\dot{\lambda}_1(\tau) = -1$ and $\dot{\lambda}_2(\tau) = 0$ for $\tau \in [t_{f-1}, t_f)$, the control (3.93) remains optimal on $\tau \in (-\infty, t_f]$. But as $t_{f-1} \to -\infty$, $x_1(t_{f-1}) \to \infty$. Thinking now in forward time, this implies that any initial condition on the x_1-axis can be brought to zero optimally with the control specified in (3.94). <u>We have therefore determined the optimal feedback control for all points on the x_1-axis!</u> This is indicated in Figure 3.9.

Suppose now that we wish to consider a more general class of trajectories associated with the end condition under discussion. What we may do is to temporarily fix t_{f-1} and stipulate that the control on $[t_{f-2}, t_{f-1})$ has $\dot{x}_2$ negative; that is, insist that x_2 "leave the boundary" backward in time. As before, the initial time t_{f-2} of the segment $[t_{f-2}, t_{f-1})$ is left free. The program to be solved is (3.89) with $\tau \in [t_{f-2}, t_{f-1})$. Now, since x_1 is on an interior arc across t_{f-1}, by (3.32) its costate must be continuous across t_{f-1}, that is

$$\lambda_1(t_{f-1}^-) = \lambda_1(t_{f-1}^+) = t_f - t_{f-1}. \tag{3.95}$$

Since (3.92) allows for only positive jumps of λ_2 forward in time, we have

$$\lambda_2(t^-_{f-1}) = \lambda_2(t^+_{f-1}) = 0. \tag{3.96}$$

Also, since both x_1 and x_2 are on interior arcs on $[t_{f-2}, t_{f-1})$, equation (3.32) gives

$$\begin{aligned} \dot{\lambda}_1(\tau) &= -1 \\ \dot{\lambda}_2(\tau) &= -1 \end{aligned} \qquad \tau \in [t_{f-2}, t_{f-1}). \tag{3.97}$$

The resultant costate trajectory is depicted in Figure 3.8. We now perform the minimization (3.89) for $\tau \in [t_{f-2}, t_{f-1})$. Since $\lambda_1(\tau) > \lambda_2(\tau) > 0$, $\tau \in [t_{f-2}, t_{f-1})$, the solution to (3.89) is

$$\underline{u}(\tau) = (0.5, 0.0, 1.0, 1.0)^T \tag{3.98}$$

so that

$$\dot{x}_1(\tau) = -1.5 \qquad \dot{x}_2(\tau) = -0.5. \tag{3.99}$$

Therefore, the optimal control gives $\dot{x}_2(\tau) < 0$, which is the situation which we desire. Once again, we see that the control is optimal for $\tau \in (-\infty, t_{f-1}]$. Since $\dot{x}_1/\dot{x}_2 = 3.0$, upon leaving the x_1 axis backward in time the state travels parallel to the line $x_1 - 3x_2 = 0$ forever. Now, recall that t_{f-1} is essentially free. Therefore, from anywhere on the x_1 axis the state leaves parallel to $x_1 - 3x_2 = 0$ with underlying optimal control (3.98). Thinking now in forward time, this implies that any initial condition lying in the region between the line $x_1 - 3x_2 = 0$ and the x_1-axis (not including the x_1-axis) may be brought optimally to the x_1-axis with the control (3.98). See

Figure 3.9. Once the state reaches the x_1-axis, the optimal control which subsequently takes the state to zero is given by (3.93). Therefore, we have now managed to fill up the space between the line $x_1 - 3x_2 = 0$ and the x_1-axis with optimal controls!

The reader may now see that the feedback control regions just derived are the convex polyhedral cone of Theorem 3.4. For instance, the shaded region of Figure 3.9 is the half open cone

$$\mathscr{C}(U, \mathcal{B}) = \{\underline{x} | 0 < x_2 \leq \frac{x_1}{3}\} \tag{3.100}$$

where
$$U = \{\underline{u}_{f-2}, \underline{u}_{f-1}\}$$
$$\underline{u}_{f-2} = (0.5, 0.0, 1.0, 1.0)^T$$
$$\underline{u}_{f-1} = (0.5, 0.0, 1.0, 0.5)^T$$

and
$$\mathcal{B} = \{\mathcal{B}_{f-2}, \mathcal{B}_{f-1}\}$$
$$\mathcal{B}_{f-2} = \{\phi\}$$
$$\mathcal{B}_{f-1} = \{x_2\}.$$

Note that $\underline{u}_{f-2}$ is the control which is actually applicable for $\underline{x} \in \mathscr{C}(U, \mathcal{B})$. The entire sequence $U = \{\underline{u}_{f-2}, \underline{u}_{f-1}\}$ was needed to construct $\mathscr{C}(U, \mathcal{B})$.

(ii) $\dot{x}_2(\tau) < 0,\ x_1(\tau) = \dot{x}_1(\tau) = 0,\ \tau \in [t_{f-1}, t_f)$.

This situation is the same as (i) with the roles of x_1 and x_2 simply reversed. If we let x_2 leave the boundary first backward in

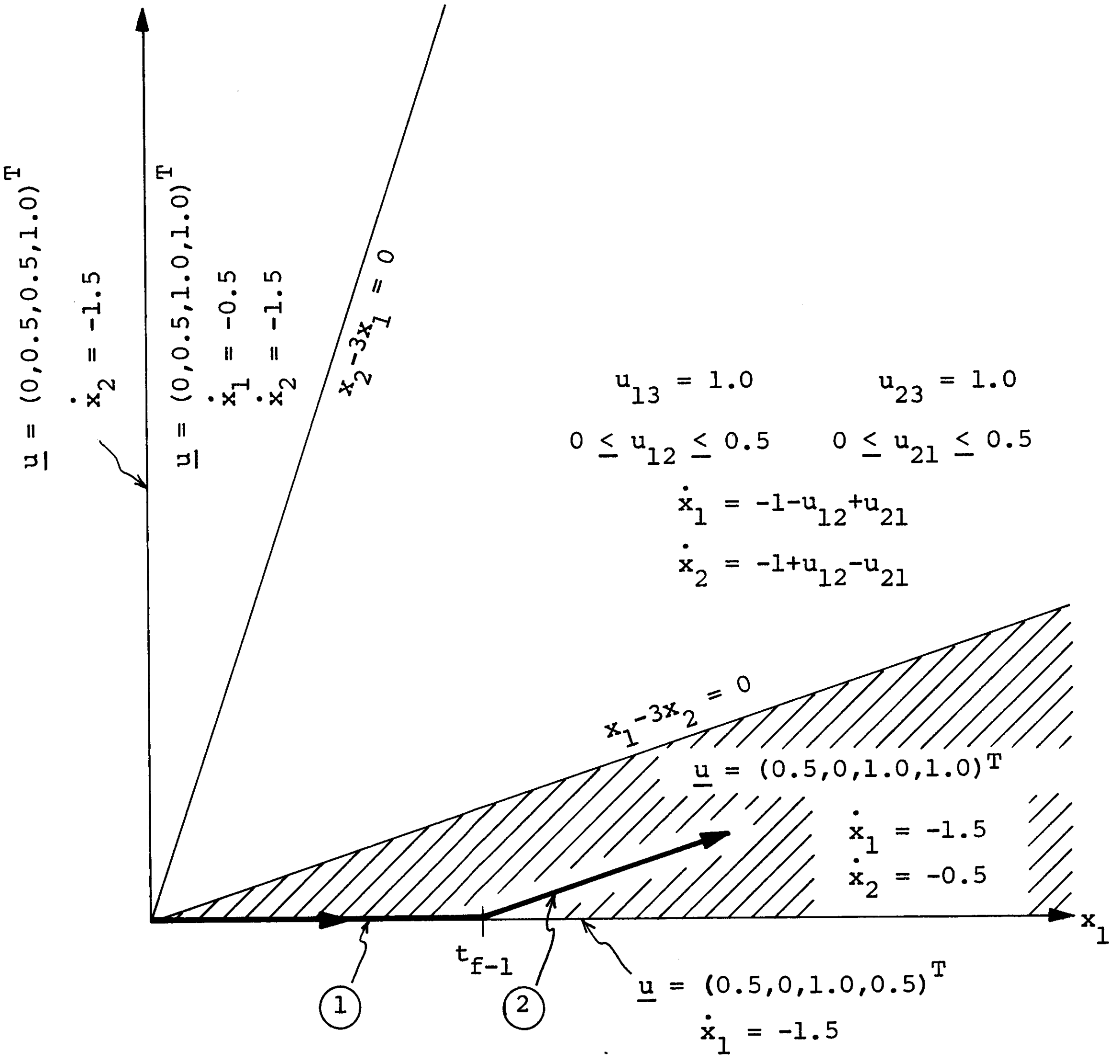


Figure 3.9 Feedback Solution for Example 3.1

time, we may then determine the optimal control anywhere on the x_2 axis. If subsequently we allow x_1 to leave the boundary backward in time, we may then determine the optimal control on the conical region $\{\underline{x} | 0 < x_1 \leq \frac{x_2}{3}\}$. These regions and associated optimal controls are illustrated in Figure 3.9.

(iii) $\dot{x}_1(\tau) < 0$, $\dot{x}_2(\tau) < 0$, $\tau \in [t_{f-1}, t_f]$.

This situation is depicted in Figure 3.10 .

We are considering the situation where both states contact the boundary at the final time. It will be shown shortly that this case is not as special as it may initially appear.

The linear program to be solved pointwise on $\tau \in [t_{f-1}, t_f]$ is (3.89) with $\lambda_1(\tau)$ and $\lambda_2(\tau)$ appropriately determined. Now, since both x_1 and x_2 are on an interior arc over this time interval, Corollary 3.2 gives

$$\lambda_1(t_f) = \lambda_2(t_f) = 0 \tag{3.101}$$

and from equation (3.32)

$$\dot{\lambda}_1(\tau) = \dot{\lambda}_2(\tau) = -1 \qquad \tau \in [t_{f-1}, t_f]. \tag{3.102}$$

Hence, the costates are always equal over this time interval as indicated in Figure 3.10 . As a result of this, the solution to the linear program (3.89) is

$$\begin{aligned} &u_{13}(\tau) = 1.0 \qquad u_{23}(\tau) = 1.0 \\ &0 \leq u_{12}(\tau) \leq 0.5 \qquad 0 \leq u_{21}(\tau) \leq 0.5 \end{aligned} \tag{3.103}$$

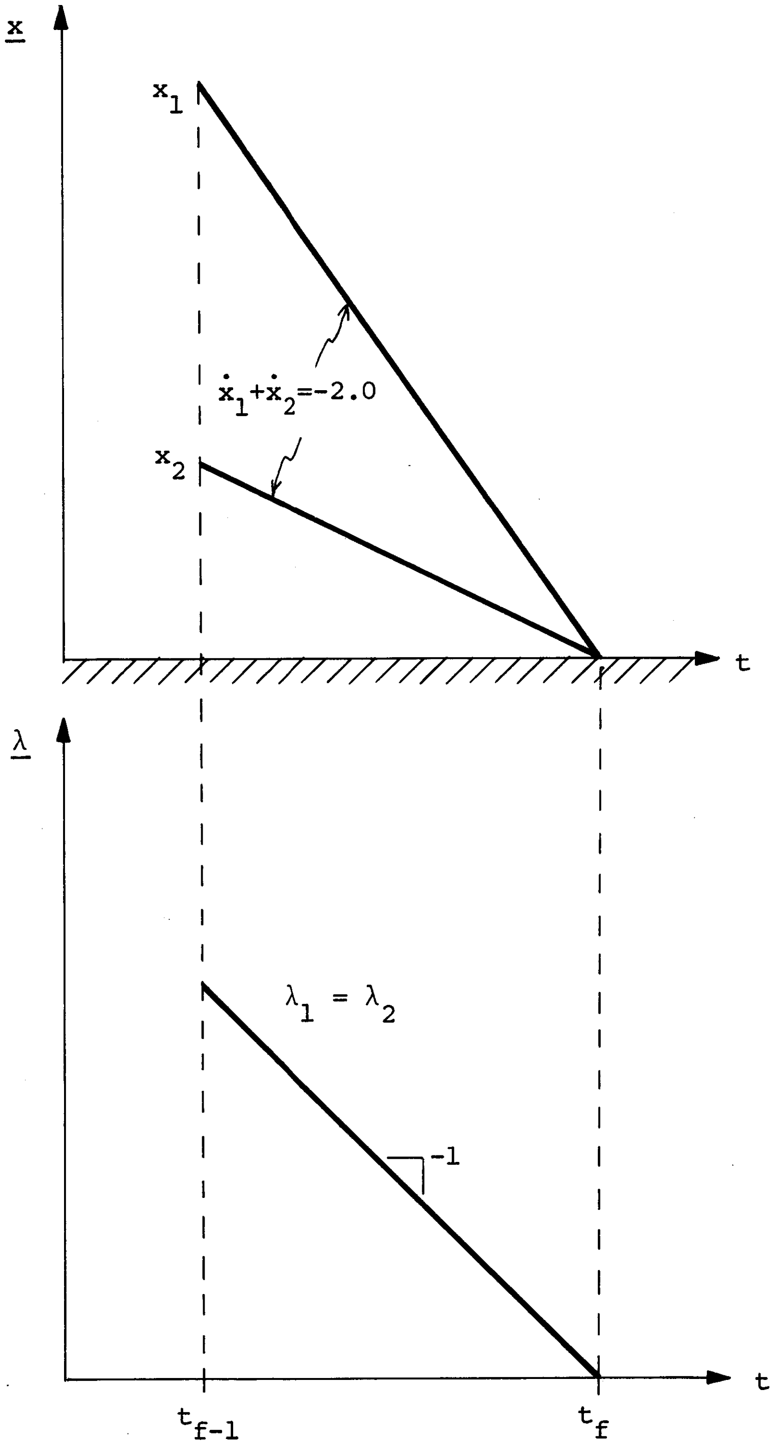


Figure 3.10 State-Costate Trajectory Pair for Example 3.1, Case (iii)

so that

$$\dot{x}_1(\tau) = -1.0 - u_{12}(\tau) + u_{21}(\tau)$$
$$\dot{x}_2(\tau) = -1.0 + u_{12}(\tau) - u_{21}(\tau). \qquad (3.104)$$

Hence, we have encountered the non-uniqueness in the optimal control discussed in a general sense at the end of Section 3.2.1. The values of u_{12} and u_{21} are completely arbitrary as long as they remain within their constraints. Moreover, for any $\tau \in (-\infty, t_f]$ the entire set of optimal controls (3.103) remains optimal. In this case, the optimal directions by which the state leaves the origin lie between $\dot{x}_1/\dot{x}_2 = 3.0$ and $\dot{x}_2/\dot{x}_1 = 3.0$, that is, between the lines $x_1 - 3x_2 = 0$ and $x_2 - 3x_1 = 0$. As before, we now switch our consciousness to forward time and realize that <u>for any point lying between the lines $x_1 - 3x_2 = 0$ and $x_2 - 3x_1 = 0$ (not including these lines) the complete set of controls (3.103) is optimal</u>. For any point on the line $x_1 - 3x_2 = 0$ only the control which has $u_{12} = 0.5$ and $u_{21} = 0.0$ is optimal. Likewise, for any point on $x_2 - 3x_1 = 0$ only the control which has $u_{12} = 0.0$ and $u_{21} = 0.5$ is optimal. See Figure 3.9.

Having filled up the entire space with optimal controls region by region, the specification of the feedback control is now complete.

● Example 3.1

<u>Example 3.2</u>

The network is the same as for Example 3.1, but the cost functional is taken as

$$D = \int_{t_0}^{t_f} \{2x_1(t) + x_2(t)\}dt. \tag{3.105}$$

As in Example 3.1, we take the approach of working backward from the final time, beginning with the three possible situations which may occur at that time.

(i) $\dot{x}_1(\tau) < 0,\ x_2(\tau) = \dot{x}_2(\tau) = 0,\ \tau \in [t_{f-1}, t_f]$

The linear program to be solved over the final time interval $\tau \in [t_{f-1}, t_f]$ is (3.89) with $\lambda_1(\tau)$ and $\lambda_2(\tau)$ appropriately determined. The final condition (3.90) applies, but since the weighting on x_1 is $\alpha_1 = 2$, the appropriate differential equation for λ_1 is

$$\dot{\lambda}_1(\tau) = -2 \qquad \tau \in [t_{f-1}, t_f]. \tag{3.106}$$

Now, $\lambda_1(\tau)$ is determined in the same fashion as $\lambda_2(\tau)$ in case (i) of Example 3.1. That is, the value

$$\lambda_2(\tau) = 0 \qquad \tau \in [t_{f-1}, t_f] \tag{3.107}$$

allows the solution to (3.89) to be such that $\dot{x}_2(\tau) = 0,\ \tau \in [t_{f-1}, t_f]$. Consequently, the optimal control (3.93) applies here. The feedback control on the x_1-axis is therefore assigned in the same fashion as in Example 3.1. See Figure 3.12.

Let us now allow x_2 to leave the boundary backward in time at t_{f-1}. This situation (as well as the previous one) is illustrated in Figure 3.11. The linear program to be solved is as always (3.89). In this case we have

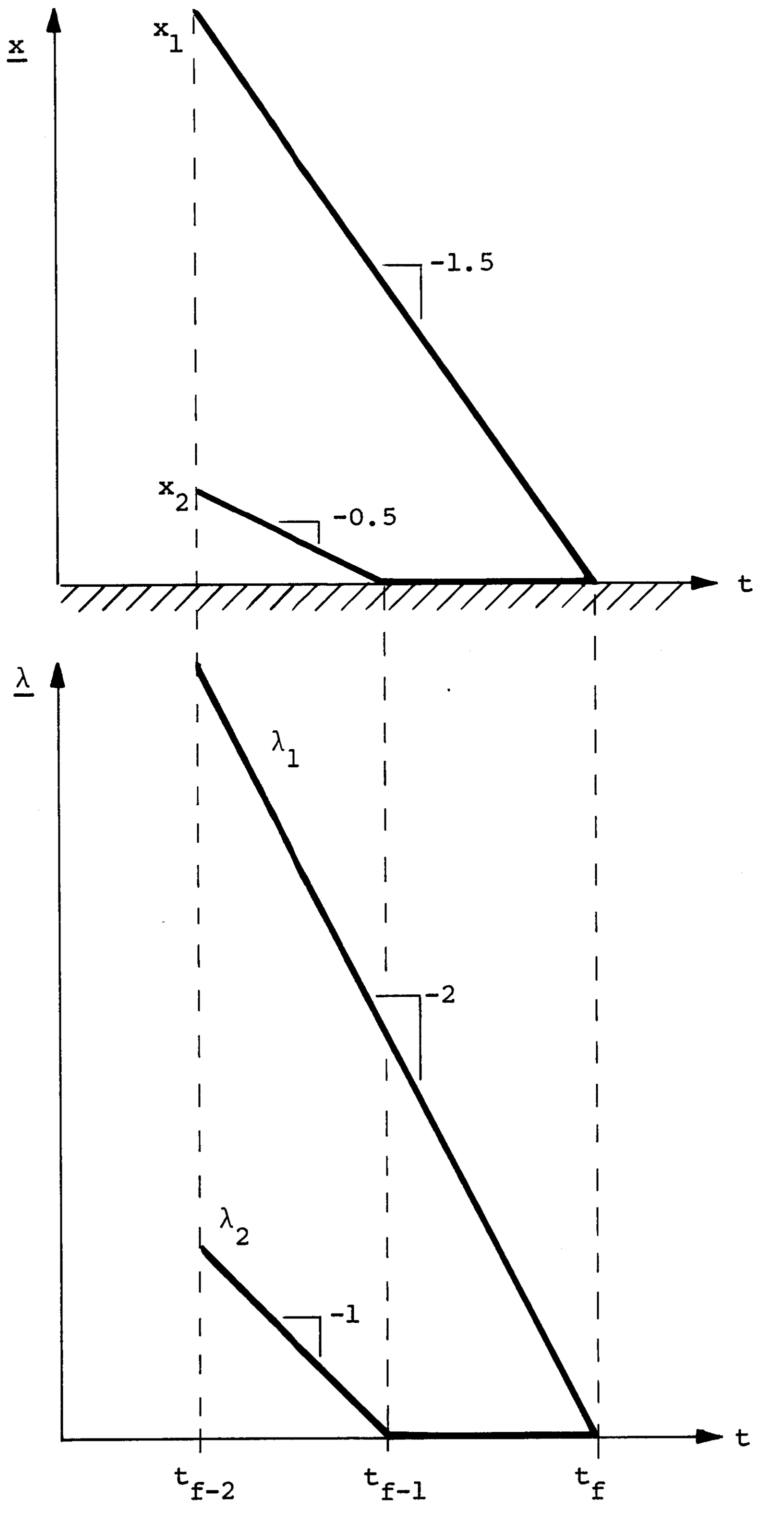
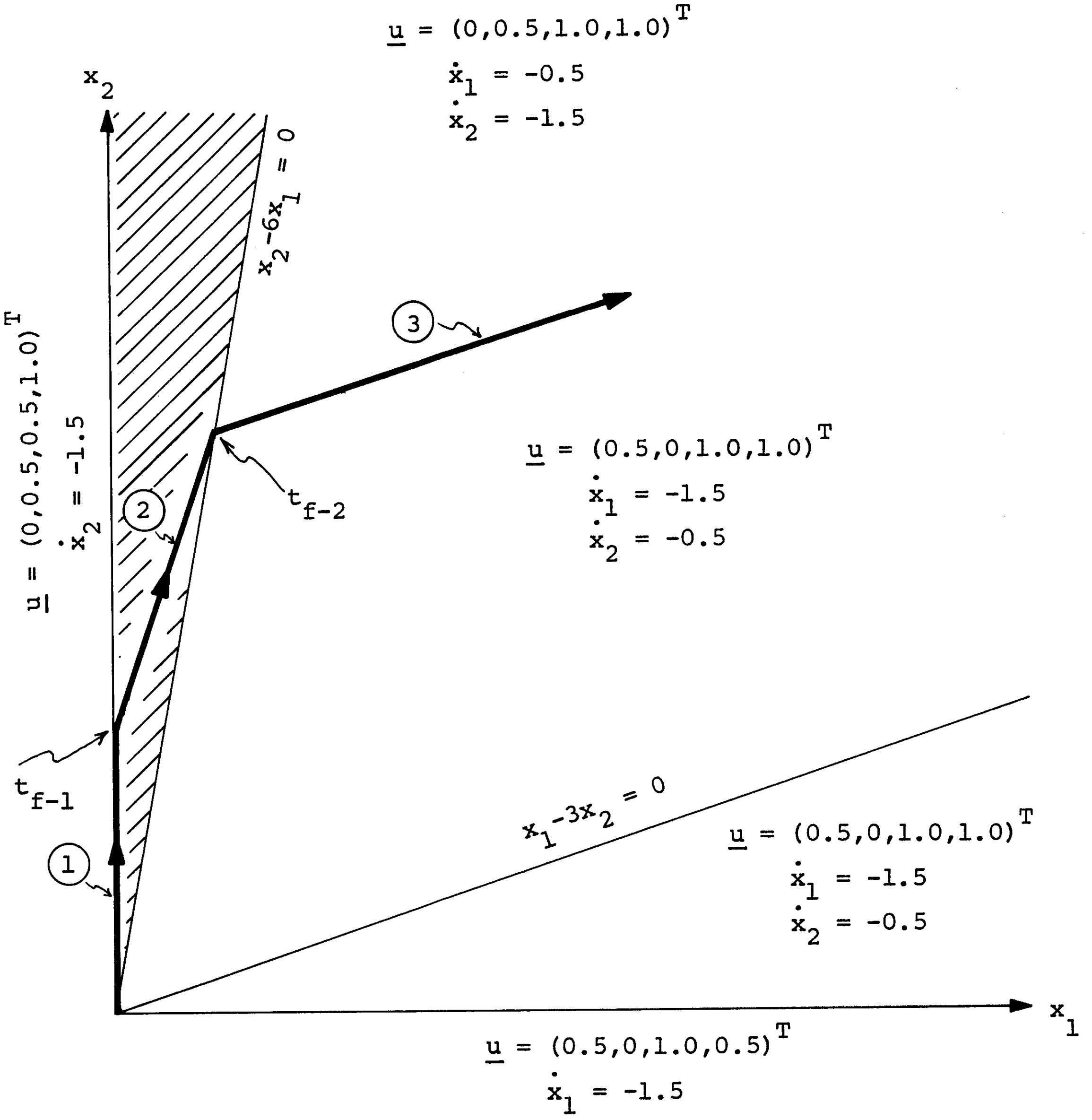


Figure 3.11 State-Costate Trajectory Pair for Example 3.2, Case (i)



Example 3.12 Feedback Solution for Example 3.2

$$\begin{aligned} \lambda_1(t_{f-1}^-) &= \lambda_1(t_{f-1}^-) = 2(t_f - t_{f-1}) \\ \lambda_2(t_{f-1}^-) &= \lambda_2(t_{f-1}^-) = 0. \end{aligned} \tag{3.108}$$

Since both x_1 and x_2 are on interior arcs over this interval, their differential equations are

$$\left.\begin{aligned} \dot{\lambda}_1(\tau) &= -2 \\ \dot{\lambda}_2(\tau) &= -1 \end{aligned}\right\} \tau \in [t_{f-2}, t_{f-1}). \tag{3.109}$$

Also, as before, all that matters in the solution of the linear program is that $\lambda_1(\tau) > \lambda_2(\tau) > 0$, $\tau \in [t_{f-2}, t_{f-1})$. Therefore, the solution is given by (3.98) and this solution applies on the conical region $\{\underline{x} | 0 < x_2 \leq \frac{x_1}{3}\}$ as argued in Example 3.1. See Figure 3.12.

(ii) $\dot{x}_2(\tau) < 0$, $x_1(\tau) = \dot{x}_1(\tau) = 0$, $\tau \in [t_{f-1}, t_f]$.

The details of this situation are depicted in Figure 3.13.

We know from Corollary 3.2 that

$$\lambda_2(t_f) = 0 \tag{3.110}$$

and from (3.32) that

$$\dot{\lambda}_2(\tau) = -1 \qquad \tau \in [t_{f-1}, t_f]. \tag{3.111}$$

We now may find by the process of elimination that the only value of $\lambda_1(\tau)$, $\tau \in [t_{f-1}, t_f]$ for which $\dot{x}_1 = 0$ is optimal is:

$$\lambda_1(\tau) = \dot{\lambda}_1(\tau) = 0 \qquad \tau \in [t_{f-1}, t_f]. \tag{3.112}$$

Therefore, the solution to (3.89) is the same as in Example 3.1,

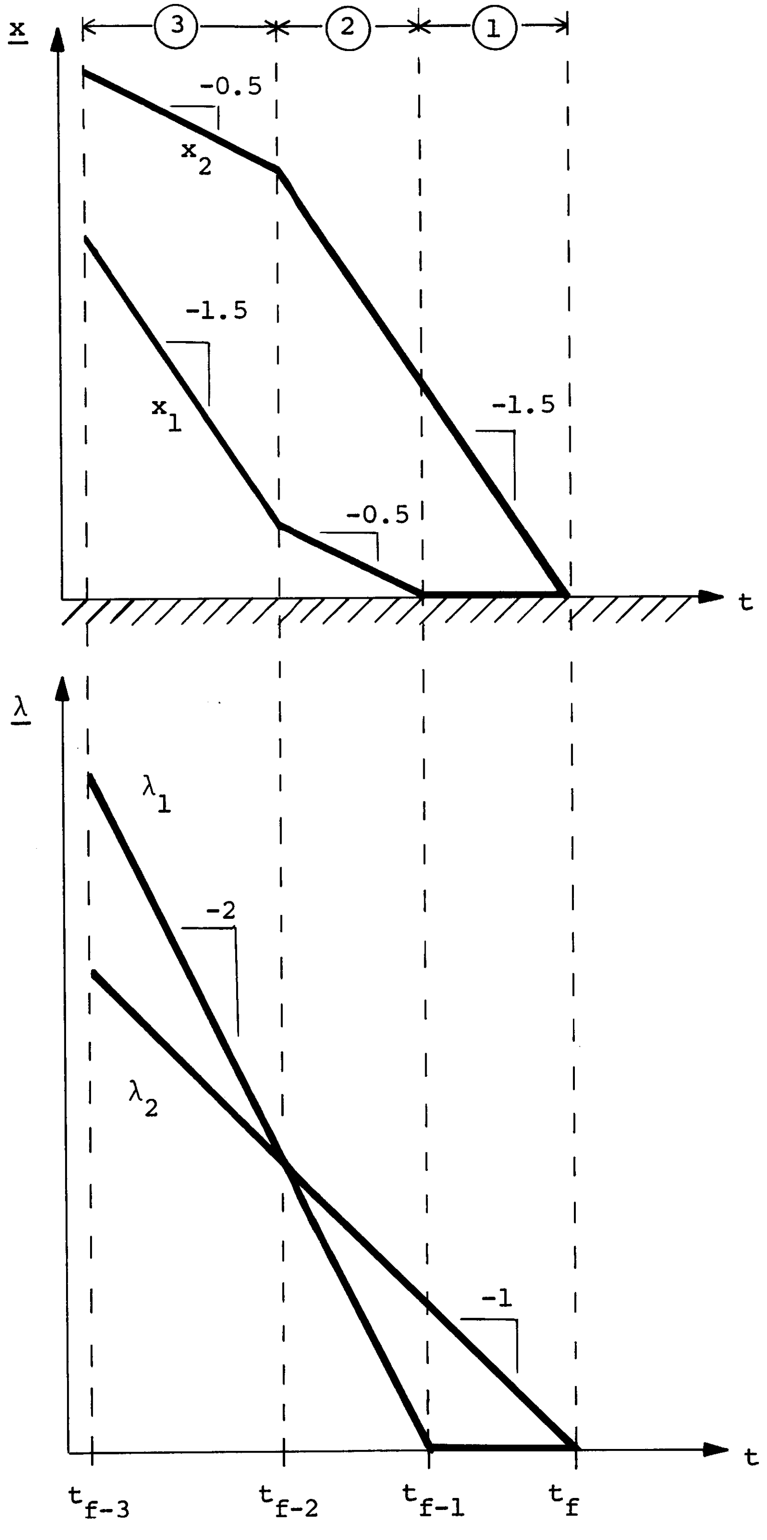


Figure 3.13 State-Costate Trajectory Pair for Example 3.2, Case (ii)

case (ii) and the control on the x_2-axis is assigned in identical fashion. See Figure 3.12.

As the next step, we now stipulate that x_1 leaves the boundary backward in time at t_{f-1}. Since $x_2(t_{f-1}) > 0$, we have from (3.32) that

$$\lambda_2(t_{f-1}^-) = \lambda_2(t_{f-1}^+) = t_f - t_{f-1}. \tag{3.113}$$

Since costate jumps can only be positive in forward time, we must have

$$\lambda_1(t_{f-1}^-) = 0. \tag{3.114}$$

Also, since $x_1(\tau) > 0$, $x_2(\tau) > 0$, $\tau \varepsilon [t_{f-2}, t_{f-1})$,

$$\left.\begin{array}{l} \dot{\lambda}_1(\tau) = -2 \\ \dot{\lambda}_2(\tau) = -1 \end{array}\right\} \quad \tau \varepsilon [t_{f-2}, t_{f-1}). \tag{3.115}$$

See Figure 3.13. We now notice a fundamental difference between this and the previous situations. <u>At some time before</u> t_{f-1} <u>the sign of</u> $(\lambda_1(\tau) - \lambda_2(\tau))$ <u>changes, which implies that the solution to the linear program (3.89) changes at that time</u>. Therefore, t_{f-2} is not allowed to run to $-\infty$, but is actually the time at which the costates cross and the control switches. The optimal controls and state velocities on either side of the switch are:

$\tau \varepsilon [t_{f-2}, t_{f-1})$:

$$\underline{u} = (0.0, 0.5, 1.0, 1.0)^T \tag{3.116}$$

$$\dot{x}_1 = -0.5 \qquad \dot{x}_2 = -1.5 \tag{3.117}$$

$\tau \in [t_{f-3}, t_{f-2})$:

$$\underline{u} = (0.5, 0.0, 1.0, 1.0)^T \tag{3.118}$$

$$\dot{x}_1 = -1.5 \qquad \dot{x}_2 = -0.5. \tag{3.119}$$

The relationship between the states x_1 and x_2 at t_{f-2} may be calculated as follows:

$$\begin{aligned} \lambda_1(t_{f-2}) &= \lambda_1(t_{f-1}) + 2(t_{f-2} - t_{f-1}) \\ \lambda_2(t_{f-2}) &= \lambda_2(t_{f-1}) + (t_{f-2} - t_{f-1}) \end{aligned} \tag{3.120}$$

but

$$\begin{aligned} \lambda_1(t_{f-1}) &= 0 \\ \lambda_2(t_{f-1}) &= (t_{f-1} - t_f). \end{aligned} \tag{3.121}$$

The crossing condition $\lambda_1(t_{f-2}) = \lambda_1(t_{f-1})$ implies from (3.120)-(3.121) that

$$t_{f-2} - t_{f-1} = t_{f-1} - t_f. \tag{3.122}$$

Now

$$\begin{aligned} x_1(t_{f-2}) &= x_1(t_{f-1}) + 0.5(t_{f-2} - t_{f-1}) \\ x_2(t_{f-2}) &= x_2(t_{f-1}) + 1.5(t_{f-2} - t_{f-1}) \end{aligned} \tag{3.123}$$

but

$$\begin{aligned} x_1(t_{f-1}) &= 0.0 \\ x_2(t_{f-1}) &= 1.5(t_{f-1} - t_f). \end{aligned} \tag{3.124}$$

Finally, (3.122)-(3.124) give

$$x_2(t_{f-2}) - 6x_1(t_{f-2}) = 0. \tag{3.125}$$

That is, the switch of control corresponding to the time t_{f-2} always occurs when the state reaches the line (3.125). Therefore, backward in time the state leaves from anywhere on the x_2 axis with control (3.116) and associated rate (3.117). The direction of travel is actually parallel to the line $x_2 - 3x_1 = 0$. Upon reaching the line $x_2 - 6x_1 = 0$, the control switches to (3.118) and the state travels parallel to the line $x_1 - 3x_2 = 0$ forever. This sequence is illustrated for a sampled trajectory whose portions are labeled ①,②,③ in Figure 3.12 and Figure 3.13.

From these observations, the following may be inferred about the feedback solution: The control (3.116) is optimal anywhere within the region bounded by the x_2-axis and the line $x_2 - 6x_1 = 0$, not including the x_2-axis. The control (3.118) is optimal anywhere within the region bounded by the lines $x_2 - 6x_1 = 0$ and $x_1 - 3x_2 = 0$ not including the former line. These regions are indicated in Figure 3.12.

Since the entire space has now been filled up with optimal controls, the feedback specification is now complete. The reader may verify by a simple exercise that case (iii) merely provides the optimal control on the line $x_1 - 3x_2 = 0$, which we already have.

● Example 3.2

We now summarize the content of the preceding examples. By starting at the final time, we have allowed states to leave the boundary backward in time and have computed the corresponding optimal trajectories as time runs from zero to minus infinity. In the instances when the control did not switch over this interval, we were able to construct one conical region of common optimal control (as well as common control and boundary sequences which take any member state to zero). When the control did switch, as in case (ii) of Example 3.2, two such regions were constructed. By considering enough cases, we were able to fill up the entire space with regions of common optimal control, thereby providing the feedback solutions. The primary difficulty in calculating the optimal controls by the necessary conditions was in the determination of the proper value of the costates corresponding to states specified to be on boundary arcs.

3.3.2 Technique for the Backward Construction of the Feedback Space

The examples of the previous section suggest an approach by which the feedback solution to (3.1)-(3.6) may be synthesized. In this section, we take our lead from these examples to suggest a general method for the synthesis of a feedback solution. We then present the basic structure of an algorithm which may realize the method. Several questions are identified which must be answered in order to implement the algorithm.

3.3.2.1 Preliminaries

We begin with a definition which describes a class of subsets of

$\mathcal{R}^n$ which shall comprise the building blocks of the feedback space:

Definition 3.6: A set $\mathscr{R}, \mathscr{R} \subset \mathcal{R}^n$, is said to be a feedback control region with control set Ω, $\Omega \subset \mathcal{U}$, if the following properties hold:

(i) Consider any two points $\underline{x}_1, \underline{x}_2 \; \varepsilon \; \mathrm{Int}(\mathscr{R})$. Suppose $U(\underline{x}_1) = U$ with associated switch time set $T(\underline{x}_1)$. Then $U(\underline{x}_2) = U$ for some switch time set $T(\underline{x}_2)$. Here $U(\underline{x})$ and $T(\underline{x})$ are as defined in (3.46).

(ii) $\mathcal{B}(\underline{x}_1) = \mathcal{B}(\underline{x}_2)$
where $\mathcal{B}(\underline{x})$ is defined in (3.48).

(iii) Any control $\underline{u} \; \varepsilon \; \Omega$ that keeps the state inside the region $\mathscr{R}$ for a non-zero interval of time is an optimal control. Formally, if $\underline{x}(t) \; \varepsilon \; \mathscr{R}$ then any $\underline{u} \; \varepsilon \; \Omega$ is an optimal control if there is an $\varepsilon > 0$ such that under the application of $\underline{u}$, $\underline{x}(t+\varepsilon) \; \varepsilon \; \mathscr{R}$.

The reader may verify that all of the regions constructed in Examples 3.1 and 3.2 are feedback control regions in the sense of Definition 3.6. Our goal now is to specify an algorithm which will construct feedback control regions for the general problem. As a corollary to Theorem 3.4 we obtain quite readily the following geometrical characterization of feedback control regions.

Corollary 3.3 Let the inputs $\underline{a}$ = constant. Then the feedback control regions of Definition 3.6 are convex polyhedral cones in $\mathcal{R}^n$.

Proof: From properties (i) and (ii) of Definition 3.6 we see that any feedback control region $\mathscr{R}$ is the intersection of a set of regions of the type $\mathscr{C}(U, \mathcal{B})$. From Theorem 3.4 we know that all regions of this type are convex polyhedral cones in the state space; therefore, the non-empty intersection of any set must also be a convex polyhedral cone. ■ Corollary 3.3

Definition 3.7: A segment of an optimal trajectory is that portion which occurs on the time interval between two successive switch times τ_p and τ_{p+1}, not including the time instant τ_{p+1}.

The basic concept underlying the construction of the feedback solution is now presented:

Constructive Dynamic Programming Concept

Construct a set of backward optimal trajectories, each starting at the final time t_f, among which all possible sequences of states leaving the boundary backward in time, both singly and in combination, are represented. Each segment of every optimal trajectory is utilized to construct a feedback control region with associated optimal control set. These feedback regions are convex polyhedral cones, and the union of all such cones is the entire space.

3.3.2.2 Description of the Algorithm

The algorithm by which the above method may be realized is now presented. We emphasize that several of the procedures which the algorithm calls for are not included in this presentation, but shall be singled out for future consideration as the discussion proceeds. For clarity, the discussion in this section is divided into three parts, which we now list:

(i) General description of a step of the algorithm.
(ii) Detailed description of a step of the algorithm.
(iii) Rule governing the frequency of execution of the step, and proof that the algorithm realizes the constructive dynamic programming concept.

(i) General description of a step of the algorithm

Each step consists of two basic operations. They are:

- Operation 1

Allow a certain set of state variables to leave the boundary $\underline{x} = 0$ backward in time and determine the subsequent optimal trajectory (trajectories) backward in time assuming that no other state variables leave the boundary as time runs to minus infinity.

- Operation 2

Utilize the relevant information associated with the optimal trajectory (trajectories) of Operation 1 to construct one or more feedback control regions with associated control sets referred to in the constructive dynamic programming concept.

(ii) Detailed description of a step of the algorithm

We first establish some shorthand notation which facilitates the discussion.

Notation 3.3

$$\mathcal{I}_p \overset{\Delta}{=} \{x_i^j \mid x_i^j(\tau) > 0,\ \tau \in [\tau_p, \tau_{p+1})\} \tag{3.126}$$

is the set of state variables which travel on interior arcs during the segment corresponding to $[\tau_p, \tau_{p+1})$.

Notation 3.4

$$\mathcal{L}_p \overset{\Delta}{=} \{x_i^j \mid x_i^j \text{ leaves the boundary at time } t_p\} \tag{3.127}$$

is the set of state variables designated to leave the boundary at t_p.

With $\mathcal{B}_p$ as defined in (3.47) we take note of the following relationships:

$$\mathcal{I}_{p-1} = \mathcal{I}_p \cup \mathcal{L}_p \tag{3.128}$$

$$\mathcal{L}_p \subset \mathcal{B}_p \tag{3.129}$$

$$\mathcal{B}_{p-1} = \mathcal{B}_p / \mathcal{L}_p. \tag{3.130}$$

Notation 3.5

$$\sigma_p \overset{\Delta}{=} \text{cardinality of } \mathcal{I}_p. \tag{3.131}$$

$$\rho_p \overset{\Delta}{=} \text{cardinality of } \mathcal{L}_p. \tag{3.132}$$

Notation 3.6

Ω_p is the set of optimal control vectors:

$$\Omega_p \triangleq \{\underline{u}^* | \underline{u}^* = \text{ARG} \min_{\underline{u} \varepsilon \mathcal{U}} \underline{\lambda}^T(\tau)\underline{B}\ \underline{u}(\tau),\ \tau\ \varepsilon\ [\tau_p, \tau_{p+1})\}. \qquad (3.133)$$

Notation 3.7

$\mathcal{R}_p$ is the feedback region determined in Operation 2 by the parameters of the trajectory segment on $[\tau_p, \tau_{p+1})$ with associated control set Ω_p.

From notations 3.3 and 3.7 we see that $\mathcal{R}_p \subset R^{\sigma_p}$, where the basis vectors of R^{σ_p} are the elements of I_p. This situation is illustrated in Figure 3.14. Due to the pictorial limitation of three dimensions, R^{σ_p} is represented in two dimensions in Figure 3.14. The remainder of R^n is $R^{n-\sigma_p}$, where the basis vectors of $R^{n-\sigma_p}$ are the elements of B_p. The third orthogonal axis represents $R^{n-\sigma_p}$. Basis sets are in parentheses.

In order to describe a single step (that is a sequence of Operations 1 and 2) we pick up the algorithm at a point at which the construction of a feedback control region $\mathcal{R}_p$ has just been completed in the previous step. We further stipulate that on the current optimal trajectory (trajectories) the state variables of I_p are on interior arcs and those of B_p are on the boundary. Refer again to Figure 3.14. In detail, the two operations are:

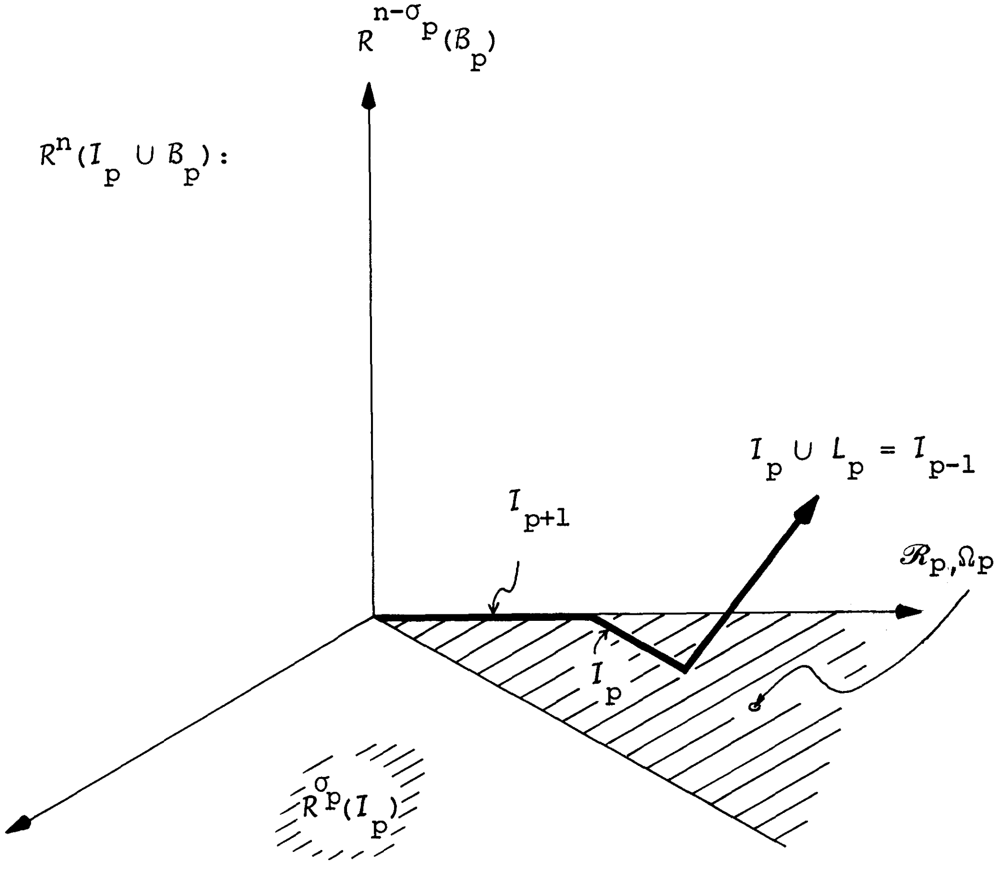


Figure 3.14 Illustration of Feedback Control Region $\mathcal{R}_p$.

- Operation 1

Suppose that we choose a specific set of state variables $L_p \subset B_p$ to leave the boundary backward in time. From Figure 3.14 we visualize that the statement "allow L_p to leave the boundary backward in time" is equivalent to "allow the state variables in L_p to leave the region $\mathscr{R}_p$ backward in time". That is, the state vector leaves the subset $\mathscr{R}_p$ of R^{σ_p} to travel into $R^{\sigma_p+\rho_p}$. Looking ahead, in Definition 3.8 we shall describe a division of $\mathscr{R}_p$ into <u>subregions with respect to</u> L_p. This definition is deferred until after the description of Operation 2 since several notions developed in that operation are required. The question of determining subregions is confronted in Sections 3.3.2.2 and 3.3.3.2.

For the purpose of the present discussion, let us assume that we have divided $\mathscr{R}_p$ into s subregions with respect to L_p. Denote these subregions by $\mathscr{R}_p^1(L_p)$, $\mathscr{R}_p^2(L_p)$, ..., $\mathscr{R}_p^s(L_p)$, where the dependence of the division on the particular set L_p is indicated in the parentheses. Looking ahead once again, the rule governing the frequency of execution of the step (part (iii) of this discussion) shall call for L_p to leave from each of these subregions. To describe a single step, we shall now stipulate that L_p leaves from a specific subregion, say $\mathscr{R}_p^r(L_p)$. This rule shall also call for L_p to leave from one or more points in $\mathscr{R}_p^r(L_p)$. In the current discussion we shall stipulate that L_p is leaving from the specific point $\underline{x}_p \in \mathscr{R}_p^r(L_p)$. Also, we assign the reference time t_p to the time at which L_p leaves $\mathscr{R}_p^r(L_p)$.

We assume that we have arrived at $\underline{x}_p$ at time t_p through a backward trajectory which has been fashioned to satisfy the necessary conditions. Then associated with $\underline{x}_p$ is some possibly infinite set of costate vectors, which we shall designate Λ_p. Fortunately, in principle at least, we shall be required to deal with only a finite subset of any infinite costate set. The question of the determination of this subset, which remains partially unresolved when it contains more than a single member, is discussed in Sections 3.3.2.3 and 3.3.3.2. We assume for the purpose of this discussion that this subset is known and that we are dealing with a specific costate vector $\underline{\lambda}_p \in \Lambda_p$.

We now make the following assumption regarding all optimal trajectories propagating backward in time.

Assumption: No state variable which is off the boundary (that is, travelling on an interior arc) must subsequently return to the boundary backward in time.

The above assumption is certainly not always true, as we demonstrate with a simple example in Section 3.3.4.4. However, in Section 4.3 we show that it is valid for problems involving single destination networks with all unity weightings in the cost functional.

Returning to our current situation, the set of state variables which are travelling on interior arcs subsequent to the departure of L_p from the boundary backward in time is $I_{p-1} = I_p \cup L_p$ as indicated in Equation (3.128). Also, the set of state variables travelling on boundary arcs is $\mathcal{B}_{p-1} = \mathcal{B}_p | \mathcal{L}_p$ by (3.130). According to the assumption, all state variables in $\mathcal{I}_{p-1}$ remain off the boundary for the entire time interval $\tau \in (-\infty, t_p)$. We now state the problem which comprises the central task of Operation 1:

> Given the state $\underline{x}_p$ and costate $\underline{\lambda}_p$ at time t_p, find <u>all</u> optimal trajectories backward in time on $\tau \in (-\infty, t_p)$ for which
>
> $$\dot{x}_i^j(t_p^-) \leq 0 \qquad \forall\, x_i^j \in \mathcal{L}_p \tag{3.134}$$
>
> $$\dot{x}_i^j(\tau) = 0 \qquad \forall\, x_i^j \in \mathcal{B}_{p-1} \tag{3.135}$$
>
> $$x_i^j(\tau) > 0 \qquad \forall\, x_i^j \in \mathcal{I}_{p-1} \tag{3.136}$$
>
> $$\forall\, \tau \in (-\infty, t_p)$$
>
> or determine that no such optimal trajectory exists.

Note that in the above problem statement the stipulation of equation (3.134) guarantees that the state variables in $\mathcal{L}_p$ do in fact leave (or stay on) the boundary backward in time at t_p. Also, equation (3.135) expresses the desire to maintain on boundary arcs all of those state variables which are on the boundary at t_p and are not members of $\mathcal{L}_p$. Finally, equation (3.136) is a formal statement of the assumption stated above. Now, by the necessary and sufficient conditions (Theorems 3.1 and 3.2 respectively) we know that any (and all) optimal trajectories must have a control which satisfies the following:

$$\underline{u}^*(\tau) = \text{ARG} \min_{\underline{u}(\tau)\varepsilon\mathcal{U}} \underline{\lambda}^T(\tau)\underline{\dot{x}}(\tau) = \text{ARG} \min_{\underline{u}(\tau)\varepsilon\mathcal{U}} \underline{\lambda}^T(\tau)\underline{B}\,\underline{u}(\tau) \tag{3.137}$$

where

$$\underline{\lambda}(t_p) = \underline{\lambda}_p \tag{3.138}$$

$$\dot{\lambda}_i^j(\tau) = -\alpha_i^j \qquad \forall\, x_i^j \,\varepsilon\, \mathcal{I}_{p-1} \tag{3.139}$$

$$\left.\begin{aligned} -d\lambda_i^j(\tau) &= \alpha_i^j d\tau + d\eta_i^j(\tau) \qquad (3.140) \\ d\eta_i^j(\tau) &\le 0 \qquad (3.141) \end{aligned}\right\} \forall\, x_i^j \,\varepsilon\, \mathcal{B}_{p-1}$$

$$\forall\, \tau \,\varepsilon\, (-\infty, t_p).$$

Our task is therefore to find all of the solutions to (3.137)-(3.141) which satisfy the constraints (3.134)-(3.136). The greatest difficulty is finding those values of η_i^j for $x_i^j \,\varepsilon\, \mathcal{B}_{p-1}$ for which the solution to (3.137)-(3.141) satisfies the constraint (3.135). The following simple proposition is helpful in resolving this dilemma.

<u>Proposition 3.1</u>: Any solution to (3.137)-(3.141) which satisfies constraints (3.134)-(3.135) is also a solution to

$$\underline{u}^*(\tau) = \text{ARG} \min_{\underline{u}(\tau)\varepsilon\mathcal{U}} \sum_{x_i^j \varepsilon \mathcal{I}_{p-1}} \lambda_i^j(\tau)\dot{x}_i^j(\tau) \tag{3.142}$$

subject to

$$\dot{x}_i^j(t_p^-) \le 0 \qquad \forall\, x_i^j \,\varepsilon\, \mathcal{L}_p \tag{3.143}$$

$$\dot{x}_i^j(\tau) = 0 \qquad \forall\, x_i^j \,\varepsilon\, \mathcal{B}_{p-1} \tag{3.144}$$

where

$$\left.\begin{aligned} \lambda_i^j(t_p) &= \text{appropriate component of } \underline{\lambda}_p && (3.145)\\ \dot{\lambda}_i^j(\tau) &= -\alpha_i^j && (3.146) \end{aligned}\right\} \quad \forall\, x_i^j \in \mathcal{I}_{p-1}$$

$$\forall\, \tau \in (-\infty,\ t_p).$$

Proof: We may rewrite (3.137) as

$$\underline{u}^*(\tau) = \text{ARG} \min_{\underline{u}(\tau)\in\mathcal{U}} \left[\sum_{x_i^j \in \mathcal{I}_{p-1}} \lambda_i^j(\tau)\dot{x}_i^j(\tau) + \sum_{x_i^j \in \mathcal{B}_{p-1}} \lambda_i^j(\tau)\dot{x}_i^j(\tau) \right]. \tag{3.147}$$

But by (3.135) we have

$$\sum_{x_i^j \in \mathcal{B}_{p-1}} \lambda_i^j(\tau)\dot{x}_i^j(\tau) = 0 \qquad \forall\, \tau \in (-\infty,\ t_p) \tag{3.148}$$

and therefore any solution to (3.147) which satisfies (3.135) is also a solution to (3.142). ■ Proposition 3.1

We use the following notation in discussing the optimization problems presented above:

Notation 3.8: The optimization problem of (3.137)-(3.141) is referred to as the global optimization problem. The minimand associated with the global optimization problem,

$$\underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau) = \sum_{x_i^j \in \mathcal{I}_{p-1}} \lambda_i^j(\tau)\dot{x}_i^j(\tau) + \sum_{x_i^j \in \mathcal{B}_{p-1}} \lambda_i^j(\tau)\dot{x}_i^j(\tau), \tag{3.149}$$

is called the <u>global Hamiltonian</u>.

We point out that the Hamiltonian of Notation 3.8 differs from that defined in (3.29) by the additive term $(\underline{\alpha}^T + \underline{\mu}^T(\tau))\underline{x}(\tau)$. However, since $\underline{x}(\tau)$ is not an explicit function of $\underline{u}$, both are identical with respect to the pointwise miminimization.

<u>Notation 3.9:</u> The optimization problem (3.142)-(3.146) of Proposition 3.1 is referred to as the <u>constrained optimization problem</u>. The minimand associated with the constrained optimization problem,

$$\sum_{x_i^j \in \mathcal{I}_{p-1}} \lambda_i^j(\tau)\dot{x}_i^j(\tau), \tag{3.150}$$

is called the <u>restricted Hamiltonian.</u>

We repeat for emphasis that according to the necessary and sufficient conditions any (and all) optimal trajectories must have a control which is a solution to the global optimization problem, that is, which minimizes the global Hamiltonian. We are looking for the class of optimal trajectories which also satisfies the constraint (3.144). Proposition 3.1 tells us that any (and all) optimal trajectories in this class must also be solutions to the constrained optimization problem. However, we must take care to note that solutions to the constrained optimization problem may not be solutions to the global optimization problem. See Section 3.3.4.1 for an example of this behavior. In order for a constrained solution to be a global solution there must exist values of $\lambda_i^j(\tau)$ for all $x_i^j \in \mathcal{B}_{p-1}$ such that the constrained solution satisfies (3.137) when (3.144) is released. These costates must satisfy (3.138)-(3.141).

Now, note that we are able to solve the constrained optimization problem immediately since we know all of the coefficients of (3.142). The values of $\lambda_i^j(\tau)$ for $x_i^j \in \mathcal{B}_{p-1}$ are not required. The above observations suggest the following two part approach to finding all of the solutions to the global optimization problem which satisfy the constraint equation (3.135):

(a) Find all solutions to the constrained optimization problem (3.142)-(3.146).

(b) Produce values of $\lambda_i^j(\tau)$ for all $x_i^j \in \mathcal{B}_{p-1}$ such that the necessary conditions (3.139)-(3.141) are satisfied and for which all solutions to part (a) are also solutions to the global optimization problem (3.137) or show that no such values exist.

The above tasks were actually performed in a simple fashion for the examples of the previous section. Due to the small dimensionality of the problems we were actually able to determine the globally optimizing values of the costates by inspection (such as the determination of λ_2 in Example 3.1, case (i)). Of course, this is not true in general.

In Section 3.3.3.1 we present a method by which the constrained optimization problem of part (a) may be solved parametrically in time using linear programming techniques. A geometrical approach to the solution of part (b) is presented in Section 3.3.3.2. However, in Section 4.3 we are able to show that for problems involving single destination networks with all unity weightings in the cost functional we are always able to produce the set of costates $\lambda_i^j(\tau)$ for all $x_i^j \in \mathcal{B}_{p-1}$ which is required in part (b) for global optimality. In this case part (b) is therefore unnecessary.

Suppose now that we have managed to solve parts (a) and (b) for the case at hand and have arrived at a sequence of switching times and optimal control sets on $(-\infty, t_p)$. If the control encounters q switches over the interval $(-\infty, t_p)$ we then denote the switching time set as

$$T = \{\tau_{p-q}, \ldots, \tau_{p-1}, t_p\}. \tag{3.151}$$

In this notation the control remains unchanged from τ_{p-q} to minus infinity. In accordance with (3.133) the collection of optimal control sets is denoted

$$\Omega = \{\Omega_{-\infty}, \ldots, \Omega_{p-2}, \Omega_{p-1}\}. \tag{3.152}$$

Here the control set $\Omega_{-\infty}$ is optimal from τ_{p-q} to minus infinity.

Note that we have used t_p to denote the time at which L_p leaves the boundary and $\tau_{p-1}, \tau_{p-2}, \ldots, \tau_{p-q}$ to denote subsequent switch times which occur as τ runs to minus infinity. Although all of the above are switch times in the general sense, we give t_p the special label of boundary junction time since it is forced on by insisting that L_p leave the boundary at that time.

- Operation 2

The purpose of this operation is to utilize the sets T and Ω produced as output from Operation 1 to construct feedback regions in $R^{\sigma_p+\rho_p}$, where the bases vectors are the elements of $I_{p-1} = I_p \cup L_p$. Recall that the trajectory is leaving backward in time from a point $\underline{x}_p$ in a subregion $\mathscr{R}_p^r(L_p)$ of $\mathscr{R}_p$.

Suppose that proceeding backward in time the optimal control encounters a switch at time τ_{p-1} and that the control on $[\tau_{p-1}, t_p)$ is Ω_{p-1}. Then by a procedure to be specified in Section 3.3.3.4 a feedback control region is to be constructed in $R^{\sigma_p+\rho_p}$ which has $\mathscr{R}_p^r(L_p)$ as a wall in R^{σ_p}. We call this region $\mathscr{R}_{p-1}$ and associate with it the optimal control set Ω_{p-1}. The construction to be specified will assure us that $\mathscr{R}_{p-1}$ is the largest such region with associated control set Ω_{p-1}.

Notation 3.10: The time τ_{p-1} is called a break time of the optimal trajectory. If $\underline{x}_{p-1}$ is the state of the optimal trajectory at τ_{p-1}, then $\underline{x}_{p-1}$ is called a break point. Finally, the feedback region $\mathscr{R}_{p-1}$ so constructed is called a break feedback control region.

Proceeding backward in time, if the control switches at time τ_{p-2}, then the break feedback control region $\mathscr{R}_{p-2}$ with associated optimal control set Ω_{p-2} is constructed adjacent to $\mathscr{R}_{p-1}$. We continue in this fashion to construct adjacent break regions. If the control switches q times, then q break regions are constructed culminating in the final break region $\mathscr{R}_{p-q}$ with associated optimal control set Ω_{p-q}.

From time τ_{p-q} to minus infinity the optimal control set does not experience a break and is $\Omega_{-\infty}$. In terms of the construction of a feedback control region, this situation is to be distinguished from that of break regions, and the construction technique is presented in Section 3.3.3.3. Applying this technique, we obtain a final feedback control region $\mathscr{R}_{-\infty}$ adjacent to $\mathscr{R}_{p-q}$ for which the associated optimal control set is $\Omega_{-\infty}$.

Notation 3.11: The region $\mathscr{R}_{-\infty}$ is referred to as a non-break feedback control region.

In summary, the sets T and Ω are to be utilized to construct a total of q+1 adjacent feedback control regions in $R^{\sigma_p+\rho_p}$: q break regions and one non-break region. This construction is illustrated in Figure 3.15.

At this point, the reader may be curious to know what the various regions described may look like. We need only refer back to Example 3.2 to find very simple examples of both types of feedback regions. In Figure 3.12, the portion of the optimal trajectory labelled ② - ③ corresponds to allowing x_1 leave from anywhere on the x_2 axis backward in time. Since the control switches at t_{f-2}, it was found that for the

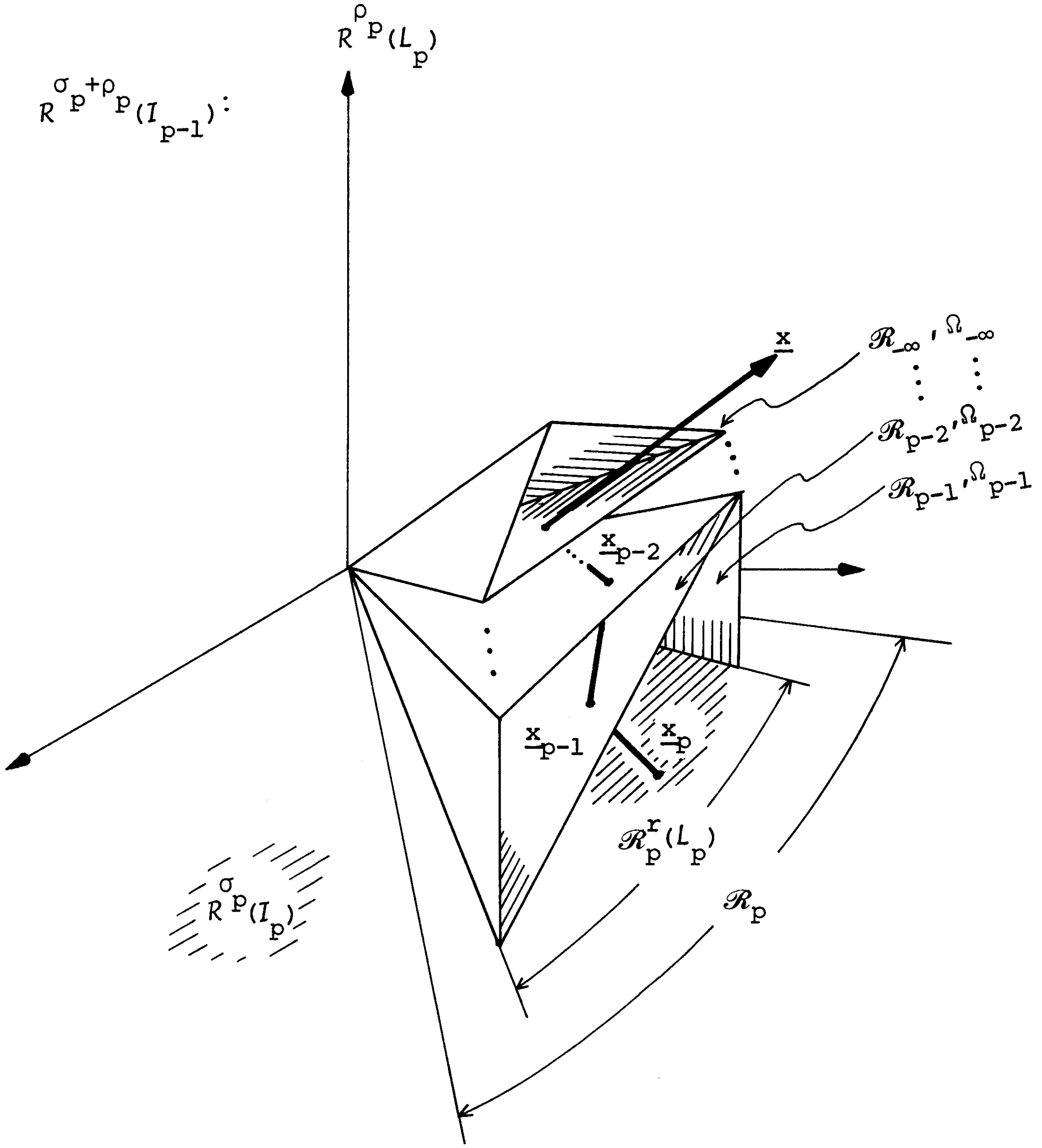


Figure 3.15 Construction of Successive Feedback Control Regions from Subregion $\mathscr{R}_p^r(L_p)$

cone bounded by the x_2-axis and the line $x_2-6x_1 = 0$, not including the x_2-axis, the optimal control is (3.116). Hence, this cone is a break feedback control region. Once the control switches at t_{f-2}, it continues on to time equals minus infinity without a switch. We then conclude that this control is optimal anywhere on the cone bounded by the lines $x_2-6x_1 = 0$ and $x_1-3x_2 = 0$, not including the former line. Therefore, this cone is a non-break feedback control region. For this simple example, we were able to show that for any trajectory of the type mentioned above, the control always switches on the line $x_2-6x_1 = 0$. Looking ahead, we shall generalize this notion by defining a class of trajectories for which the control always switches on a portion of a given hyperplane. These surfaces are the $\sigma_p+\rho_p-1$ dimensional walls which separate the adjacent feedback control regions created in Operation 2, and shall be referred to as breakwalls.

In Sections 3.3.3.3 and 3.3.3.4, we shall show that the set of break walls and the collection of optimal control sets Ω are the parameters which enter into the construction of the feedback control regions for a given backward optimal trajectory. It is clear that these parameters are dependent upon the value of the costates at the point at which L_p leaves $\mathscr{R}_p$, and therefore dependent upon the point. In order to accommodate the constructions we now define a subregion as follows:

Definition 3.8: Suppose the set of state variables in L_p is designated to leave the feedback control region $\mathscr{R}_p$ backward in time. Then a subregion $\mathscr{R}_p(L_p)$ of $\mathscr{R}_p$ is the set of all those points in $\mathscr{R}_p$ which when taken as the point of departure of L_p result in a common Ω and common set of breakwalls.

The problem of determining subregions is confronted in Section 3.3.3.2. However, once again the special case of single destination networks with all unity weightings in the cost functional provides us with a simplifying situation. In section 4.3 we are able to prove that for this class of problems there is exactly one subregion per region with respect to any given L_p, thus eliminating the need to determine subregions.

(iii) Rule governing the frequency of execution of the step and proof that the algorithm realizes the constructive dynamic programming concept

Suppose that some feedback control region $\mathscr{R}_p$ (break or non-break) has been constructed as in Operation 2. Let $\mathscr{R}_p$ be a subset of R^{σ_p}, where the coordinate axes of R^{σ_p} are the elements of the interior state variable set I_p. Furthermore, assume that $\sigma_p < n$, that is, $\mathscr{R}_p$ is a feedback region whose dimension is strictly less than the dimension of the entire state space. Therefore, the set of state variables B_p remaining on the boundary is non-empty. We now consider all subsets of B_p which are combinations of its elements taken 1, 2, ..., $n-\sigma_p$ at a time. The total number of such subsets is $2^{n-\sigma_p}-1$. We then let L_p

range over all of the subsets so determined. For each subset we first divide $\mathscr{R}_p$ into subregions with respect to that set as specified in Definition 3.8. We then perform the basic two operation step of the algorithm with the set $\mathcal{L}_p$ leaving from each of its subregions in $\mathscr{R}_p$. Every step performed results in a set of feedback control regions constructed as part of Operation 2. However, the feedback regions constructed at each step may not be new in that they may have been constructed in some previous step. We may therefore be "writing over" regions. In essence, we are being conservative in insisting that $\mathcal{L}_p$ be set equal successively to all possible subsets of $\mathcal{B}_p$, but no method is currently known for the a priori elimination of those subsets which will produce previously constructed regions.

Let us now view in general terms what the algorithm is doing. The algorithm is initiated at t_f with the first feedback control region $\mathscr{R}_f$ being the origin (which is essentially a degenerate region). The set $\mathcal{B}_f$ is then composed of all the state variables of the problem. We then allow $\mathcal{L}_f$ to range over all possible 2^n-1 combinations of state variables in $\mathcal{B}_f$; that is, we allow all possible combinations of state variables to leave the boundary backward in time at t_f. By Corollary 3.2 we know that the values of the costates at t_f corresponding to those state variables leaving the boundary at t_f are zero. We are then able to solve the constrained optimization problem of this section. (Operation 1, part (a)) on $(-\infty, t_f)$ since all we require are those costates which correspond to state variables off of the boundary. For those sets of state variables leaving the boundary which are found to have globally optimal trajectories (Operation 1, part (b)), feedback control regions are constructed which range all the way from one dimensional (that is, axes of R^n) to n-dimensional (that is, n-dimensional subsets of R^n). We then use each region of this set as the starting point for the sequence of steps which builds new higher dimensional regions. This process is continued until all feedback control regions constructed are n-dimensional.

Let us now pick a point $\underline{x}$ in a feedback region $\mathscr{R}_p$ and follow the optimal trajectory which takes the state from $\underline{x}$ to $\underline{0}$. Since $\mathscr{P}_p$ is a feedback control region, there exists some control in Ω_p which is optimal when applied at $\underline{x}$. By applying any control in Ω_p which keeps the state in $\mathscr{R}_p$, we will eventually intersect the wall which separates $\mathscr{R}_p$ from some adjacent feedback control region $\mathscr{R}_{p+1}$. The trajectory will then travel in $\mathscr{R}_{p+1}$ with appropriate control in Ω_{p+1}, until it strikes the wall separating $\mathscr{R}_{p+1}$ from some feedback control region $\mathscr{R}_{p+2}$. We continue in this fashion until the state works its way through a sequence of adjacent feedback control regions until it finally reaches $\underline{0}$.

<u>Proposition 3.2</u>: Execution of the constructive dynamic programming algorithm will result in the complete specification of the optimal feedback control.

<u>Proof:</u> The rule governing the execution of the step of the algorithm

insures that feedback control regions are constructed for every conceivable type of optimal trajectory in terms of sequences of state variables on and off boundary arcs. Moreover, we are finding the largest such regions since we are solving for all optimal controls corresponding to each sequence. Therefore, the complete execution of the algorithm will result in the entire space being filled up with feedback control regions. ■ Proposition 3.2

3.3.2.3 Discussion of the Algorithm

We have spoken in general terms of constructing feedback control regions and allowing all possible sets of state variables remaining on the boundary to leave from these regions backward in time. However, the reader should not lose sight of the fact that we are employing a specific set of optimal trajectories constructed backward in time to satisfy the necessary conditions. In this section we take a look at a typical backward optimal trajectory and enumerate some of the basic problems involved in the execution of the algorithm. Specifically, we establish the notational setting in which these problems are to be attacked in subsequent sections.

Global Optimality

Consider the situation in which we allow the state variables in L_{p+1} to leave the boundary at some boundary junction time t_{p+1}. This situation is illustrated in Figure 3.16. We defer the question of how

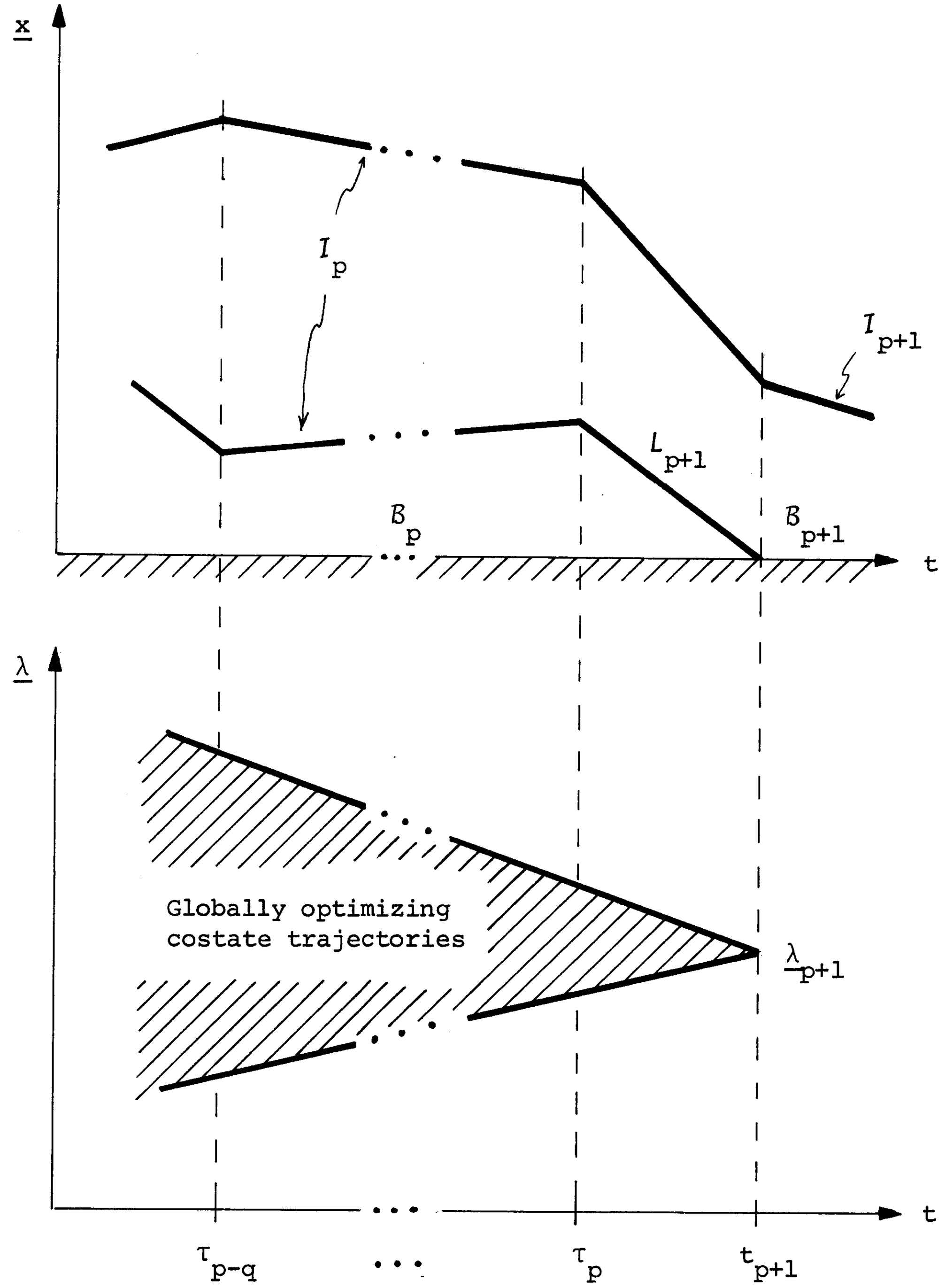


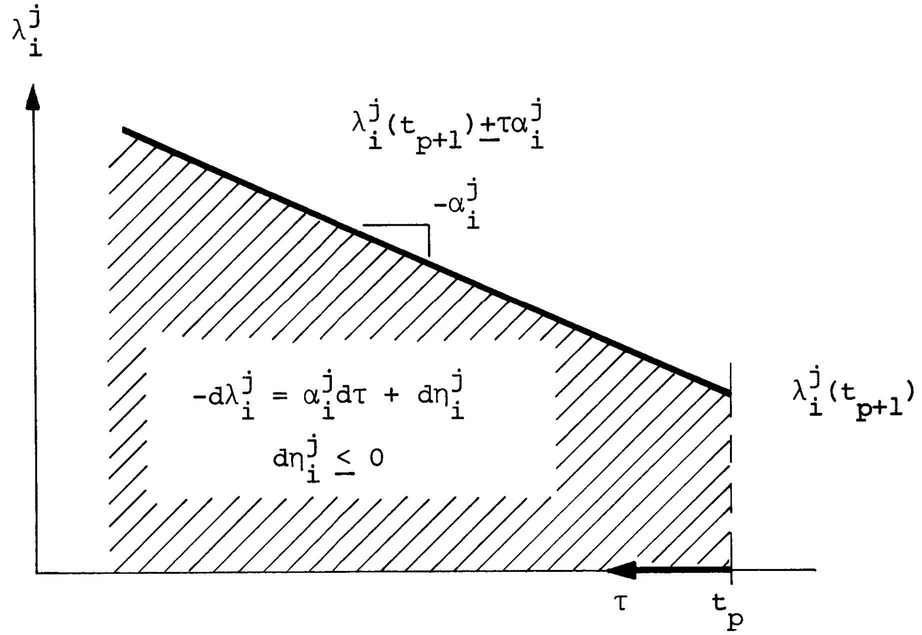
Figure 3.16 Globally Optimizing Costate Trajectories for Given State Trajectory

boundary junction times are determined until the discussion under the heading of "subregions" below. Also, we assume that we are given a particular value of $\underline{\lambda}_{p+1}$ at t_{p+1} from among a potentially non-unique set Λ_{p+1} of appropriate costates at that time. See Figure 3.16. We shall specify what we mean by "appropriate" in the discussion under the heading of "leave-the-boundary costates" below.

We do not know a priori if the situation described above is optimal. We begin by calculating all optimal solutions on $(-\infty, t_{p+1})$ via the constrained optimization technique of Operation 1, part (a). This is illustrated for a particular trajectory in Figure 3.16 . Once this is accomplished, we must satisfy part (b) of Operation 1 by producing a legitimate set of costates on $(-\infty, t_{p+1})$ for which the solution on this interval is globally optimal. To be more specific, since $\lambda_i^j(\tau)$ for $x_i^j \varepsilon \mathcal{I}_p$ are determined exactly by (3.139) for the given $\underline{\lambda}_{p+1}$, the task is actually to produce a set of costates $\lambda_i^j(\tau)$ for $x_i^j \varepsilon \mathcal{B}_p$ which: (i) satisfy the necessary conditions and (ii) together with $\lambda_i^j(\tau)$ for $x_i^j \varepsilon \mathcal{I}_p$ result in a globally optimal solution on $(-\infty, t_{p+1})$.

We shall now specify all of those values of $\lambda_i^j(\tau)$ for $x_i^j \varepsilon \mathcal{B}_p$ which satisfy the necessary conditions. At time t_{p+1} we have assumed that we are given a specific costate value $\underline{\lambda}_{p+1}$ from among a potentially non-unique costate set Λ_{p+1}; that is, the values of $\lambda_i^j(t_{p+1})$ for all $x_i^j \varepsilon \mathcal{B}_p$ are given. If we designate τ to be time running backward from t_{p+1}, then equations (3.140)-(3.141) indicate that the maximum value any $\lambda_i^j(\tau)$ may achieve is when we take $d\eta_i^j(\tau) = 0$. In this case we have

$$\lambda_i^j(\tau) \leq \lambda_i^j(t_{p+1}) + \tau\alpha_i^j \quad \forall\, x_i^j \in \mathcal{B}_p . \tag{3.153}$$



λ_i^j
$\lambda_i^j(t_{p+1}) + \tau\alpha_i^j$
$-\alpha_i^j$
$-d\lambda_i^j = \alpha_i^j d\tau + d\eta_i^j$
$d\eta_i^j \leq 0$
$\lambda_i^j(t_{p+1})$
τ
t_p

Figure 3.17 Possible values of λ_i^j when x_i^j is on boundary arc ($x_i^j = 0$).

Therefore, as a function of time the value of $\lambda_i^j(\tau)$ for $x_i^j \in \mathcal{B}_p$ which satisfy the necessary conditions are those which fall below $\lambda_i^j(t_{p+1}) + \tau\alpha_i^j$. This is depicted in Figure 3.17. We are free to pick any of these values in our attempt to satisfy global optimality on $(-\infty, t_{p+1})$.

Finally, we note that part (b) of Operation 1 requires only that we produce a single set of costates for which the solution is globally optimal. However, there may be a non-unique class of such costates which are legitimate, as indicated in Figure 3.16.

Leave-the-boundary costates

If the solutions obtained in Operation 1 are globally optimal, then we proceed to construct from these solutions a set of feedback control regions in Operation 2. The details of the constructions are presented in Sections 3.3.3.3 and 3.3.3.4. Suppose now that we have completed Operation 2 and that the feedback control regions which have been constructed are $\mathscr{R}_{-\infty}$, $\mathscr{R}_{p-q}$, ..., $\mathscr{R}_{p-1}$, $\mathscr{R}_p$. Here we have assumed that the control switches q times on $(-\infty, t_{p+1})$ and that the switch time set and control set are as in (3.151) and (3.152) respectively. Therefore $\mathscr{R}_p$ is constructed from the solution parameters on $[\tau_p, t_{p+1})$, $\mathscr{R}_{p-1}$ is constructed from the solution parameters on $[\tau_{p-1}, \tau_p)$, ..., $\mathscr{R}_{-\infty}$ is constructed from the solution parameters on $(-\infty, \tau_{p-q})$. Recall that $\mathscr{R}_{-\infty}$ is a non-break feedback control region and all of the rest ($\mathscr{R}_{p-q}$, ..., $\mathscr{R}_{p-1}$, $\mathscr{R}_p$) are break feedback control regions.

Now, the rule governing the frequency of execution of the step of the algorithm calls for us to let all possible combinations of state variables in $\mathcal{B}_p$ to leave backward in time from each of these regions. As an example, let us restrict attention to the particular feedback control region $\mathscr{R}_p$ and a particular set of state variables $\mathcal{L}_p \subset \mathcal{B}_p$. Allowing the state variables in $\mathcal{L}_p$ to leave backward in time from some point in $\mathscr{R}_p$ is equivalent to allowing them to leave at some boundary junction time t_p, where t_p falls on the interval $[\tau_p, t_{p+1})$. In order to allow the state variables in $\mathcal{L}_p$ to leave $\mathscr{R}_p$

at t_p we must perform the following task:

> Determine the values of $\lambda_i^j(t_p)$ for $x_i^j \varepsilon \mathcal{B}_p$ which allow $\mathcal{L}_p$ to leave the boundary backward in time optimally for all potential boundary junction times $t_p \varepsilon [\tau_p, t_{p+1})$ <u>or</u> show that no such values exist which satisfy the necessary conditions.

These values may be nonunique and will be a subset of the globally optimizing values as illustrated in Figure 3.18. Now, the costates so determined constitute Λ_p and will serve as initial conditions for the determination of the backward optimal trajectory which must be solved for the state variables in $\mathcal{L}_p$ leaving from the feedback control region $\mathscr{R}_p$. We refer to the above task as the <u>determination of leave-the-boundary costates</u> at t_p, where the implication is that the costates are such that they allow the states in $\mathcal{L}_p$ to leave the boundary optimally at t_p.

<u>Subregions</u>

The required division of $\mathscr{R}_p$ into subregions with respect to $\mathcal{L}_p$ is equivalent to dividing $[\tau_p, t_{p+1})$ into a set of time subintervals. Over each subinterval, the subsequent optimal control sets and breakwalls on $(-\infty, t_p)$ corresponding to the state variables in $\mathcal{L}_p$ leaving the boundary are the same for any point on the subinterval, by Definition 3.8. The values of $\lambda_i^j(t_p)$ for $x_i^j \varepsilon \mathcal{B}_p$ are drawn from the leave-

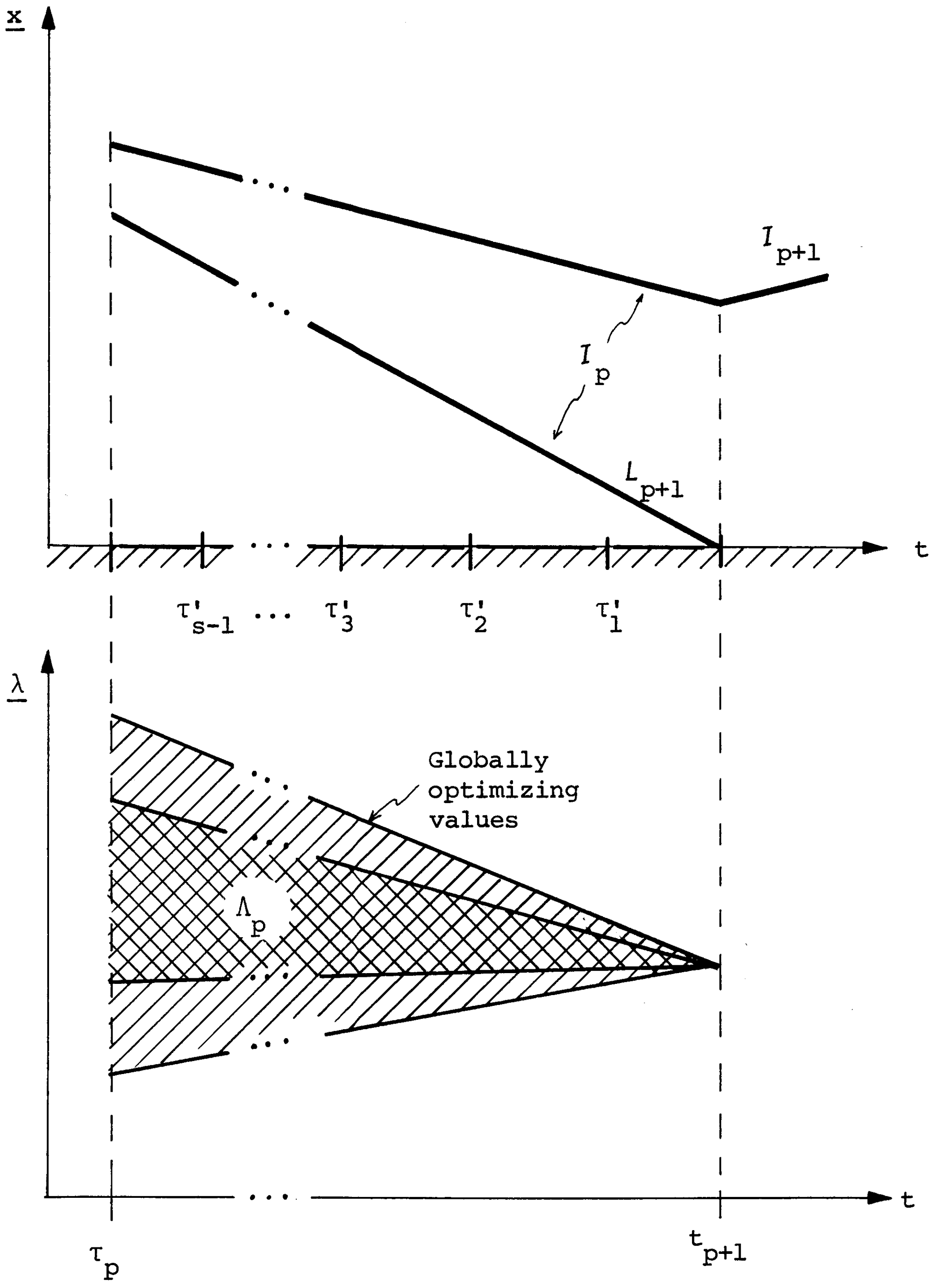


Figure 3.18 State-Costate Trajectory Pair Illustrating Global Optimality, Leave-the-Boundary Costates and Subregions

the-boundary set Λ_p discussed above. Since Λ_p may be infinite, we must somehow reduce to a finite number those values which must be considered in this fashion. This problem is discussed in Section 3.3.4.2. In Section 3.3.4.3 we discuss the issues involved in performing the subdivision for a particular $\underline{\Lambda}_p \in \Lambda_p$. The time subdivision which we seek is illustrated conceptually in Figure 3.18, which is an enlarged picture of the interval $[\tau_p, t_{p+1})$. In this case there are s subintervals corresponding to s subregions with respect to L_p. We use the symbols $\tau'_1, \tau'_2, \ldots, \tau'_{s-1}$ to designate the extremities of the subintervals on $[\tau_p, t_{p+1})$, with the exception of τ_p and t_{p+1}. The subinterval $[\tau'_1, t_{p+1})$ corresponds to the subregion $\mathscr{R}^1_p(L_p)$, the subinterval $[\tau'_2,\tau'_1)$ to the subregion $\mathscr{R}^2_p(L_p)$, ..., until finally the subinterval $[\tau_p, \tau'_{s-1})$ corresponds to the subregion $\mathscr{R}^s_p(L_p)$.

We clearly have many questions to answer in order to be in a position to perform the algorithm. The basic procedures are confronted in Section 3.3.3 and complicating features are discussed in Section 3.3.4.

3.3.3 Basic Procedures of the Algorithm

3.3.3.1 Solution of Constrained Optimization Problem

The first task of Operation 1 is the solution of the constrained optimization problem (3.142)-(3.146) pointwise in time on the interval $\tau \in (-\infty, t_p)$. We begin by expressing the program in terms of the underlying decision variables, which are the controls, and also by integrating (3.146) backward in time:

$$\underline{u}^*(\tau) = \text{ARG} \min_{\underline{u}(\tau)\varepsilon\mathcal{U}} \sum_{x_i^j \varepsilon \mathcal{I}_{p-1}} \lambda_i^j(\tau)\,[\underline{b}_i^j\, \underline{u}(\tau) + a_i^j] \tag{3.154}$$

subject to

$$\underline{b}_i^j\, \underline{u}(t_p^-) \leq -a_i^j \qquad \forall\, x_i^j \;\varepsilon\; \mathcal{L}_p \tag{3.155}$$

$$\underline{b}_i^j\, \underline{u}(\tau) = -a_i^j \qquad \forall\, x_i^j \;\varepsilon\; \mathcal{B}_{p-1} \tag{3.156}$$

$$\lambda_i^j(\tau) = \lambda_i^j(t_p) + \tau\alpha_i^j \qquad \forall\, x_i^j \;\varepsilon\; \mathcal{I}_{p-1} \tag{3.157}$$

$$\tau \;\varepsilon\; (-\infty,\, t_p)$$

where τ is time measured backward from t_p and $\underline{b}_i^j$ represents the row of $\underline{B}$ corresponding to x_i^j. We re-write (3.154)-(3.157) by first recognizing that a_i^j in (3.154) is a constant additive term which does not affect the minimization and also be substituting (3.157) into (3.154) to obtain

$$\underline{u}^*(\tau) = \text{ARG} \min_{\underline{u}(\tau)\varepsilon\mathcal{U}'} (\underline{c}_0 + \tau\underline{c}_1)\underline{B}\; \underline{u}(\tau) \tag{3.158}$$

where

$$\mathcal{U}' = \left\{ \begin{array}{lll} \underline{D}\; \underline{u}(\tau) \leq \underline{C} & & (3.159a) \\ \underline{u}(\tau) \geq \underline{0} & & (3.159b) \\ \underline{b}_i^j\underline{u}(0^-) \leq -a_i^j & \forall\, x_i^j \;\varepsilon\; \mathcal{L}_p & (3.159c) \\ \underline{b}_i^j\underline{u}(\tau) = -a_i^j & \forall\, x_i^j \;\varepsilon\; \mathcal{B}_{p-1} & (3.159d) \end{array} \right.$$

and

$$\underline{c}_0 = \sum_{x_i^j \varepsilon \mathcal{I}_{p-1}} \lambda_i^j(0)\underline{b}_i^j \tag{3.160}$$

$$\underline{c}_1 = \sum_{x_i^j \varepsilon \mathcal{I}_{p-1}} \alpha_i^j \underline{b}_i^j \qquad (3.161)$$

$$\tau \,\varepsilon\, (0, \infty).$$

Here we have simply set $t_p = 0$. Equations (3.158)-(3.161) represent the final form of the program with which we shall be dealing. This program differs from the global minimization in that we have the additional constraints (3.159c) and (3.159d). Hence, the solution set specified by the control law (3.41)-(3.43) is too large for our purpose. Unfortunately, the additional constraints preclude the solution of the program directly as a function of the costates as in (3.41)-(3.43).

However, since for fixed τ (3.158)-(3.161) is a linear program, the Simplex technique may be applied to find a solution. Moreover, the cost function of (3.158) is a linear function of the single independent parameter τ, while the constraints are not a function of τ since $\underline{a}$ is constant. This is precisely the form which can be accommodated by parametric linear programming with respect to the cost coefficients. The solution proceeds as follows:

Set $\tau = \delta$, where δ is some small positive number which serves to perturb all costate values by $\alpha_i^j \delta$. We wish to start our solution at time $t_p - \delta$ since we may have $\lambda_i^j(t_p) = 0$ for some $x_i^j \,\varepsilon\, \mathcal{B}_{p-1}$, so that the solution exactly at t_p may not correspond to x_i^j leaving the boundary. Also, constraint (3.159c) is active at this point to insure that the state boundaries $x_i^j \geq 0$, $x_i^j \,\varepsilon\, \mathcal{L}_p$, are not violated just after t_p (in backward time). The number δ must be such that $0 < \delta < \tau_{p-1}$, where τ_{p-1} is the first break time to be encountered backward in time.

We now use the Simplex technique to solve the program at $\tau = \delta$. There are many linear programming computer packages which may be enlisted for this task which utilize efficient algorithmic forms of the Simplex technique to arrive at a single optimal extremum solution. Given this starting solution which we call $\underline{u}_{p-1}$, most packages are also equipped to employ parametric linear programming to find the value of τ for which the current solution ceases to be optimal as well as a new optimal solution. These are the break time τ_{p-1} and the optimal control $\underline{u}_{p-2}$ respectively. We continue in this fashion to find controls and break times until the solution remains the same for τ arbitrarily large. This final solution is the control $\underline{u}_{-\infty}$.

The linearity of the pointwise minimization associated with the necessary conditions has enabled us to find a sequence of optimal controls on the time interval $(-\infty, t_p)$ by the efficient technique of parametric linear programming. However, in the description of Operation 1 in Section 3.3.2.2 we call for <u>all</u> optimal solutions on every time segment. Since we are dealing with a linear program, the specification of all optimal solutions is equivalent to the specification of all optimal extremum of the solution set. Unfortunately, it turns out that the problem of finding all the optimal extremum solutions to a linear program is an extremely difficult one. In Appendix A we show that given an initial optimal extremum solution this problem is equivalent to finding all the vertices of a convex polyhedral set defined by a system of linear equality and inequality constraints. Discussion of this problem has appeared intermittently in the linear programming literature since the early 1950's, where several algorithms based upon different approaches have been presented. See for example Balinski [1961] and Mattheiss [1973]. However, none of these methods has proven computationally efficient for a reasonably large variety of problems. The fundamental difficulty which appears to foil many algorithms, no matter what their underlying approach, is degeneracy in the original linear program. As our problem is characterized by a high degree of degeneracy, one would expect poor performance from any of these algorithms.

In Appendix A we describe an algorithm developed by Bloom [1976] in conjunction with this research for the solution of the problem of finding all of the extremum solutions of a linear program. It is based upon an algorithm of Chernikova [1965] which finds the extreme rays of a convex cone defined by linear equality and inequality constraints. By a simple transformation, this algorithm can be adapted to find the vertices of a convex polyhedral set, which is the problem to be solved. Limited experience with the application of the algorithm indicates that the amount of computation and storage involved may be excessive. These problems are magnified by the extreme numerical sensitivity of the algorithm.

It appears at this time that the development of an efficient algorithm for the solution of this problem is contingent upon the discovery of methods for resolving degeneracy in linear programs. As degeneracy is a fundamental nuisance in most linear programming procedures, this problem is the subject of much ongoing research.

At any rate, assuming that all optimal extremum may be found for every time segment, the output of this procedure consists of the break time set (3.151) and the collection of optimal control sets (3.152). Note that Ω_p is now considered to be the set of all optimal extremum solutions to the constrained linear program on $\tau \in [t_p, t_{p+1})$. In Sections 3.3.3.3 and 3.3.3.4 the requirement that we find all optimal extremum is justified.

3.3.3.2 Determination of Leave-the-Boundary Costates, Subregions and Global Optimality

In this section we introduce a geometrical notion which serves as a setting for understanding and evaluating various issues of the algorithm which were identified in Sections 3.3.2.2 and 3.3.2.3.

Preliminaries

We begin by considering the basic linear program (3.37) of the necessary conditions to be solved pointwise in time. The elements of the m-dimensional control vector $\underline{u}(\tau)$ comprise the underlying set of decision variables in R^m. However, since the cost functional $\underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau)$ is explicitly linear in $\dot{\underline{x}}$, it is quite natural to view the activity of the linear program in the space R^n with the decision vector $\dot{\underline{x}}$. We begin by denoting

$$\underline{y}(\tau) = -\dot{\underline{x}}(\tau) = -\underline{B}\ \underline{u}(\tau) - \underline{a} \tag{3.162}$$

where the negative sign has been introduced solely for the convenience of discussion.

Definition 3.9: We denote by

$$\mathcal{Y} \stackrel{\Delta}{=} \{\underline{y} \ \varepsilon \ R^n | \underline{y} = -\underline{B}\ \underline{u} - \underline{a} \ \text{ and } \ \underline{u} \ \varepsilon \ \mathcal{U}\} \subset R^n \tag{3.163}$$

the set of negative feasible flows attainable through the admissible controls in the presence of constant inputs $\underline{a}$. We shall refer to $\mathcal{Y}$ as the y-constraint figure.

Note that the only difference between $\mathcal{Y}$ as defined above and $\dot{\mathcal{X}}$ as defined in (3.44) is that the former accounts for the inputs $\underline{a}$; therefore, $\mathcal{Y}$ is obtained from $\dot{\mathcal{X}}$ through simple translation by the vector $\underline{a}$. As such, $\mathcal{Y}$ is also a bounded convex polyhedron in R^n. We may now state the representation of the pointwise linear program (3.37) as it appears in R^n:

$$\underline{y}^*(\tau) = \text{ARG} \max_{\underline{y}(\tau)\varepsilon\mathcal{Y}} [\underline{\lambda}^T(\tau)\underline{y}(\tau)] \tag{3.164}$$

$$\tau \ \varepsilon \ [t_0, t_f].$$

Note that the maximum has been taken in (3.164) as opposed to the minimum in (3.37) due to the introduction of the negative sign in (3.162). The solution $\underline{y}^*(\tau)$ to (3.164) represents the negative of the optimal state rate $\underline{\dot{x}}^*(\tau) = \underline{B}\ \underline{u}^*(\tau) + \underline{a}$ at every point in time. However, we shall not be interested in solving (3.164) directly since in general we do not have an explicit set of linear constraints which comprise $\mathscr{Y}$. Instead, we shall be interested in the underlying properties of the linear program as it is expressed in (3.164).

If we consider time to be fixed and express $z = \underline{\lambda}^T \underline{y}$ as the objective function of (3.164), then the geometrical characterization of linear programs calls for the equation $z = \underline{\lambda}^T \underline{y}$ to be visualized as an n-1 dimensional hyperplane in $\mathcal{R}^n$. For a fixed value of $\underline{\lambda}$, varying the value of z causes the hyperplane $z = \underline{\lambda}^T \underline{y}$ to translate parallel to the hyperplane $\underline{\lambda}^T \underline{y} = 0$. The optimal solution (solutions) is achieved for $z^* = \underline{\lambda}^T \underline{y}$ tangent to the constant set $\mathscr{U}$. The solution set consists of all the points of tangency, and may range from a single vertex of $\mathscr{Y}$ to an n-1 dimensional face of $\mathscr{Y}$.

The geometrical interpretation is depicted in Figure 3.19.

The linear program (3.164) is the $\underline{y}$-space version of the global optimization (3.37). However, in part (a) of Operation 1 of the algorithm we are dealing with the constrained optimization problem. We now proceed to express this problem as it appears in $\underline{y}$-space. First the notation:

<u>Notation 3.12</u>

$$z(\tau) = \sum_{x_i^j \in \mathcal{I}_p} \lambda_i^j(\tau) y_i^j \tag{3.165}$$

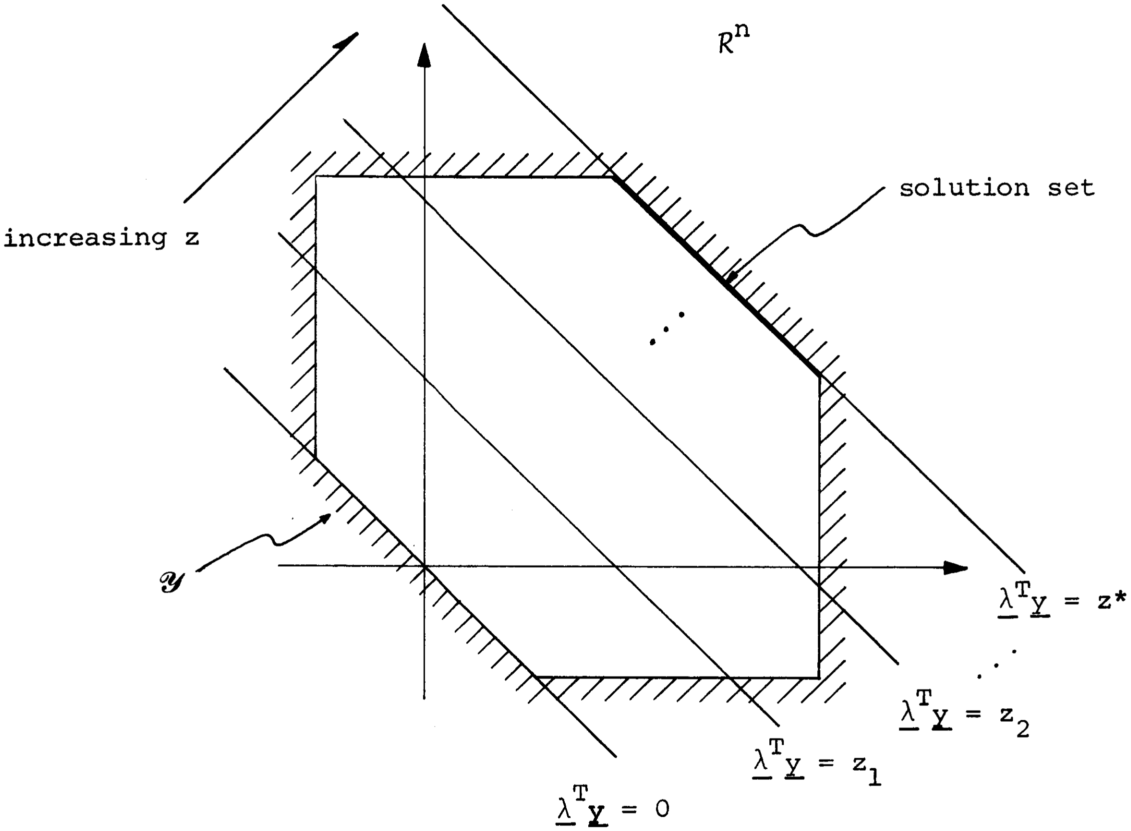

$\mathcal{R}^n$
solution set
increasing z
$\mathcal{Y}$
$\underline{\lambda}^T\underline{y} = z^*$
$\underline{\lambda}^T\underline{y} = z_2$
$\underline{\lambda}^T\underline{y} = z_1$
$\underline{\lambda}^T\underline{y} = 0$

Figure 3.19 Geometrical Interpretation of Pointwise Linear Program in $\underline{y}$-space

is the <u>restricted Hamiltonian</u> at time τ. Geometrically, it is the <u>σ_p-1 dimensional restricted Hamiltonian hyperplane</u> at time τ. We denote this hyperplane as $H_p(\tau)$.

<u>Notation 3.13</u>

$$\mathcal{Y}_p \triangleq \{\underline{y} \in \mathcal{R}^{\sigma_p} | \underline{y} \in \mathcal{Y} \text{ and } y_i^j = 0 \quad \forall\, x_i^j \in \mathcal{B}_p\} \subset \mathcal{R}^{\sigma_p} \qquad (3.166)$$

is the <u>σ_p dimensional restricted $\underline{y}$-constraint figure.</u>

The constrained optimization problem in $\underline{y}$-space is

$$\underline{y}^*(\tau) = \underset{\underline{y}(\tau)\,\varepsilon\,\mathcal{Y}_p}{\text{ARG}\ \ \text{MAX}}\ z(\tau) \qquad (3.167)$$

$$\tau\ \varepsilon\ (-\infty,\ t_p).$$

The solution to (3.167) is piecewise constant on time segments identical to those for the underlying $\underline{u}$-space problem. We therefore use similar notation for specifying the solutions.

Notation 3.14

$$\left.\begin{aligned}
\underline{y}_p &\overset{\Delta}{=} \text{a solution to (3.167) on } [\tau_p,\ \tau_{p+1}),\ \text{referred to} \\
&\quad\ \text{as a } \sigma_p\text{-dimensional operating point} \\
Y_p &\overset{\Delta}{=} \text{the set of all solutions to (3.167) on } [\tau_p,\ \tau_{p+1}). \\
Y &\overset{\Delta}{=} \{\underline{Y}_{-\infty},\ \ldots,\ Y_{f-2},\ Y_{f-1}\}\ \text{is the sequence of the} \\
&\quad\ \text{solution sets to (3.167) on } (-\infty,\ t_p).
\end{aligned}\right\},\qquad (3.168)$$

We now demonstrate the above concepts with the aid of a familiar example.

Example 3.3

Take the network and cost functional of Example 3.1. Then the global constraint figure is

$$\mathcal{Y} = \{y_1,y_2 | y_1 = u_{13}+u_{12}-u_{21},\ y_2 = u_{23}-u_{12}+u_{21}, \qquad (3.169)$$

$$0\le u_{12}\le 0.5,\ 0\le u_{21}\le 0.5,\ 0\le u_{13}\le 1.0,\ 0\le u_{23}\le 1.0\}.$$

This is illustrated in Figure 3.20.

We now follow on Figure 3.20 the steps of the algorithm associated with situation (i) of Example 3.2. First, we let $L_f = \{x_1\}$, that is, x_1 leaves the boundary at t_f. The sought after solution must have $\dot{x}_1$ negative and $\dot{x}_2 = 0$. This corresponds to the constrained maximization

$$\underline{y}^*(\tau) = \text{ARG} \max_{\underline{y}(\tau) \in \mathscr{Y}_{f-1}} \lambda_1(\tau) y_1(\tau) \qquad \tau \in [t_{f-1}, t_f] \tag{3.170}$$

where

$$\mathscr{Y}_{f-1} = \{y_1, y_2 | y_2 = 0,\ y_1 = u_{13} + u_{12} - u_{21} \quad 0 \leq u_{12} \leq 0.5,\ 0 \leq u_{21} \leq 0.5,\ 0 \leq u_{13} \leq 1.0\}. \tag{3.171}$$

The one dimensional restricted constraint figure $\mathscr{Y}_{f-1}$ is illustrated by the heavy line in Figure 3.20, along with the restricted Hamiltonian $H_{f-1}(\tau)$: $z(\tau) = \lambda_1(\tau) y_1$. Clearly, the maximum of H_{f-1} over $\mathscr{Y}_{f-1}$ is achieved at the one dimensional operating point $\underline{y}_{f-1} = (1.5, 0)^T$. This point is globally maximizing if and only if $\lambda_2(\tau) = 0$. The solution point $\underline{y}_{f-1}$ corresponds to the control (3.93) and allows us to specify the feedback control on the x_1-axis as in Figure 3.9.

Next, we set $L_{f-1} = \{x_2\}$, that is, allow x_2 to leave the boundary backward in time at t_{f-1}. Since all states are now in the interior, the maximization is global:

$$\underline{y}^*(\tau) = \text{ARG} \max_{\underline{y}(\tau) \in \mathscr{Y}_{f-2} = \mathscr{Y}} [\lambda_1(\tau) y_1(\tau) + \lambda_2(\tau) y_2(\tau)] \tag{3.172}$$

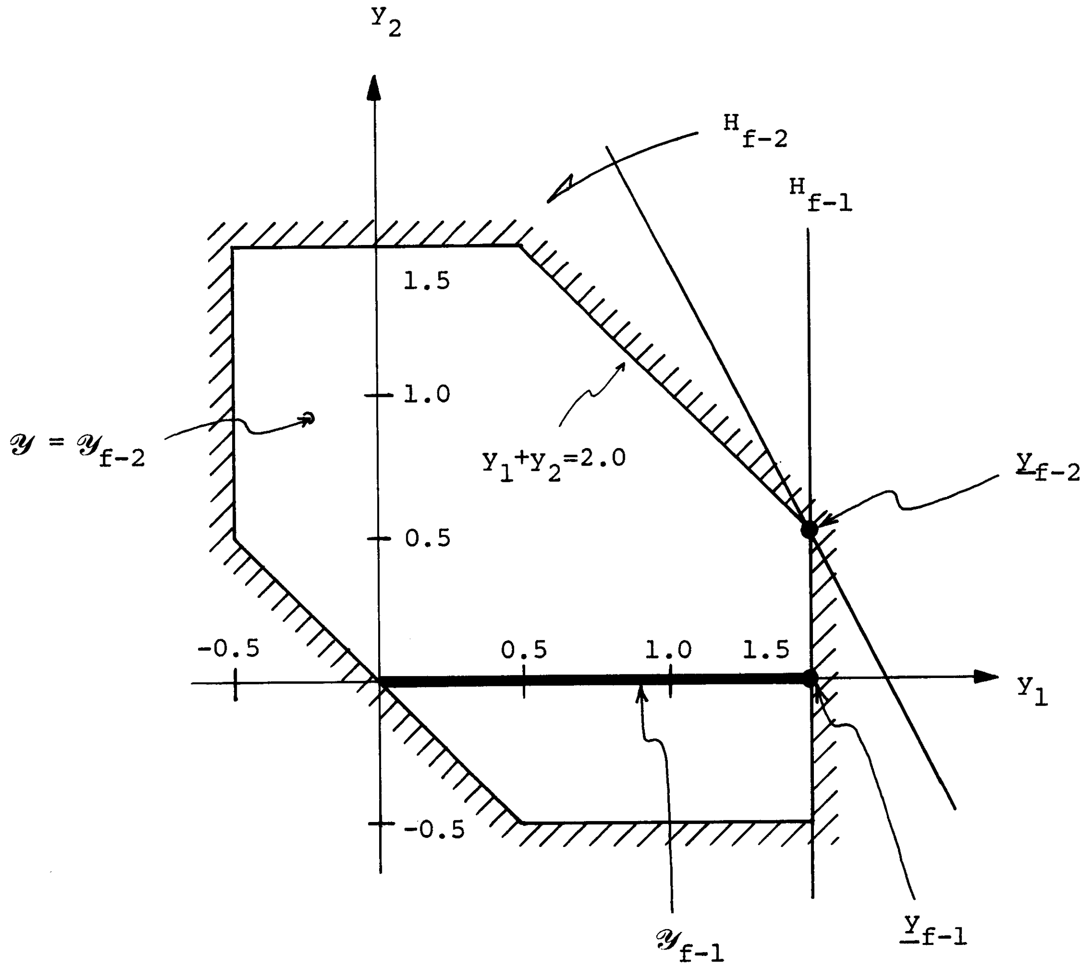


Figure 3.20 Geometrical Interpretation Applied to Problem of Example 3.3

$$\tau \in [t_{f-2}, t_{f-1}).$$

The two dimensional global constraint figure $\mathcal{Y}_{f-2} = \mathcal{Y}$ is illustrated in Figure 3.20, along with the global Hamiltonian $H_{f-2}(\tau):z(\tau) = \lambda_1(\tau)y_1 + \lambda_2(\tau)y_2$. Since the costate behavior as specified in (3.96) and (3.97) implies $\lambda_1(\tau) > \lambda_2(\tau)$, the two dimensional operating point is $\underline{y}_{f-2} = (1.5, 0.5)^T$. This point corresponds to the control (3.98). This is easily seen from Figure 3.20. As time proceeds backward, the relative values of the costates cause the Hamiltonian H_{f-2} to rotate in the direction indicated by the arrow. As time approaches minus infinity, H_{f-2} approaches the face of $\mathcal{Y}$ which lies on the line $y_1+y_2 = 2.0$. Therefore, there is no break point in the control as time approaches minus infinity. Finally, $\underline{y}_{f-2}$ is used to construct the shaded non-break feedback region of Figure 3.9. ■ Example 3.3

Geometrical Interpretation

The observations of the preceding examples suggest the following geometrical interpretation of the backward constructive algorithm presented in Section 3.3: Suppose the set of state variables in $\mathcal{L}_{p+1}$ leaves the boundary backward in time at t_{p+1} resulting in the set of state variables $\mathcal{I}_p$ on interior arcs. Then the constrained optimization problem of Operation 1, part (a), takes place in $\mathcal{R}^{\sigma_p}$, where the basis vectors of $\mathcal{R}^{\sigma_p}$ are the elements of $\mathcal{I}_p$. The constraint set is

$$\mathcal{Y}_p = \mathcal{Y} \cap \mathcal{R}^{\sigma_p} \tag{3.173}$$

where $\mathcal{Y}$ is the global constraint set in $\mathcal{R}^n$ defined by (3.163). The objective function is represented by the σ_p-1 dimensional restricted Hamiltonian hyperplane in $\mathcal{R}^{\sigma_p}$:

$$H_p(\tau) : z(\tau) = \sum_{x_i^j \varepsilon \mathcal{I}_p} \lambda_i^j(\tau) y_i^j \qquad (3.174)$$

$$\tau \ \varepsilon \ (-\infty, t_{p+1}).$$

The initial solution set Y_p of the linear program consists of all points of tangency between $\mathcal{Y}_p$ and $H_p(\tau)$ on $\tau \ \varepsilon \ [\tau_p, \tau_{p+1})$. This solution set consists of one or more extreme points of $\mathcal{Y}_p$ and all the points which are convex combinations of the extreme points. Each such point is a σ_p-dimensional operating point. Note, however, that an extreme point of $\mathcal{Y}_p$ is not necessarily an extreme point of $\mathcal{Y}$ (e.g., $\underline{Y}_{f-1}$ in Figure 3.20 is an extreme point of $\mathcal{Y}_{f-1}$ but only a boundary point of $\mathcal{Y}$). Now, as time runs backward toward minus infinity, the costates evolve according to (3.157) in general causing the orientation of the hyperplane $H_p(\tau)$ to change with respect to $\mathcal{Y}_p$. If $H_p(\tau)$ moves a sufficient amount, then the surface of tangency with $\mathcal{Y}_p$ may change. This occurrence defines the break time τ_p and the solution set will change to the new points of tangency. This will continue until some solution set persists to time equals minus infinity. This version of the constrained optimization problem, which represents part (a) of Operation 1 of the algorithm, is illustrated in Figure 3.21. For the purpose of this discussion, let us now assume that the solutions

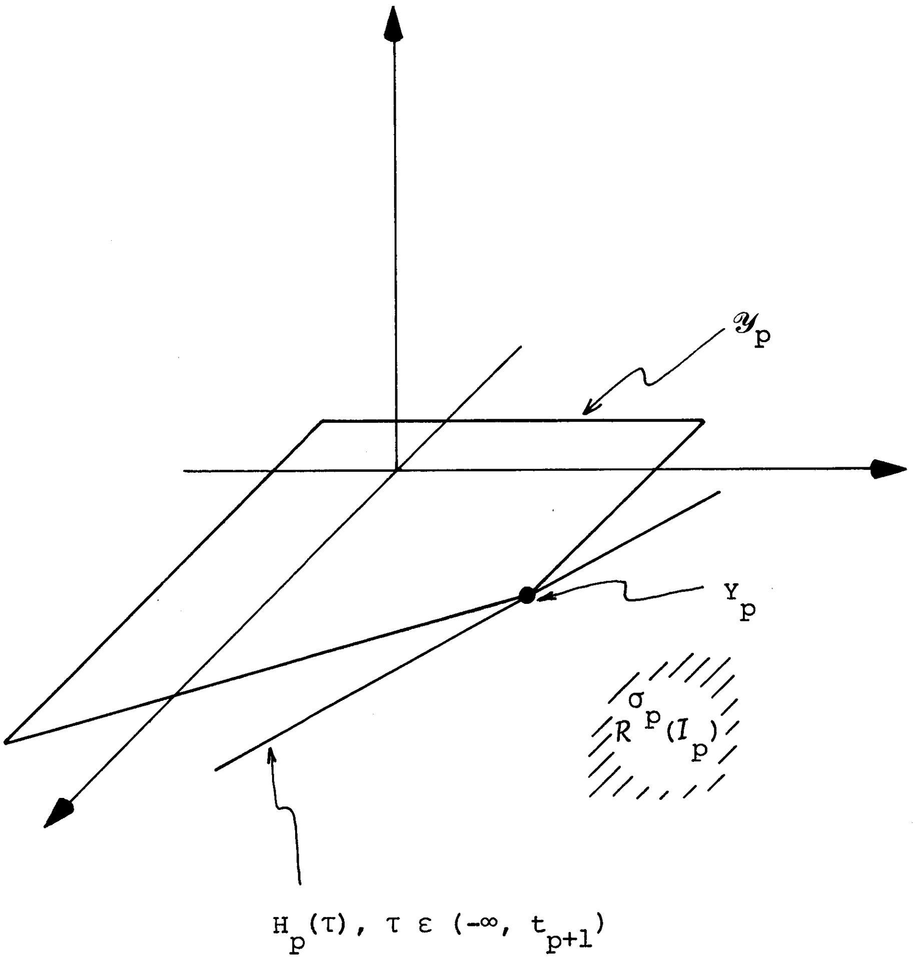


Figure 3.21 Geometrical Interpretation in $\underline{y}$-space of Operation 1, part (a).

obtained satisfy part (b) of Operation 1, that is, they are globally optimal.

Suppose that we now let the set of state variables $\mathcal{L}_p$ leave the boundary some time on the interval $[\tau_p, t_{p+1})$ at boundary junction time t_p. Then the constrained optimization problem takes place in $\mathcal{R}^{\sigma_p+\rho_p}$, where the basis vectors are the coordinates of $\mathcal{I}_{p-1} = \mathcal{I}_p \cup \mathcal{L}_p$. The constraint set is now

$$\mathcal{Y}_{p-1} = \mathcal{Y} \cap \mathcal{R}^{\sigma_p+\rho_p}. \tag{3.175}$$

The objective function is represented by the $\sigma_p+\rho_p-1$ dimensional restricted Hamiltonian hyperplane in $\mathcal{R}^{\sigma_p+\rho_p}$:

$$H_{p-1}(\tau):z(\tau) = \sum_{x_i^j \varepsilon \mathcal{I}_{p-1}} \lambda_i^j(\tau) y_i^j \tag{3.176}$$

$$\tau \;\varepsilon\; (-\infty, t_p).$$

Once again, solutions to the linear program consist of all points of tangency of $H_{p-1}(\tau)$ with $\mathcal{Y}_{p-1}$. See Figure 3.22. These solutions consist of one or more extreme points of $\mathcal{Y}_{p-1}$ and all the points which are convex combinations of these extreme points. These points are $\sigma_p+\rho_p$ dimension operating points of $\mathcal{Y}_{p-1}$. We repeat for emphasis that they are not necessarily extreme points of $\mathcal{Y}$. As we let time run to minus infinity, $H_{p-1}(\tau)$ rotates and picks up new solution sets until finally one persists to minus infinity. For the moment we continue to assume that all constrained solutions obtained are also globally optimal.

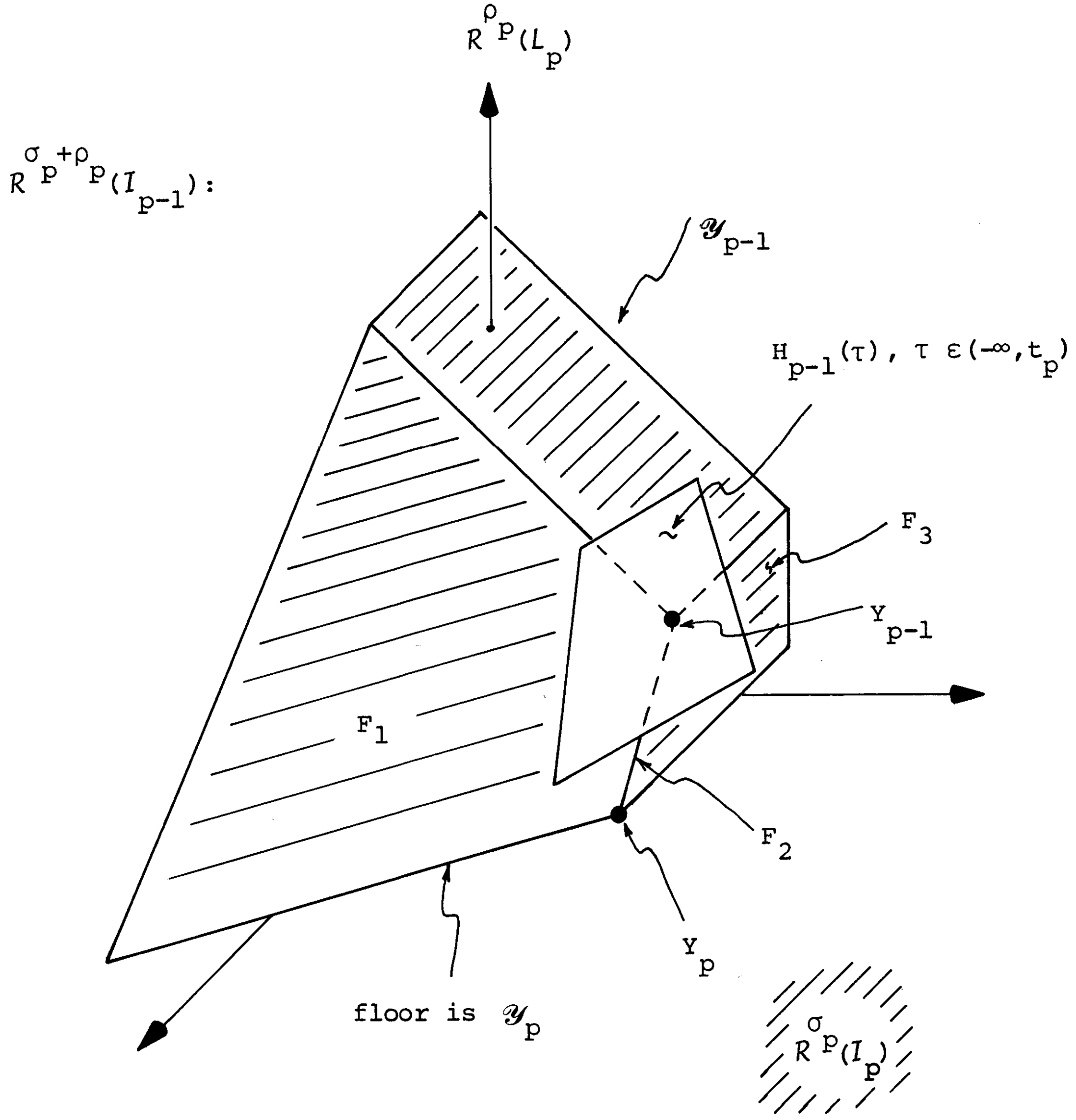


Figure 3.22 Geometrical Interpretation of Minimization in $\underline{y}$-space when state variables in L_p leave the boundary at t_p.

We have now described in geometrical terms the nature of succeeding constrained optimization steps of the algorithm. In summary, as we allow state variables to leave the boundary backward in time, we continually enlarge the space of free decision variables by releasing constraints of the form $y_i^j = 0$. As the constraint figure grows in dimension, so also does the associated Hamiltonian hyperplane. When all the state variables have finally left the boundary backward in time, the constraint figure is the n-dimensional figure $\mathcal{Y}$ and the Hamiltonian is represented by an n-1 dimensional hyperplane.

The geometrical interpretation introduced above shall now be applied to discuss in detail the basic issues presented in Section 3.3.2.3 (global optimality, leave-the-boundary costates, subregions). We begin by establishing some definitions and a theorem which characterize an important geometrical relationship between successive constrained optimization problems.

Geometrical Relationship between Successive Constrained Optimization Problems

Suppose we have a set of state variables $\mathcal{I}_p$ on interior arcs and that the set of σ_p-dimensional operating points is Y_p. At some boundary junction time $t_p \in [\tau_p, \tau_{p+1})$ we wish to allow the set of state variables in the set $\mathcal{L}_p$ to leave the boundary backward in time, where $\mathcal{L}_p \subset \mathcal{B}_p$. Consider now the relationship between the σ_p dimensional constrained optimization problem at t_p^+ and the $\sigma_p+\rho_p$ dimensional constrained optimization problem at t_p^-. We begin by noting that equations (3.173) and (3.175) together imply

$$\mathcal{Y}_p = \mathcal{Y}_{p-1} \cap R^{\sigma_p} \tag{3.177}$$

Therefore, all boundary points of $\mathcal{Y}_p$ are also boundary points of $\mathcal{Y}_{p-1}$. In particular, extreme points of $\mathcal{Y}_p$ are boundary points of $\mathcal{Y}_{p-1}$.

Definition 3.9: Consider the set of operating points $Y_p \subset \mathcal{Y}_p$. Let R^{ρ_p} be the ρ_p dimensional space whose basis vectors are the elements of L_p. Then we call any face of $\mathcal{Y}_{p-1}$ which contains the set Y_p and contains at least one point in the positive orthant of R^{ρ_p} an L_p-positive face.

Most central to this definition is that by (3.162) any point in the positive orthant of R^{ρ_p} has $\dot{x}_i^j$ strictly negative for all $x_i^j \in L_p$ (recall that this is in the presence of inputs $\underline{a}$). Since we intend for the state variables $x_i^j \in L_p$ to leave the boundary backward in time, then we require $\dot{x}_i^j$ strictly negative (in forward time). To illustrate the notion of Definition 3.9, we refer to Figure 3.22 as an example. In this case, the L_p-positive faces of $\mathcal{Y}_{p-1}$ with respect to Y_p are those labelled F_1, F_2 and F_3 in the figure.

The major result of this section is contained in the following theorem. Here we provide the link between the σ_p-1 dimensional Hamiltonian hyperplane $H_p(t_p)$ and the $\sigma_p+\rho_p-1$ dimensional hyperplane $H_{p-1}(t_p)$ which has the state variables in L_p leaving the boundary backward in time at t_p.

Theorem 3.5

Suppose at time t_p the current set of operating points is $Y_p \subset \mathscr{Y}_p$ for the restricted Hamiltonian $H_p(t_p)$. Let L_p be the set of state variables to leave the boundary at t_p. Then any (possibly restricted) Hamiltonian $H_{p-1}(t_p)$ which allows L_p to leave the boundary at t_p is such that:

(a) $H_p(t_p) = H_{p-1}(t_p) \cap R^{\rho_p}$

(b) $H_{p-1}(t_p)$ contains a L_p-positive face of $\mathscr{Y}_{p-1}$

where R^{ρ_p} is as defined in Definition 3.9.

Proof:

(a) The restricted Hamiltonian hyperplane H_p at the time just prior to the departure of L_p is

$$H_p(t_p^+):z(t_p^+) = \sum_{x_i^j \varepsilon I_p} \lambda_i^j(t_p^+)y_i^j. \tag{3.178}$$

The (possibly restricted) Hamiltonian hyperplane H_{p-1} at the time just following the departure of L_p has the form:

$$H_{p-1}(t_p^-):z(t_p^-) = \sum_{x_i^j \varepsilon I_p} \lambda_i^j(t_p^-)y_i^j + \sum_{x_i^j \varepsilon L_p} \lambda_i^j(t_p^-)y_i^j. \tag{3.179}$$

Now, $H_p(t_p^+)$ contains all points in Y_p. Also, since we are assuming Y_{p-1} is such that the state variables in L_p leave the boundary, there must be at least one point $\underline{y}_{p-1} \varepsilon Y_{p-1}$ for which $y_i^j > 0$ for all $x_i^j \varepsilon L_p$. That is, the slope in forward time associated with $x_i^j \varepsilon L_p$ must be strictly negative in order that x_i^j leave the boundary backward in time. If we now evaluate $H_p(t_p^+)$ at any point $\underline{y}_p \varepsilon Y_p$ and $H_{p-1}(t_p^-)$ at the point $\underline{y}_{p-1} \varepsilon Y_{p-1}$, necessary condition (3.14), which stipulates continuity of the Hamiltonian everywhere, gives

$$z(t_p^+) = z(t_p^-) = z(t_p). \qquad (3.180)$$

Furthermore, we have that

$$\lambda_i^j(t_p^+) = \lambda_i^j(t_p^-) = \lambda_i^j(t_p) \qquad \forall\, x_i^j \varepsilon I_p \qquad (3.181)$$

by equation (3.32). By virtue of (3.180) and (3.181) we may write the Hamiltonians as

$$H_p(t_p):z(t_p) = \sum_{x_i^j \varepsilon I_p} \lambda_i^j(t_p) y_i^j \qquad (3.182)$$

$$H_{p-1}(t_p):z(t_p) = \sum_{x_i^j \varepsilon I_p} \lambda_i^j(t_p) y_i^j + \sum_{x_i^j \varepsilon L_p} \lambda_i^j(t_p^-) y_i^j. \qquad (3.183)$$

We conclude immediately from (3.182) and (3.183) that $H_p(t_p) = H_{p-1}(y_p) \cap R^{\rho_p}$.

(b) From part (a) we see that $H_{p-1}(t_p)$ contains $H_p(t_p)$ and therefore contains Y_p. Also, $H_{p-1}(t_p)$ contains at least one point $\underline{y}_{p-1}$ in the positive orthant of R^{ρ_p}. Therefore, $H_{p-1}(t_p)$ must contain the L_p-positive face which contains both Y_p and $\underline{y}_{p-1}$. ■ Theorem 3.5

We are now prepared to discuss the issues of the algorithm presented in Section 3.3.2.3.

Global Optimality

The determination of global optimality is the basic task of part (b) of Operation 1 of the algorithm. For this discussion let us assume that $\underline{y}_p$ is a σ_p-dimensional operating point which maximizes the restricted Hamiltonian H_p over the restricted constraint figure $\mathcal{Y}_p$. We have fixed time here. The relevant question is:

> Do there exist legimate values of λ_i^j for $x_i^j \in \mathcal{B}_p$ such that H is maximized over $\mathcal{Y}$ at $\underline{y}_p$, where H is the n-1 dimensional global Hamiltonian?

By the basic geometrical property of linear programs, $\underline{y}_p$ maximizes H over $\mathcal{Y}$ if and only if H is tangent to $\mathcal{Y}$ at $\underline{y}_p$. Therefore, the values of λ_i^j for $x_i^j \in \mathcal{B}_p$ which we seek are those which rotate H about H_p until H becomes tangent to $\mathcal{Y}$ at $\underline{y}_p$. This is illustrated for a simple situation in Figure 3.23.

In Section 3.3.4.1 we present a simple example of a constrained optimum that is not globally optimal.

Leave-the-boundary costates

The problem here is to

> Determine the values of $\lambda_i^j(t_p^-)$ for all $x_i^j \in \mathcal{L}_p$ such

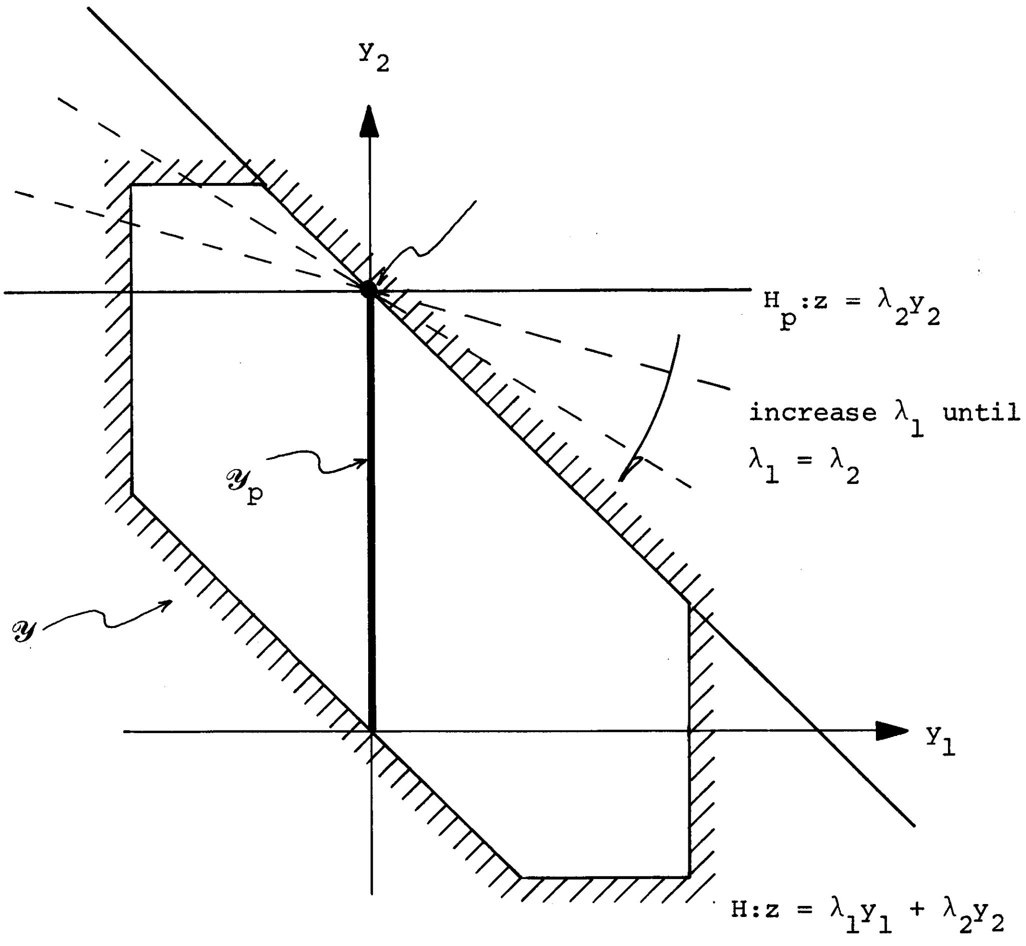


Figure 3.23 Test for Global Optimality in $\underline{y}$-space (Operation 1, part (b))

that when maximizing $H_{p-1}(t_p)$ the solution results in the set of state variables in L_p leaving the boundary backward in time.

Now, statement (a) of Theorem 3.5 says that we essentially want to rotate $H_{p-1}(t_p)$ around $H_p(t_p)$ by varying $\lambda_i^j(t_p^-)$ for $x_i^j \in L_p$ until $H_{p-1}(t_p)$ touches $\mathscr{Y}_{p-1}$ on an L_p-positive face. Note that this rotation is performed while holding time at t_p, and is to be distinguished from the rotation of the Hamiltonian which results from the costates evolving backward in time. See Figure 3.24.

If there are no legitimate values of the leave-the-boundary costates which allow $H_{p-1}(t_p)$ to reach an L_p-positive face, then clearly it is non-optimal for the state variables in L_p to leave the boundary at t_p. Furthermore, if there is more than one L_p-positive face to which $H_{p-1}(t_p)$ may be rotated, then the set of all values of the leave-the-boundary costates will be infinitely non-unique. See Section 3.3.4.2 for an example of this behavior.

Subregions

The problem of determining subregions is a very difficult one. In order to divide the feedback control region $\mathscr{R}_p$ into subregions with respect to L_p, it is necessary to answer the following question:

> What are the collection of optimal control sets $\Omega = \{\Omega_{-\infty}, \ldots, \Omega_{p-2}, \Omega_{p-1}\}$ and set of breakwalls asso-

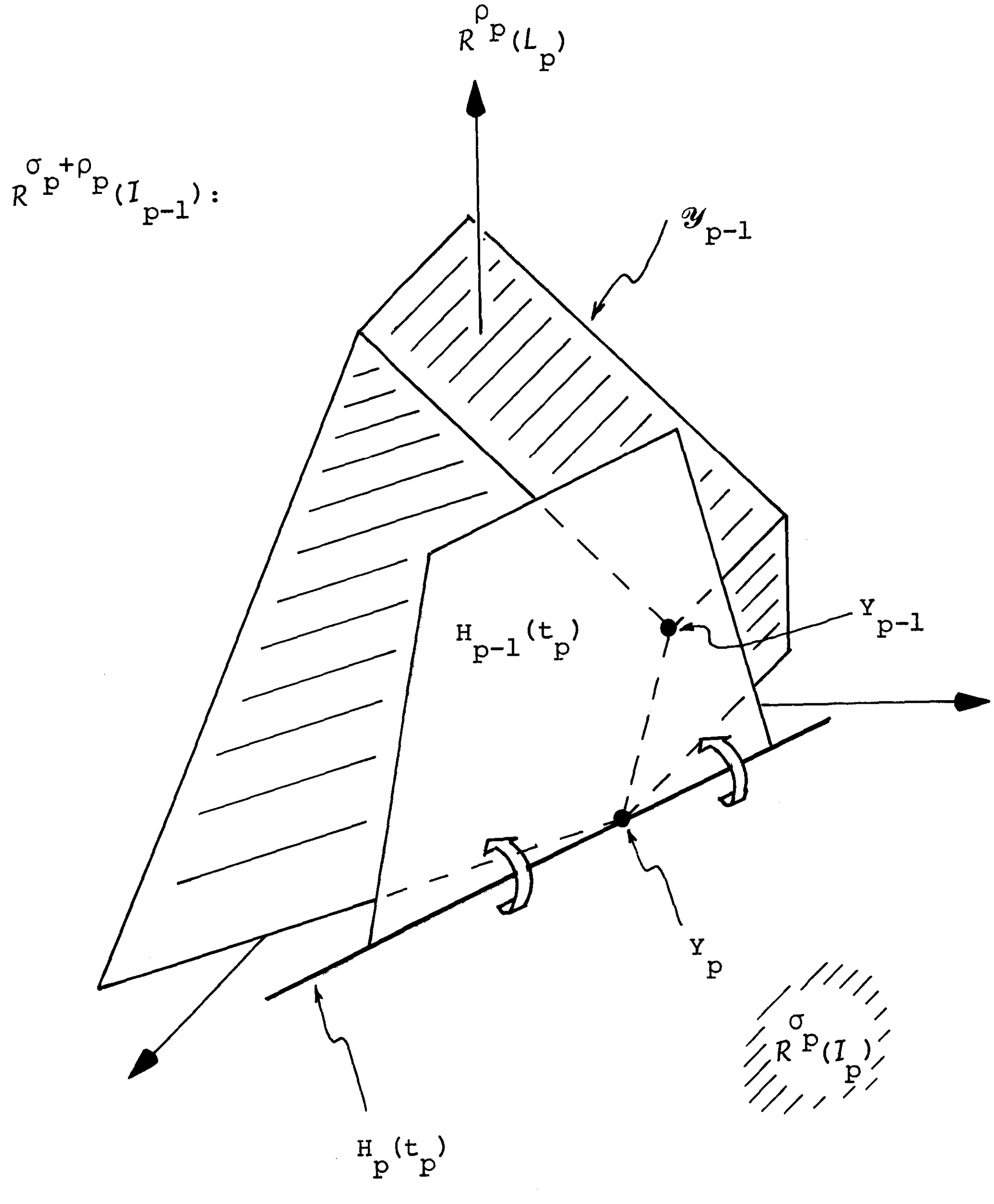


Figure 3.24 Determination of Leave-the-Boundary Costates via Rotation of the Hamiltonian Hyperplane

ciated with the state variables in L_p leaving the boundary backward in time at the potential boundary junction time t_p for all $t_p \varepsilon [\tau_p, t_{p+1})$?

Note that the determination of Ω is equivalent to the determination $Y = \{Y_{-\infty}, \ldots, Y_{p-2}, Y_{p-1}\}$. To get some idea of what is involved here, let us consider the simple case in which we assume that the control does not break as time runs to minus infinity. In this case $\Omega = \{\Omega_{-\infty}\}$, $Y = \{Y_{-\infty}\}$ and there are no breakwalls with which to be concerned. The problem then is to determine the solution set $Y_{-\infty}$ associated with the state variables in L_p leaving the boundary backward in time at the potential boundary junction time t_p for all $t_p \varepsilon [\tau_p, t_{p+1})$. Let us presently fix t_p. Clearly, $Y_{-\infty}$ will then consist of points which lie on L_p-positive faces to which $H_{p-1}(t_p)$ may be rotated (around $H_p(t_p)$). In order to determine exactly which points of these L_p-positive faces are in $Y_{-\infty}$, H_{p-1} must be rotated slightly further by perturbing time backward and solving the constrained optimization problem. In the circumstance that there exists more than a single L_p-positive face to which $H_{p-1}(t_p)$ may be rotated, this procedure must be performed for all such faces in order to determine the total composition of $Y_{-\infty}$. We therefore must "probe ahead" backward in time in order to examine the ensuing optimal solution $Y_{-\infty}$. Furthermore, this must be done for all $t_p \varepsilon [\tau_p, t_{p+1})$.

This is undoubtedly a difficult problem, and does not even take into consideration the general situation in which the control encounters breaks. In that case, we must probe ahead to look at the entire subsequent trajectory on $(-\infty, t_p)$ for every $t_p \varepsilon [\tau_p, t_{p+1})$! Although no technique is known for solving this problem in general, it is solved for a simple three dimensional problem in Section 3.3.4.3.

We have now provided a geometrical interpretation of the three basic problems associated with the algorithm. Unfortunately, we do not know any way in which to apply the suggested techniques in general. However, in Section 4.3 we utilize the geometrical approach to prove several simplifications which permit a straightforward computational scheme. The special case involves single destination networks with all unity weightings in the cost functional. The simplifications are:

(i) Every constrained solution is also globally optimal.

(ii) The set of leave-the-boundary costates is unique; that is, there is a unique L_p-positive face which can be reached by $H_{p-1}(t_p)$. In this case, it is possible to devise a computational technique for the determination of the values of the leave-the-boundary costates.

(iii) There is only one subregion per region.

3.3.3.3 Construction of Non-Break Feedback Control Regions

In this section we specify the construction of feedback control regions of the type $\mathscr{R}_{-\infty}$ with associated control set $\Omega_{-\infty}$. In specific, we assume that the set of state variables in L_p are leaving either a

<u>subregion</u> $\mathscr{R}_p^r(L_p)$ of some region $\mathscr{R}_p$ or the <u>breakwall</u> of a final break feedback control region $\mathscr{R}_{p-q}$. The set of optimal controls with which the state variables in L_p leave persists without a switch until time equals minus infinity, and is therefore denoted $\Omega_{-\infty}$.

In Corollary 3.3 we showed that all feedback control regions are convex polyhedral cones. It can be shown without much difficulty that this characterization also applies to both subregions and breakwalls. Also, by Definition 3.8 we are assured that the state variables in L_p leave backward in time from any point in a given subregion or breakwall with the same set of optimal controls $\Omega_{-\infty}$. There is consequently no need to differentiate between the situation where the state variables in L_p leave backward in time from either a subregion or a breakwall. We therefore introduce the following simplifying notation:

<u>Notation 3.15</u>: Both subregions and breakwalls of $\mathscr{R}_p$ are referred to as <u>previous regions</u>, denoted $\hat{\mathscr{R}}_p$, when the intent is to allow the state variables in L_p to leave from them backward in time.

Recall that $Y_{-\infty}$ is the $\underline{y}$-space solution set corresponding to $\Omega_{-\infty}$. Since $Y_{-\infty}$ is the solution set to a linear program, it is completely described by its extreme points.

<u>Definition 3.10:</u> For every extreme point of $Y_{-\infty}$, form the ray in $R^{\sigma_p+\rho_p}$ which originates at the origin and passes through that extreme point. If there are ω extreme points, the set of rays is denoted

$$V_{-\infty} = \{v_1, v_2, \ldots, v_\omega\}.$$

This definition is illustrated in Figure 3.25.

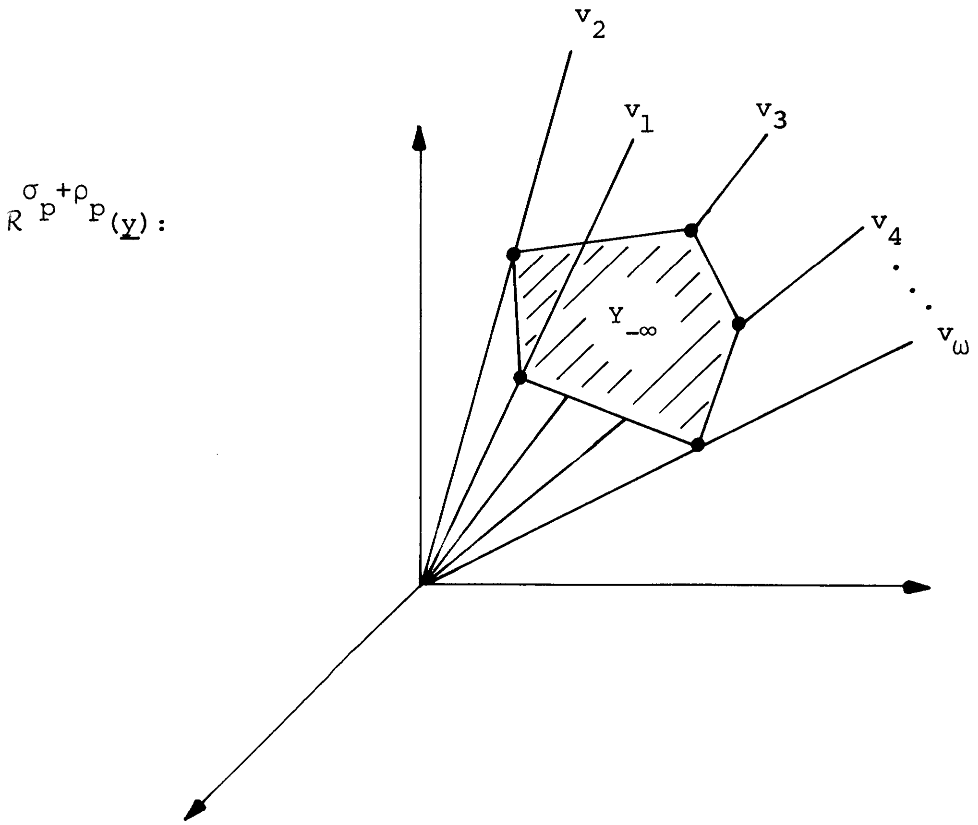


Figure 3.25 The set of rays $V_{-\infty}$

The rays of Definition 3.10 have a very special significance. Consider the ray v which corresponds to the extreme point $\underline{y} \in Y_{-\infty}$. In turn, $\underline{y}$ corresponds to some optimal control $\underline{u} \in \Omega_{-\infty}$ via

$$\underline{y} = -\dot{\underline{x}} = -\underline{B}\ \underline{u} - \underline{a}. \tag{3.184}$$

Therefore, for any interval of time Δt between switch times we have

$$\underline{y} = -\frac{\Delta\underline{x}}{\Delta t} \tag{3.185}$$

where $\Delta\underline{x}$ represents the direction of travel in forward time of the state under the influence of the optimal control $\underline{u}$ in the presence of inputs $\underline{a}$. Therefore, (3.185) tells us that $\underline{y}$ represents the corresponding optimal direction in $\underline{x}$-space in which to leave the previous region $\hat{\mathscr{R}}_p$. By definition 3.10, v is colinear with $\underline{y}$. Therefore, if we consider v to reside in $\underline{x}$-space, it is parallel to the optimal direction of travel of the state. See Figure 3.26.

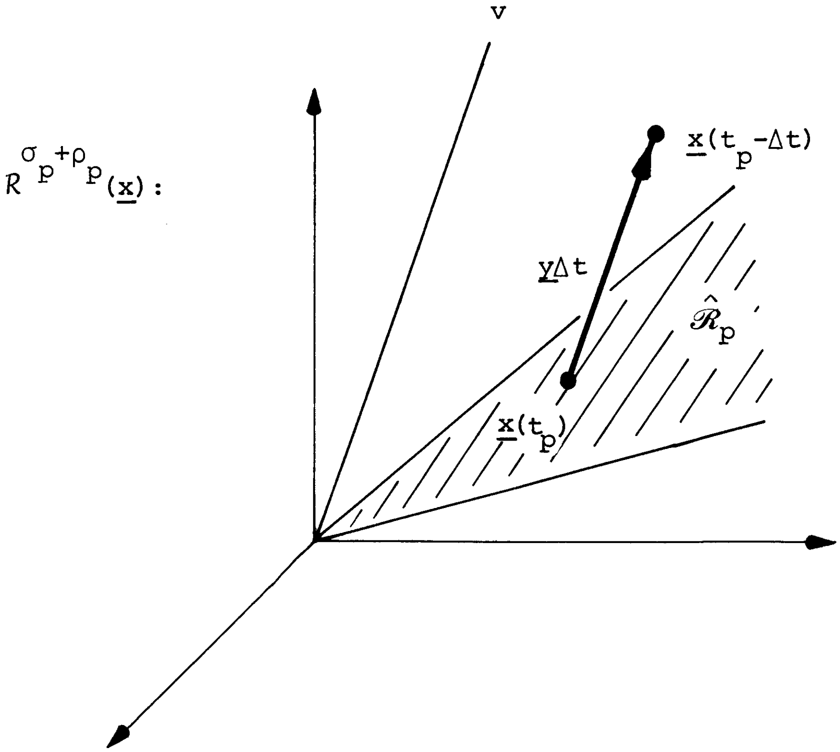

$R^{\sigma_p+\rho_p}(\underline{x})$:
v
$\underline{x}(t_p-\Delta t)$
$\underline{y}\Delta t$
$\hat{\mathscr{R}}_p$
$\underline{x}(t_p)$

Figure 3.26 Interpretation of the ray v

Note that any optimal direction for leaving $\hat{\mathscr{R}}_p$ is expressible as a convex combination of members of $V_{-\infty}$ and any convex combination of members of $V_{-\infty}$ is an optimal direction for leaving $\hat{\mathscr{R}}_p$.

We now denote by $Co(\mathscr{V})$ the convex hull of all the points in the set $\mathscr{V}$. The following is the basic constructive theorem for non-break feedback control regions.

Theorem 3.6 (Construction of non-break feedback control regions)

Suppose $\Omega_{-\infty}$ is the set of controls with which the state variables in L_p leave $\hat{\mathscr{R}}_p$. Then

$$\mathscr{R}_{-\infty} = Co(\hat{\mathscr{R}}_p \cup V_{-\infty})/\hat{\mathscr{R}}_p \tag{3.186}$$

is the non-break feedback control region with associated control set $\Omega_{-\infty}$ in the vein of Definition 3.6.

Proof: We must show that items (i), (ii) and (iii) of Definition 3.6 apply to $\mathscr{R}_{-\infty}$ and $\Omega_{-\infty}$. The situation is depicted in Figure 3.27. Items (i) and (ii) follow from the fact that $\hat{\mathscr{R}}_p$ is itself a feedback control region.

To prove item (iii) consider any point $\underline{x} \in \hat{\mathscr{R}}_p$. Translate each ray in V by placing its origin at $\underline{x}$ and call the translated set $V'_{-\infty} = \{v'_1, v'_2, \ldots, v'_\omega\}$. Next form the conical region $\mathscr{h}(\underline{x}) = Co(\underline{x} \cup V'_{-\infty})/\underline{x}$. See Figure 3.27. If $\underline{x}_1 \in \mathscr{h}(\underline{x})$, then there exists a direction which is some convex combination of the members of $V'_{-\infty}$ which takes $\underline{x}_1$ optimally to $\underline{x}$. Hence, for any $\underline{x}_1 \in \mathscr{h}(x)$ there exists a

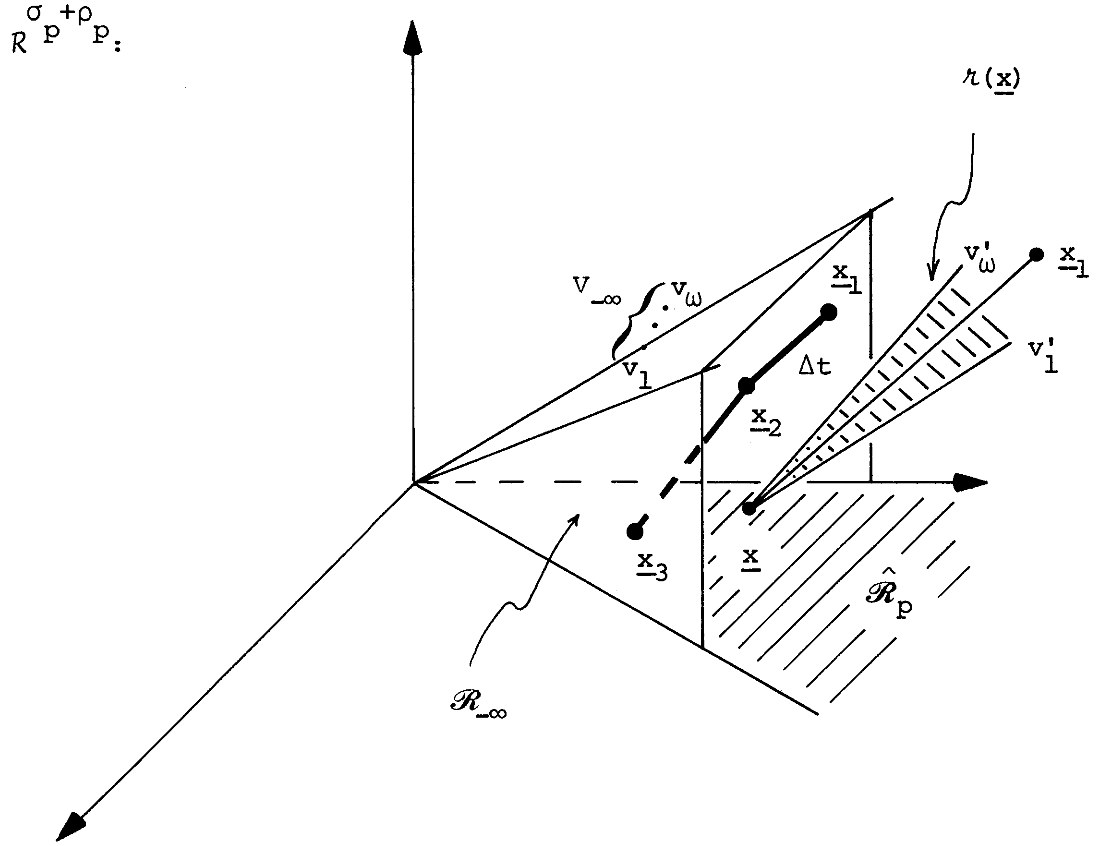


Figure 3.27 Geometry for Proof of Theorem 3.6

$\underline{u} \in \Omega_{-\infty}$ which takes $\underline{x}_1$ optimally to $\underline{x}$. Now, $\mathscr{R}_{-\infty} = \text{Co}(\{\mathscr{r}(\underline{x}) \mid \underline{x} \in \hat{\mathscr{R}}_p\})$ since the smallest convex set containing $\{\mathscr{r}(\underline{x}) \mid \underline{x} \in \hat{\mathscr{R}}_p\}$ is clearly $\mathscr{R}_{-\infty}$. Therefore, for any $\underline{x}_1 \in \mathscr{R}_{-\infty}$ there exists some direction which is a convex combination of members of $V_{-\infty}$ which carries $\underline{x}_1$ to some point $\underline{x} \in \hat{\mathscr{R}}_p$. This is equivalent to saying that for any $\underline{x}_1 \in \mathscr{R}_{-\infty}$, there exists a $\underline{u} \in \Omega_{-\infty}$ such that $\dot{\underline{x}} = \underline{B}\,\underline{u} + \underline{a}$ carries $\underline{x}_1$ to some point $\underline{x} \in \hat{\mathscr{R}}_p$.

Now, let us select some $\underline{x}_1 \in \mathscr{R}_{-\infty}$ and apply any control $\underline{u}_1 \in \Omega_{-\infty}$ for any period of time Δt which keeps the resulting state $\underline{x}_2$ in $\mathscr{R}_{-\infty}$. Then by the above argument there exists some control $\underline{u}_2 \in \Omega_{-\infty}$ which takes $\underline{x}_2$ to some point $\underline{x}_3 \in \hat{\mathscr{R}}_p$. See Figure 3.27. Now, the control $\underline{u}_2$ is clearly optimal since by hypothesis any $\underline{u} \in \Omega_{-\infty}$ is optimal to move the state off of $\hat{\mathscr{R}}_p$. Finally, $\underline{u}_1$ is optimal since the trajectory segment $\underline{x}_2 \to \underline{x}_1$ is part of the trajectory $\underline{x}_3 \to \underline{x}_2 \to \underline{x}_1$ which leaves from $\hat{\mathscr{R}}_p$. We have therefore showed that item (iii) of Definition 3.6 is satisfied. ■ Theorem 3.6

As $\hat{\mathscr{R}}_p$ is a convex polyhedral cone, the convex hull construction of Theorem 3.6 results in the convex polyhedral cone $\mathscr{R}_{-\infty}$. This is in agreement with Corollary 3.3. In brief, the <u>construction of non-break feedback regions proceeds as follows:</u>

(1) Begin with non-break control solution set $\Omega_{-\infty}$ corresponding to the state variables in L_p leaving the previous region $\hat{\mathscr{R}}_p \subset R^{\sigma_p}$ backward in time.

(2) Find the image $Y_{-\infty}$ of $\Omega_{-\infty}$ under the mapping (3.162).

(3) Form the set of rays $V_{-\infty}$ from the extreme points of $Y_{-\infty}$.

(4) Form the conical region in $R^{\sigma_p+\rho_p}$:

$$\mathscr{R}_{-\infty} = \mathrm{Co}(\hat{\mathscr{R}}_p \cup V_{-\infty})/\hat{\mathscr{R}}_p.$$

The reader may verify that all of the constructions performed in Example 3.1 are elementary applications of Theorem 3.6. In Appendix B, non-break feedback regions are constructed for a three dimensional example.

3.3.3.4 Construction of Break Feedback Control Regions

In this section we specify the construction of a feedback control region associated with a control set which applies on the interval $[\tau_{p-1}, \tau_p)$. To be more specific, we assume that the state variables in L_p are leaving backward in time from either a subregion $\mathscr{R}_p^r(L_p)$ of some feedback control region $\mathscr{R}_p$ or the breakwall of a previously constructed break feedback region. In either case, we invoke the convention of Section 3.3.3.3 and say that the state variables in L_p are leaving backward in time from the previous region $\hat{\mathscr{R}}_p$. Our goal is to construct the break feedback region $\mathscr{R}_{p-1}$ associated with the control set Ω_{p-1}.

The approach differs from that for non-break feedback regions principally in the definition of the set of rays. We now wish each ray to represent an edge of the breakwall of $\mathscr{R}_{p-1}$. To find such rays, we reason that each edge of the breakwall serves as the break surface of a trajectory emanating from some edge of $\hat{\mathscr{R}}_p$. Additionally, this trajectory will correspond to an extreme point of Y_{p-1}, although there is no way of determining a priori the appropriate one. We therefore define the set of rays for this case as follows:

Definition 3.11: Consider all edges of the previous region $\hat{\mathscr{R}}_p$. Allow the state variables in L_p to leave backward in time from some point on each edge (where the point may not be the origin) once for each extreme point direction of Y_{p-1} (these correspond to extreme points of Ω_{p-1}). Let $\underline{x}(\tau_p)$ represent the point of departure on a given edge. Then for each trajectory find the break point

$$\underline{x}(\tau_{p-1}) = \underline{x}(\tau_p) + (\tau_p - \tau_{p-1})(\underline{B}\,\underline{u}_{p-1} + \underline{a}) \tag{3.187}$$

where $\underline{u}_{p-1}$ is the appropriate extreme point of Ω_{p-1}. For every such break point, form the ray in $R^{\sigma_p+\rho_p}$ which originates at the origin and and passes through the break point. If the total number of break points is ω, the set of rays is denoted

$$V_{p-1} = \{v_1, v_2, \ldots, v_\omega\}.$$

See Figure 3.28.

Theorem 3.7 (Construction of break feedback regions)

Suppose Ω_{p-1} is the set of controls with which the state variables in L_p leave $\hat{\mathscr{R}}_p$ backward in time. Then

$$\mathscr{R}_{p-1} = \text{Co}(\hat{\mathscr{R}}_p \cup V_{p-1})/\hat{\mathscr{R}}_p \tag{3.188}$$

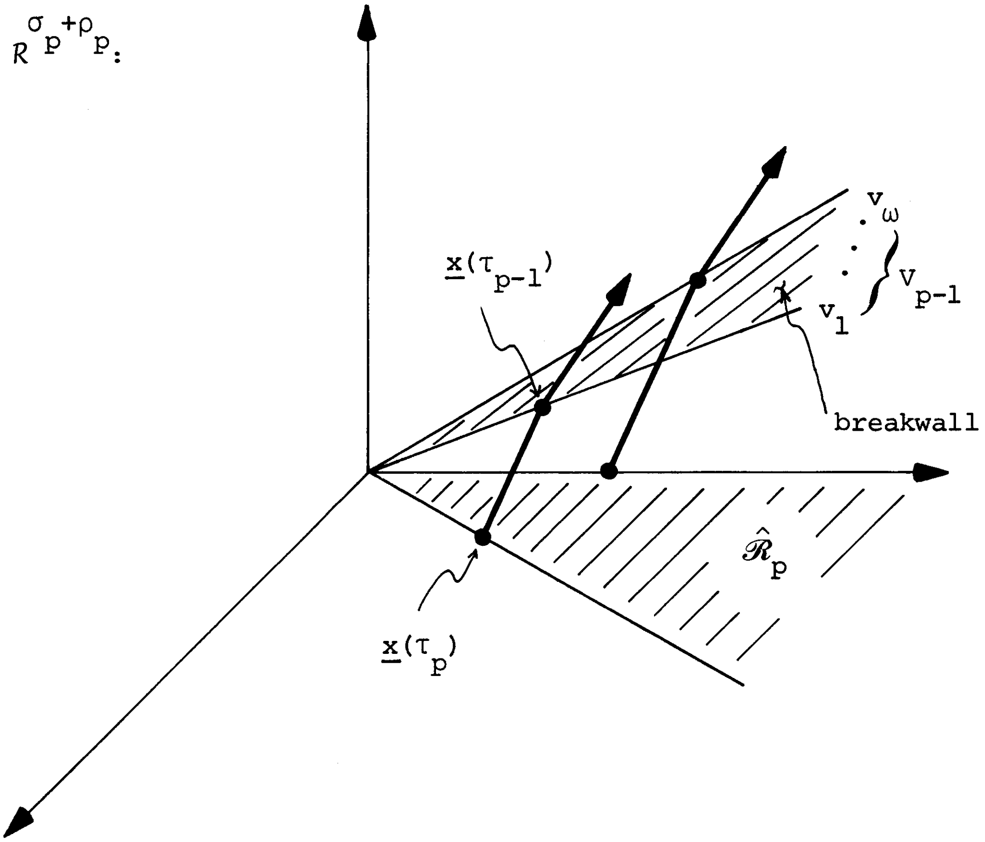


Figure 3.28 The set of rays V_{p-1}

is the break feedback control region with associated control set Ω_{p-1} in the vein of Definition 3.6.

Proof: We must show that items (i), (ii) and (iii) of Definition 3.6 apply to $\mathscr{R}_{p-1}$ and Ω_{p-1}. The situation is depicted in Figure 3.28.

Items (i) and (ii) follow from the fact that $\hat{\mathscr{R}}_p$ is itself a feedback control region.

To prove item (iii) consider any point $\underline{x} \in \hat{\mathscr{R}}_p$. Form the set of rays from Y_{p-1} as specified in Definition 3.10 and call them V. As discussed previously, these rays represent the directions of the optimal trajectories which result from the application of the extreme point controls of Ω_{p-1}. Moreover, the direction of the trajectory corresponding to any control in Ω_{p-1} can be expressed as a convex combination of of the rays in V. Next, translate each ray in V by placing its origin at the point $\underline{x}$ and call the translated set V'. Denote by S the closed halfspace of $R^{\sigma_p+\rho_p}$ which lies on the same side of the hyperplane which contains the break wall as does $\underline{x}$. Consider the region $\mathcal{h}(\underline{x}) = (\mathrm{Co}(\hat{\mathscr{R}}_p \cup V')/\underline{x}) \cap S$. See Figure 3.29. If $\underline{x}_1 \in \mathcal{h}(\underline{x})$, then there exists a direction which is some convex combination of the members of V' which takes $\underline{x}_1$ optimally to $\underline{x}$. Hence, for any $\underline{x}_1 \in \mathcal{h}(\underline{x})$ there exists a $\underline{u} \in \Omega_{p-1}$ which takes $\underline{x}_1$ optimally to $\underline{x}$. Now, $\mathscr{R}_{p-1} = \mathrm{Co}(\{\underline{\mathcal{h}}(x) \mid \underline{x} \in \hat{\mathscr{R}}_p\})$ since the smallest convex set containing $\{\underline{\mathcal{h}}(x) \mid \underline{x} \in \hat{\mathscr{R}}_p\}$ is $\mathscr{R}_{p-1}$. We have therefore shown that for any $\underline{x}_1 \in \mathscr{R}_{p-1}$, there exists a $\underline{u} \in \Omega_{p-1}$ such that $\dot{\underline{x}} = \underline{B}\ \underline{u} + \underline{a}$ carries $\underline{x}_1$ to some point $\underline{x} \in \hat{\mathscr{R}}_p$.

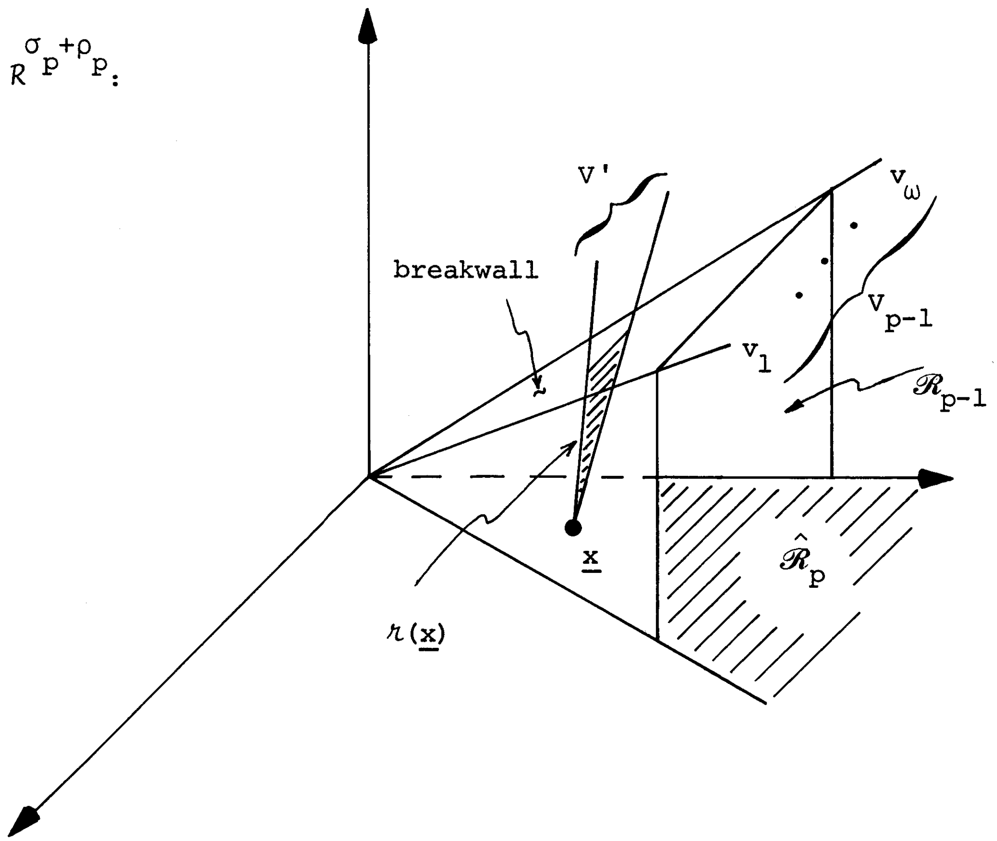


Figure 3.29 Geometry for the Proof of Theorem 3.7

The reasoning from this point on follows precisely the same lines as that presented at this point in the proof of Theorem 3.6.

■ Theorem 3.7

As before, we obtain the fact that $\mathscr{R}_{p-1}$ as constructed in Theorem 3.7 is a convex polyhedral cone. In brief, the construction of break feedback regions proceeds as follows:

(1) Begin with the break control solution set Ω_{p-1} corresponding to the state variables in L_p leaving the previous region $\hat{\mathscr{R}}_p \subseteq R^{\sigma_p}$ backward in time.

(2) Let the state variables in L_p leave backward in time from each edge of $\hat{\mathscr{R}}_p$ with every extreme point control in Ω_{p-1} and calculate the resulting break points from (3.187).

(3) Form the set of rays V_{p-1} from the break points determined in (2).

(4) Form the conical region in $R^{\sigma_p+\rho_p}$:

$$\mathscr{R}_{p-1} = \mathrm{Co}(\hat{\mathscr{R}}_p \cup V_{p-1})/\hat{\mathscr{R}}_p.$$

The reader may verify that the construction of the break feedback region labeled ② in Figure 3.12 is a simple application of Theorem 3.7. In Appendix B, a break feedback region is constructed for a three dimensional example.

3.3.4 Discussion of Fundamental Properties of the Algorithm

In this section, we discuss several of the properties of the algorithm presented in Section 3.3.2.2. We are particularly interested in those properties which are troublesome from the point of view of constructing a computational scheme to implement the algorithm. The discussion is focused on a simple example presented for each property, which serves not only to illustrate the particular problem at hand but also to give the reader an opportunity to fix the essential notion more firmly in his mind. In these examples we utilize the geometrical point of view presented in Section 3.3.3.2, so that it is possible to study the properties by examining the activity of the pointwise linear program in $\underline{y}$-space.

3.3.4.1 Non-Global Optimality of Certain Sequences

Part (a) of Operation 1 of the algorithm consists of solving the constrained optimization problem with a restricted Hamiltonian. In part (b) we assert the necessity to determine if the solution to part (a) can be made globally optimal with the appropriate choice of costates which satisfy the necessary conditions. In this section, we present a simple example in which a constrained optimum is not a global optimum, thus justifying our concern for investigating global optimality.

Example 3.4

The general network topology to be considered in this and several other examples is depicted in Figure 3.30. The link capacities are indicated in brackets and for simplicity the inputs are all taken to be

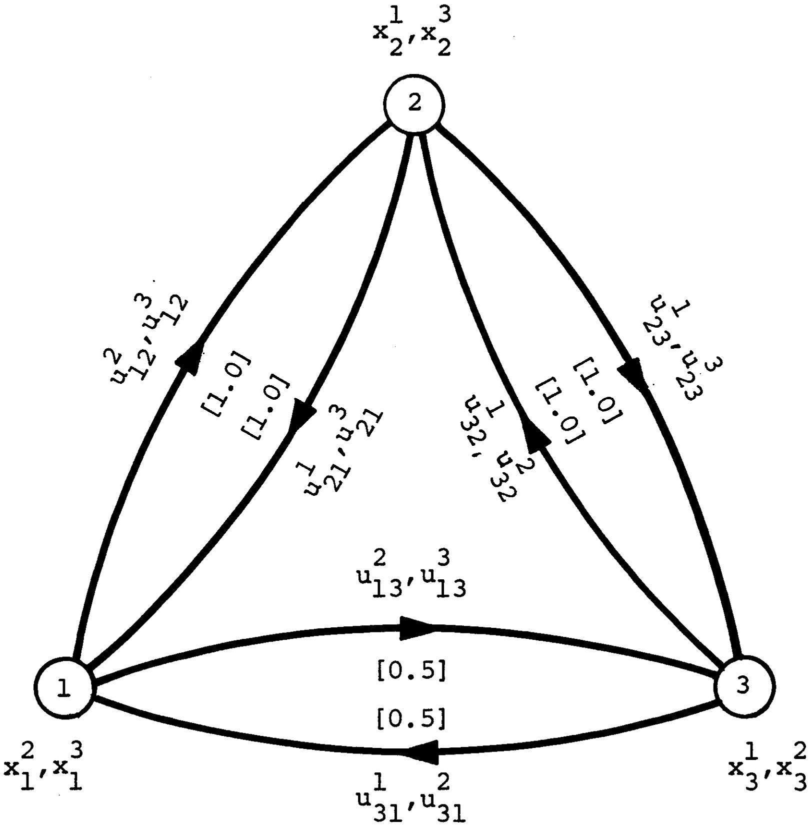


Figure 3.30 Network Topology for Example 3.4

zero. The equations of motion are

$$\begin{aligned}
\dot{x}_1^2(t) &= -u_{12}^2(t) - u_{13}^2(t) + u_{31}^2(t)\\
\dot{x}_2^1(t) &= -u_{21}^1(t) - u_{23}^1(t) + u_{32}^1(t)\\
\dot{x}_1^3(t) &= -u_{13}^3(t) - u_{12}^3(t) + u_{21}^3(t)\\
\dot{x}_3^1(t) &= -u_{31}^1(t) - u_{32}^1(t) + u_{23}^1(t)\\
\dot{x}_2^3(t) &= -u_{23}^3(t) - u_{21}^3(t) + u_{12}^3(t)\\
\dot{x}_3^2(t) &= -u_{32}^2(t) - u_{31}^2(t) + u_{13}^2(t).
\end{aligned} \tag{3.189}$$

For the purpose of this example, we limit attention to the state variables x_3^1 and x_2^1 and consider the cost function

$$J = \int_{t_0}^{t_f} [2x_2^1(t) + x_3^1(t)]\,dt. \tag{3.190}$$

We are therefore considering a single destination network with non-unity weightings in the cost functional. The $\underline{y}$-space constraint figure is presented in Figure 3.31.

We begin by letting x_2^1 leave the boundary backward in time at t_f. Then the constrained optimization problem in $\underline{y}$-space calls for the maximization of the zero dimensional restricted Hamiltonian $H_{f-1}(\tau):z(\tau) = \lambda_2^1(\tau)y_2^1$ over the constraint region $\mathscr{Y}_{f-1}$ depicted in Figure 3.31. The costate trajectory for λ_2^1 is shown in Figure 3.32.

The solution to the constrained optimization problem is $Y_{f-1}:y_2^1 = 1.5,\ y_3^1 = 0$ as illustrated in Figure 3.31. Now, at arbitrary

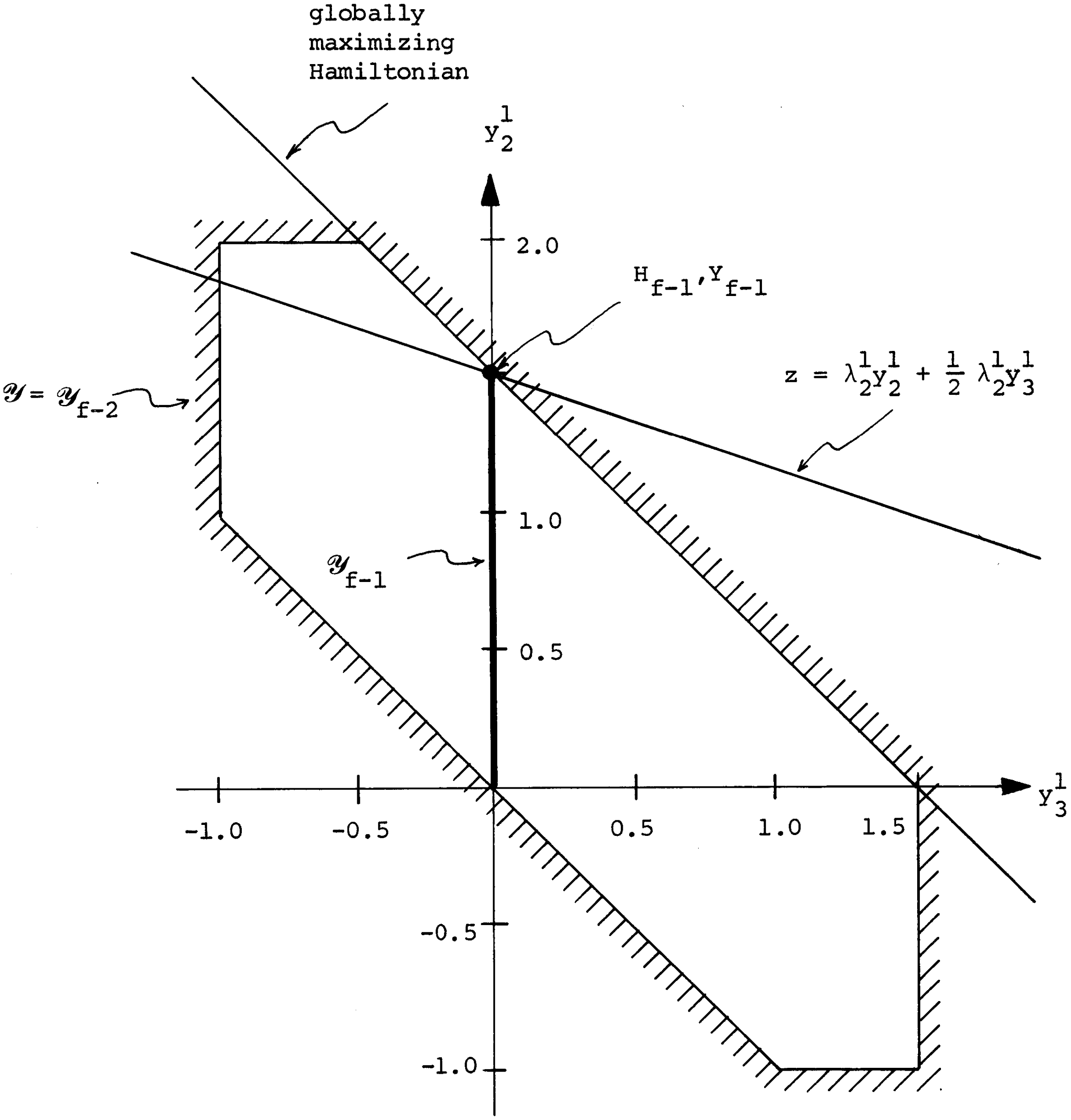

globally maximizing Hamiltonian
y_2^1
2.0
H_{f-1}, Y_{f-1}
$z = \lambda_2^1 y_2^1 + \frac{1}{2} \lambda_2^1 y_3^1$
$\mathscr{Y} = \mathscr{Y}_{f-2}$
1.0
$\mathscr{Y}_{f-1}$
0.5
-1.0
-0.5
0.5
1.0
1.5
y_3^1
-0.5
-1.0

Figure 3.31 Geometrical Interpretation of Sequence Which is Not Globally Optimal

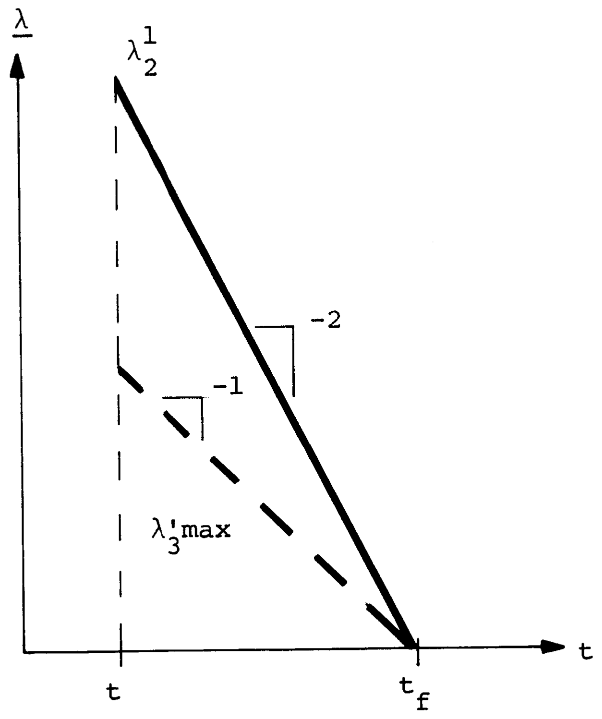


Figure 3.32 Costate Trajectories for Example 3.4

time $t < t_f$ let us attempt to see if there exists some legitimate value of $\lambda_3^1(t)$ such that for $\lambda_2^1(t) = 2(t_f - t)$ the Hamiltonian $H(t):z(t) = \lambda_2^1(t)y_2^1 + \lambda_3^1(t)y_3^1$ maximizes $\mathscr{Y}$ at Y_{f-1}. From Figure 3.31 we see that the <u>only</u> possible globally maximizing Hamiltonian has $\lambda_3^1(t) = \lambda_2^1(t)$. However, from (3.153) we find that the maximum value of $\lambda_3^1(t)$ is $(t_f - t)$, that is, one half the value of $\lambda_2^1(t)$. Therefore, the candidate operating point Y_{f-1} obtained from the constrained optimization cannot be globally optimal. We then conclude that it is never optimal for x_2^1 to strike the boundary last forward in time.

● Example 3.4

3.3.4.2 Non-Uniqueness of Leave-the-Boundary Costates

Refer to Definition 3.9 for the definition of $\mathcal{L}_p$-positive face. The significance of this notion is established in Theorem 3.5, where we show that the values of the leave-the-boundary costates at t_p are actually those of the appropriate coefficients of $H_{p-1}(t_p)$ rotated about $H_p(t_p)$ until it touches $\mathcal{Y}_{p-1}$ on an $\mathcal{L}_p$-positive face. Here we establish the fact that there may be more than one $\mathcal{L}_p$-positive face which may be reached in this fashion. As we shall see, this situation implies that there may be an infinitely non-unique set of costates corresponding to state variables which leave the boundary and enter onto interior arcs. Therefore, state variables which are travelling on interior arcs may have associated with them costates whose values are non-unique. This is a most interesting property which characterizes the linear state constrained optimal control problem. Although non-uniqueness of costates may occur in non-linear state constrained problems, it is usually limited to costates corresponding to state variables travelling on boundary arcs. See Bryson, Denham and Dreyfus [1963].

Example 3.5

Once again consider the network topology of Figure 3.30.

Let us now limit attention to the states x_3^1, x_3^2 and x_2^1 and take the cost functional to be

$$J = \int_{t_0}^{t_f} [x_3^1(t) + x_3^2(t) + x_2^1(t)]dt. \tag{3.191}$$

The $\underline{y}$-space constraint figure $\mathscr{Y}$ is pictured in Figure 3.33. This figure can be obtained by finding the extreme points of $\mathcal{U}$ and transforming them to $\underline{y}$-space via equation (3.162).

Suppose we let x_3^1 leave the boundary backward in time at t_f. Then $H_{f-1}(\tau):z(\tau) = \lambda_3^1(\tau)y_3^1$ is maximized at the point $Y_{f-1}:y_3^1 = 1.5$, $y_3^2 = y_2^1 = 0$ for all $\tau \in (-\infty, t_f)$. This solution is easily shown to be globally optimal with $\lambda_2^1(\tau) = \lambda_3^1(\tau) = 0$. The zero dimensional restricted Hamiltonian $H_{f-1}(\tau)$ is actually the point Y_{f-1} in Figure 3.33. This solution does not break as time runs to minus infinity. By a trivial application of Theorem 3.6 the underlying controls ($u_{31}^1 = 0.5$, $u_{32}^1 = 1.0$, $u_{21}^1 = 1.0$, all other controls zero) apply everywhere on the x_3^1 axis.

Suppose we know specify that x_3^2 and x_2^1 leave the boundary simultaneously at some arbitrary boundary junction time $t_{f-1} \in (-\infty, t_f)$. Now, since $\dot{\lambda}_3^1(\tau) = -1$, $\tau \in [t_{f-1}, t_f]$, then $\lambda_3^1(t_{f-1}) = t_f - t_{f-1}$. The L_p-positive faces of $\mathscr{Y}_{f-2}$ (in this case, the L_{f-1}-positive faces) are F_1, F_2 and F_3 of Figure 3.33. Furthermore, the two dimensional Hamiltonian $H_{f-1}(t_{f-1}):z(t_{f-1}) = \lambda_3^1(t_{f-1})y_3^1 + \lambda_3^2(t_{f-1})y_3^2 + \lambda_2^1(t_{f-1})y_2^1$ can be rotated about the zero dimensional Hamiltonian $H_{f-1}(t_{f-1})$ to touch $\mathscr{Y}$ in all faces F_1, F_2 and F_3. In fact, it may lie on F_2 anywhere between F_1 and F_3. Therefore, the leave-the-boundary costates λ_3^2 and λ_2^1 at t_{f-1} may achieve values anywhere between

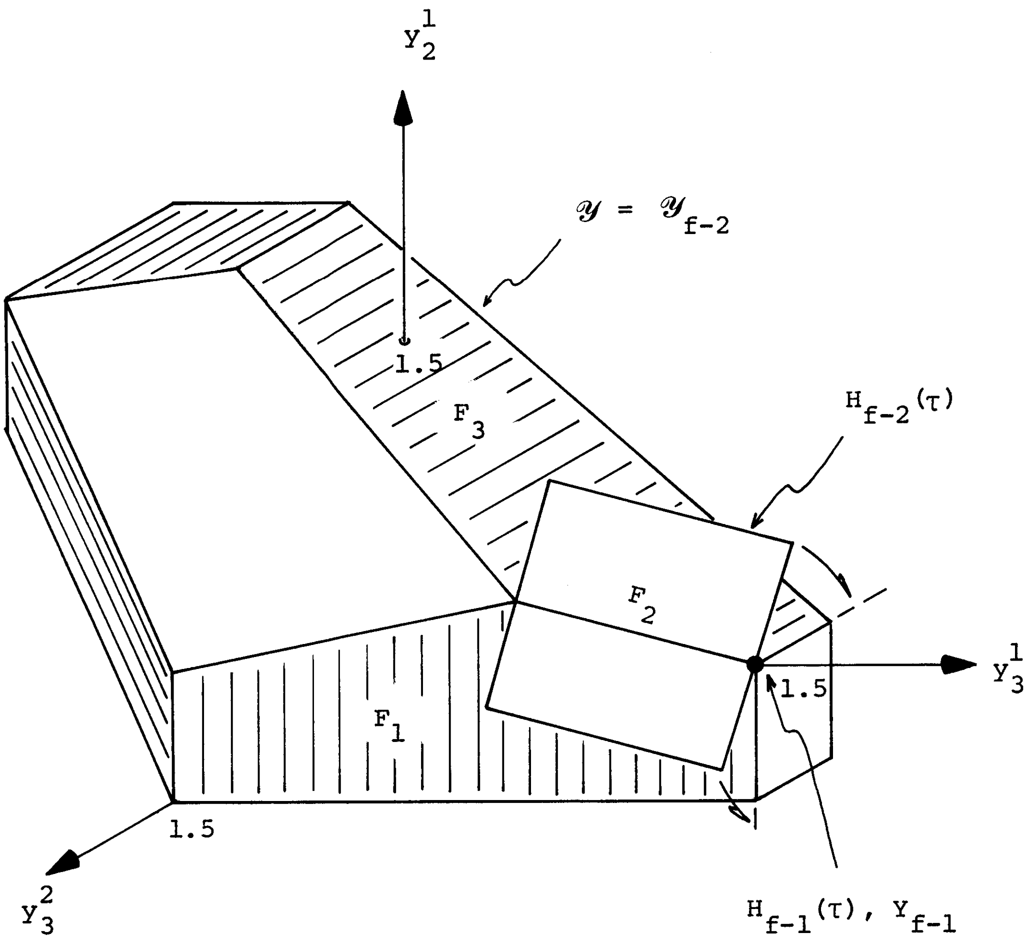


Figure 3.33 Geometrical Demonstration of Non-Uniqueness of Leave-the-Boundary Costates

$$\left\{\begin{aligned} &\lambda_3^2(t_{f-1}) = \lambda_3^1(t_{f-1}) = t_f - t_{f-1} \\ &\lambda_2^1(t_{f-1}) = 0 \end{aligned}\right.$$

and

$$\left\{\begin{aligned} &\lambda_3^2(t_{f-1}) = 0 \\ &\lambda_2^1(t_{f-1}) = \lambda_3^1(t_{f-1}) = t_f - t_{f-1}. \end{aligned}\right.$$

Hence, we have an infinitely non-unique set of leave-the-boundary costates at t_{f-1}. This non-uniqueness persists even after x_3^2 and x_2^1 leave the boundary and enter onto interior arcs, thus verifying our contention in the introduction to this section. ◗ Example 3.5

In general, the backward constructive algorithm calls for the entire set of costate values if they are indeed non-unique. However, it is clear that Hamiltonian hyperplanes which lie between faces of $\mathscr{Y}$ will provide us with no more information in terms of new operating points than will those which actually lie on the faces. If the state variables in L_p leave some region $\hat{\mathscr{R}}_p$, we therefore must find all of the highest dimensional L_p-positive faces of $\mathscr{Y}_{p-1}$ upon which $H_{p-1}(t_p)$ may be made to lie. We then utilize the finite costate set corresponding to the coefficients of $H_{p-1}(t_p)$ which bring it to lie on these L_p-positive faces. In Example 3.5, this costate set corresponds to the faces F_1 and F_3.

The reduction of the required costates to a finite set is a welcome result but we still have the problem of finding the appropriate

L_p-positive faces of $\mathcal{Y}_{p-1}$. Although this is a simple problem for three dimensions or less, we have no method for performing this task for a general dimensional multi-destination network. The solution to this problem is contingent upon discovering a method for explicitly determining the constraints which determine $\mathcal{Y}_{p-1}$ in the vicinity of the operating point Y_p. In Section 4.4 we show how this may be achieved when the L_p-positive face is unique as in the case of problems involving single destination networks.

3.3.4.3 Subregions

Recall that a subregion $\mathcal{R}_p(L_p)$ of a region $\mathcal{R}_p$ is that set of points in $\mathcal{R}_p$ which when taken as the point of departure of the state variables in L_p result in a common control set and common set of break-walls (Definition 3.8). We now present an example in which there are two subregions in a particular feedback control region.

Example 3.6

The network is picture in Figure 3.34. Once again, for simplicity we are considering the no inputs case. This is a single destination network with all messages intended for node 4; therefore, we may eliminate the destination superscript on the state and control variables. The dynamical equations are

$$\begin{aligned} \dot{x}_1(t) &= u_{21}(t) + u_{31}(t) - u_{14}(t) \\ \dot{x}_2(t) &= -u_{21}(t) \\ \dot{x}_3(t) &= -u_{31}(t) \end{aligned} \tag{3.192}$$

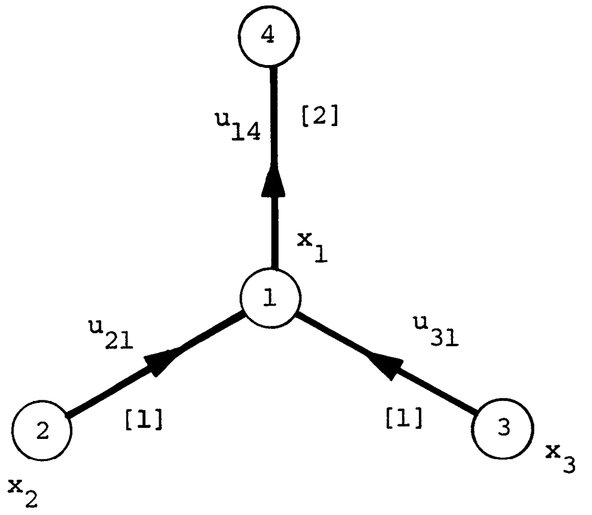


Figure 3.34 Network Topology for Example 3.6

and we consider the cost functional

$$J = \int_{t_0}^{t_f} [2x_1(t) + x_2(t) + 2x_3(t)]dt. \tag{3.193}$$

The $\underline{y}$-constraint figure is depicted in Figure 3.35(a).

We begin by letting x_2 leave the boundary backward in time at t_f. The constrained optimization problem calls for the maximization of the zero dimensional Hamiltonian $H_{f-1}(\tau):z(\tau) = \lambda_2(\tau)y_2$ over the constraint figure $\mathscr{Y}_{f-1}$ depicted in Figure 3.35(a). The costate trajectory for λ_2 is shown in Figure 3.35(b). The solution to the constrained optimization problem is $Y_{f-1}:y_1 = 0$, $y_2 = 1.0$, $y_3 = 0$. Moreover, it is easy to see by examining Figure 3.35(a) that this solution is globally optimal for the costate values $\lambda_1(\tau) = \lambda_2(\tau) = 0$. Also, Y_{f-1} persists

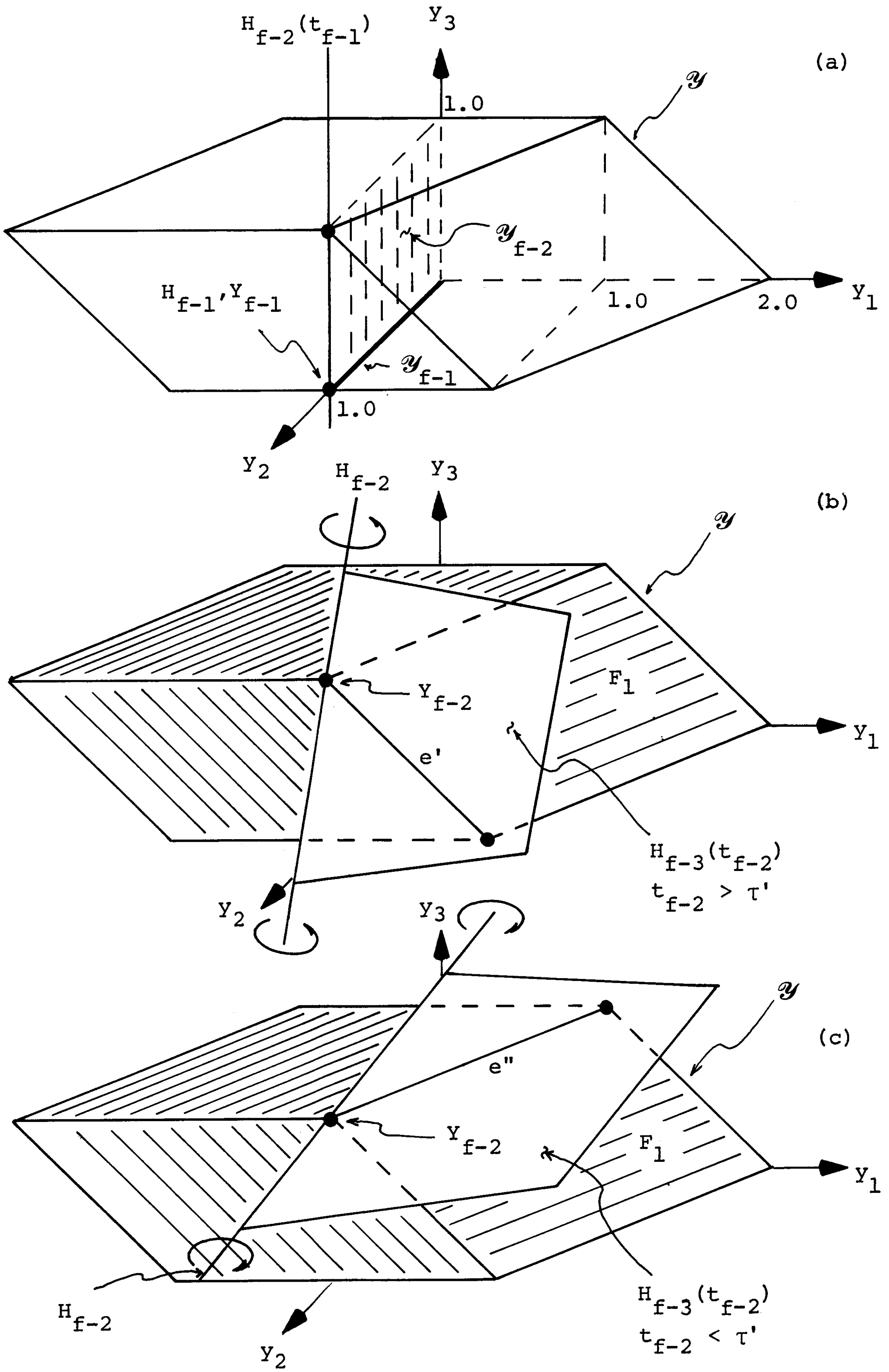


Figure 3.35 Geometry for Example 3.6

as time runs to minus infinity. The trajectory is illustrated in Figure 3.36. Utilizing the controls underlying this solution, we are able to assign the feedback control on the x_2-axis as specified in Figure 3.37. We denote this region $\mathscr{R}_{f-1}$.

Next we stipulate that x_3 leave the boundary at some arbitrary boundary junction time t_{f-1}. The one dimensional Hamiltonian that touches the constraint figure $\mathscr{Y}_{f-2}$ in the positive orthant of y_3 is simply $H_{f-2}(t_{f-1}):z(t_{f-1}) = \lambda_2(t_{f-1})y_2$. Therefore, $\lambda_3(t_{f-1}) = 0$. As we proceed backward in time from t_{f-1}, the one dimensional constrained Hamiltonian $H_{f-2}(\tau):z(\tau) = \lambda_2(\tau)y_2 + \lambda_3(\tau)y_3$ is maximized over $\mathscr{Y}_{f-2}$ at the point $Y_{f-2}:y_1 = 0,\ y_2 = 1.0,\ y_3 = 1.0$. This solution is globally optimal for $\lambda_1(\tau) = 0$ and does not experience a break as time runs to minus infinity. We may then apply Theorem 3.6 to construct the two dimensional non-break feedback control region labeled $\mathscr{R}_{f-2}$ in Figure 3.37.

We now want to allow x_1 to leave the boundary backward in time at some time $t_{f-2} < t_{f-1}$; that is, allow the state to leave from $\mathscr{R}_{f-2}$. This is achieved by rotating the two dimensional Hamiltonian $H_{f-3}(t_{f-2})$: $z(t_{f-2}) = \lambda_1(t_{f-2})x_1 + \lambda_2(t_{f-2})x_2 + \lambda_3(t_{f-2})x_3$ about $H_{f-2}(t_{f-2})$ until it touches $\mathscr{Y}$ on a face in the positive orthant of y_1. By examining Figure 3.35(b) and (c) we distinguish the following cases:

(i) If $\lambda_3(t_{f-2}) < \lambda_2(t_{f-2})$, then $H_{f-3}(t_{f-2})$ is rotated to touch the edge labeled e' in Figure 3.35(b). This edge becomes the

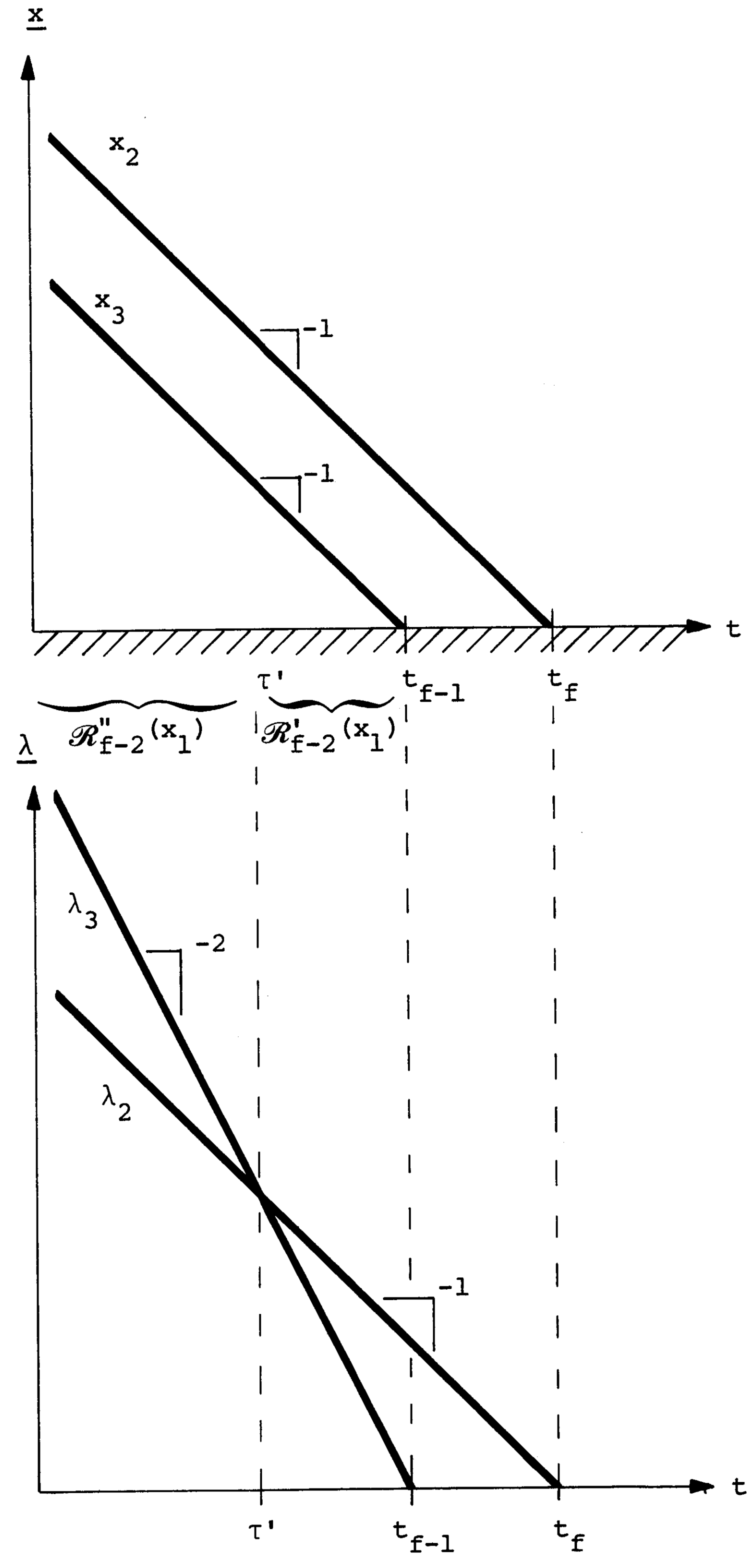


Figure 3.36 State-Costate Trajectory Pair for Example 3.6

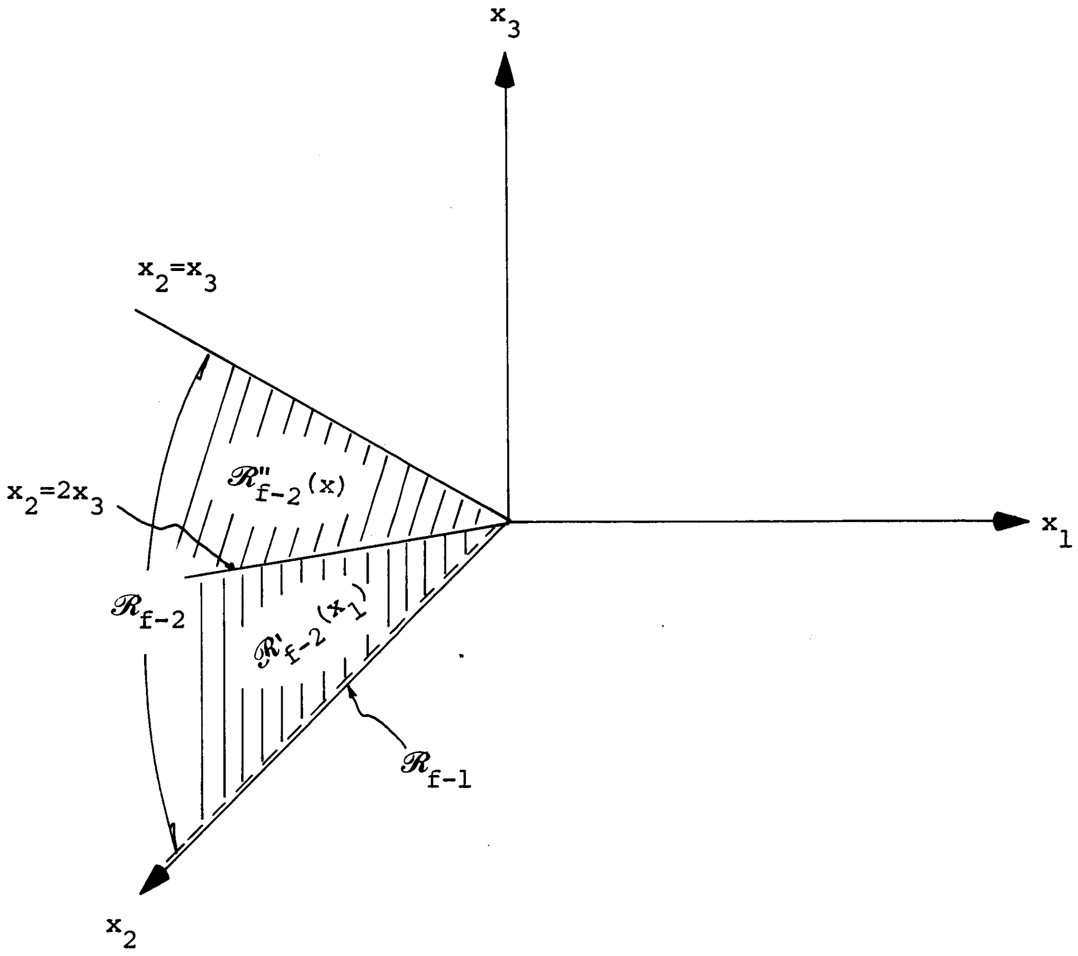


Figure 3.37 The Subregions $\mathscr{R}'_{f-2}(x_1)$ and $\mathscr{R}''_{f-2}(x_1)$ of the Feedback Control Region $\mathscr{R}_{f-2}$

new set of operating points Y_{f-3}.

(ii) If $\lambda_3(t_{f-2}) = \lambda_2(t_{f-2})$, then $H_{f-3}(t_{f-2})$ is rotated to touch the face labeled F_1 in Figure 3.35(c). Subsequent rotation of H_{f-3} causes the edge e" of Figure 3.35(c) to become the new set of operating points Y_{f-3}.

(iii) If $\lambda_3(t_{f-2}) > \lambda_2(t_{f-2})$, then $H_{f-3}(t_{f-2})$ is rotated to touch the edge labeled e" in Figure 3.35(b). This edge becomes the new set of operating points Y_{f-3}.

If we denote the time at which λ_2 equals λ_3 by τ_1, then from Figure 3.36 we easily determine that $x_2(\tau_1) = 2x_3(\tau_1)$. Therefore, we divide the region $\mathscr{R}_{f-1}$ into two subregions: $\mathscr{R}'_{f-2}(x_1)$ is that portion of $\mathscr{R}_{f-1}$ beneath the line $x_2=2x_3$, not including $x_2=2x_3$; $\mathscr{R}''_{f-2}(x_1)$ is that portion of $\mathscr{R}_{f-2}$ above and including the line $x_2=2x_3$. When the state leaves $\mathscr{R}'_{f-2}(x_1)$ the new set of operating points is $Y_{f-3} = e'$. On the other hand, when the state leaves $\mathscr{R}''_{f-2}(x_1)$ the new set of operating points is $Y_{f-3} = e''$. ● Example 3.6

3.3.4.4 Return of States to Boundary Backward in Time

In Operation 1 of the algorithm we work under the assumption that no state variable which is off the boundary must subsequently return to the boundary backward in time. This is equivalent to assuming that in forward time optimality will never require that a state variable increase its current value. In Section 4.3 we show that this assumption is in fact true for the case of single destination networks with all unity weightings in the cost functional. However, in general this assumption is not true as illustrated by the following example.

Example 3.7

We once again refer to the network topology of Example 3.4 and concern ourselves with the state variables x_2^1 and x_3^1 and the cost functional (3.190) of Example 3.4. As in that case, the $\underline{y}$-constraint figure appears as in Figure 3.31, which we present again in Figure 3.38 for illustration on the current discussion.

In Example 3.4, we attempted to allow x_2^1 to leave the boundary backward in time at t_f and discovered that this trajectory cannot be globally optimal. We now try letting x_3^1 leave the boundary backward in time at t_f. Then the constrained optimization problem calls for the maximization of the zero dimensional restricted Hamiltonian $H_{f-1}(\tau)$: $z(t) = \lambda_3^1(\tau)y_3^1$ over the constraint figure $\mathcal{Y}_{f-1}$ depicted in Figure 3.38. The costate trajectory is shown in Figure 3.39. The solution to the constrained optimization problem is $Y_{f-1}: y_3^1 = 1.5,\ y_2^1 = 0$. This solution can be shown to be globally optimal for the costate values $\lambda_3^1(\tau) = \lambda_2^1(\tau),\ \tau \in [t_{f-1}, t_f]$. See Figure 3.38. Also, Y_{f-1} persists as time runs to minus infinity. Over this interval x_3^1 travels forward in time with a slope of -1.5 as illustrated in Figure 3.39.

Now, suppose at an arbitrary boundary junction time t_{f-1} we choose to allow x_2^1 to leave the boundary backward in time. Then the costate $\lambda_3^1(t_{f-1}) = \lambda_2^1(t_{f-1}) = t_f - t_{f-1}$ causes the enlarged Hamiltonian

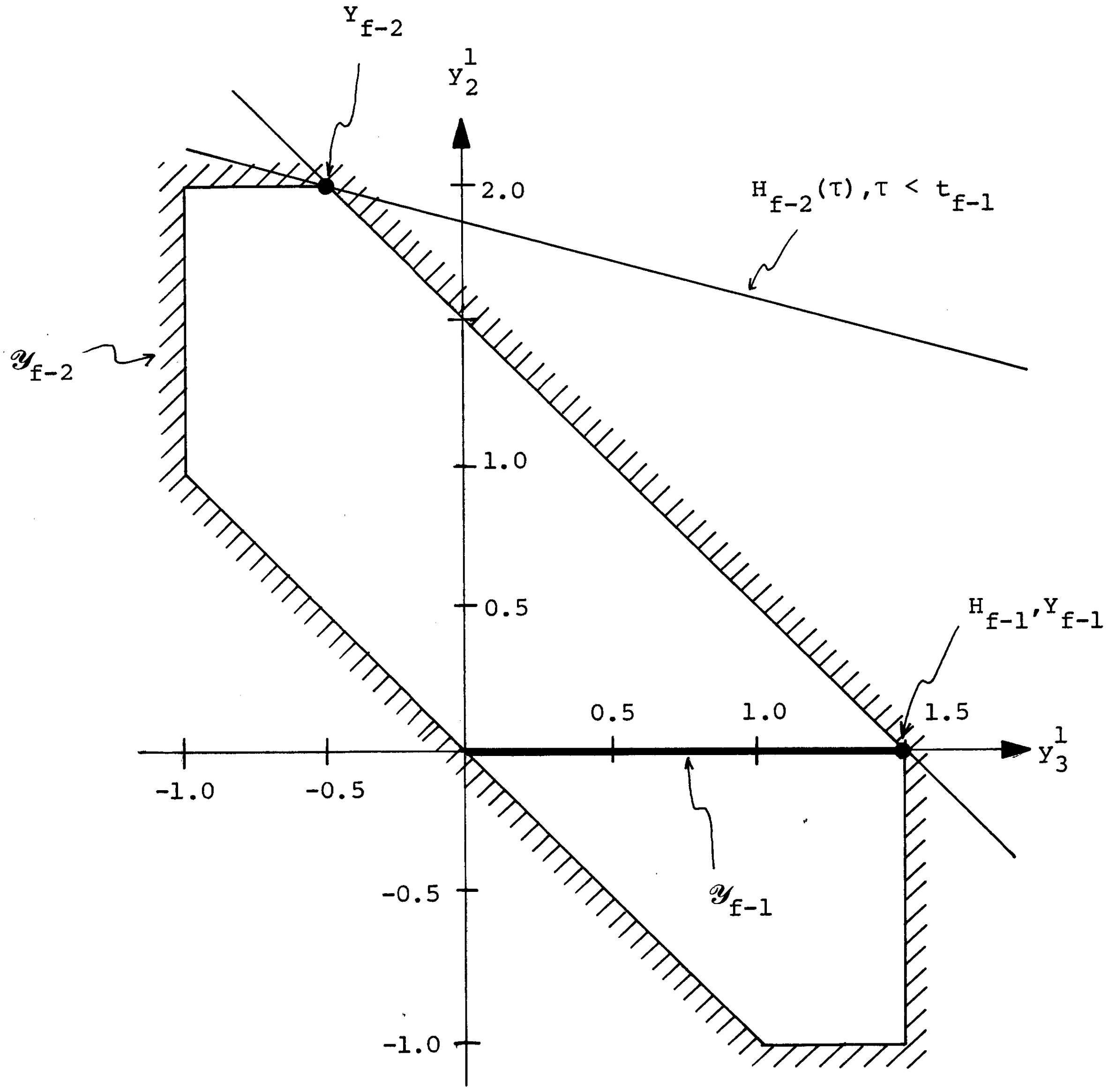


Figure 3.38 Geometrical Interpretation of Situation in Which State Variable Returns to Boundary Backward in Time

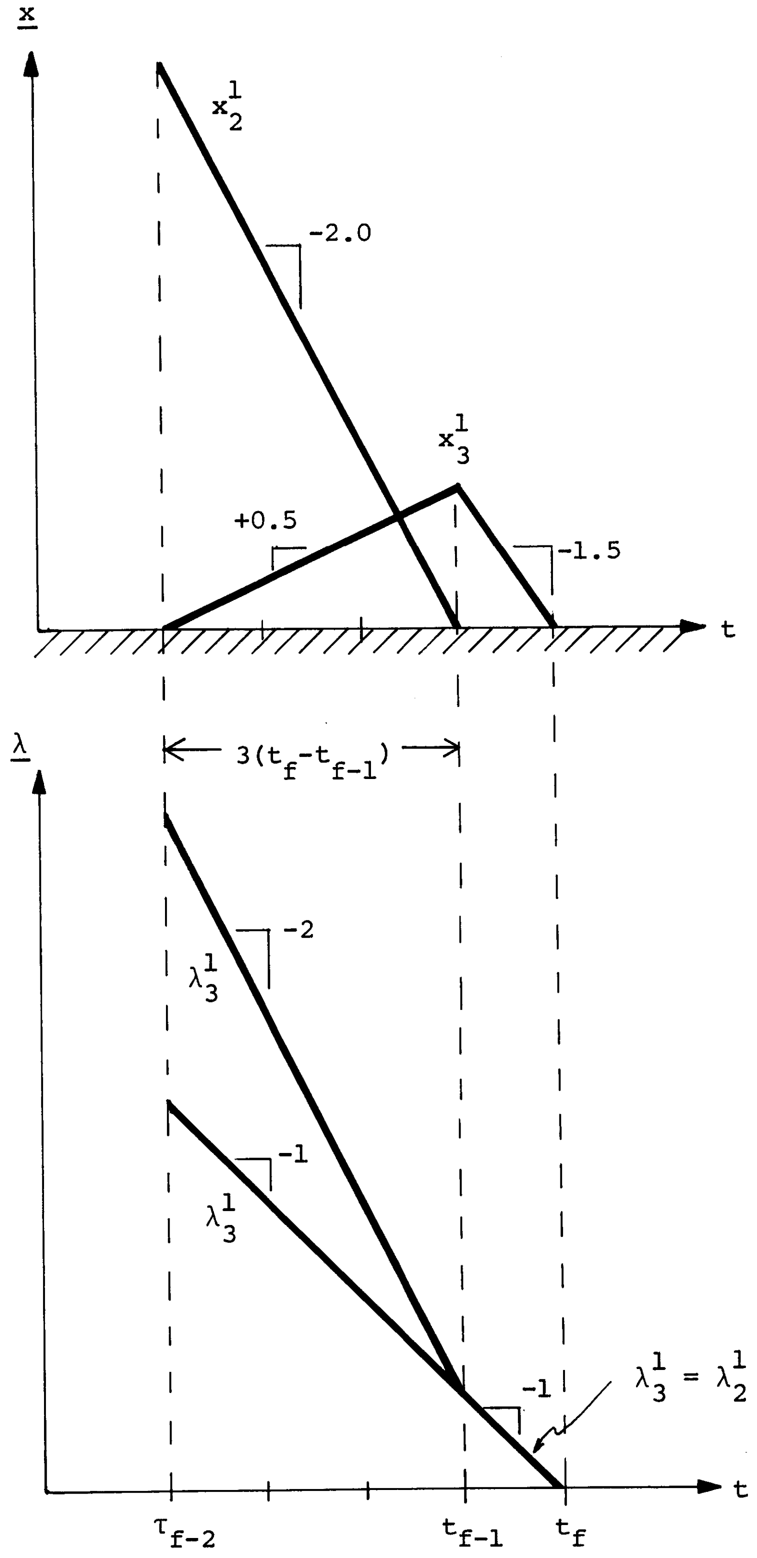
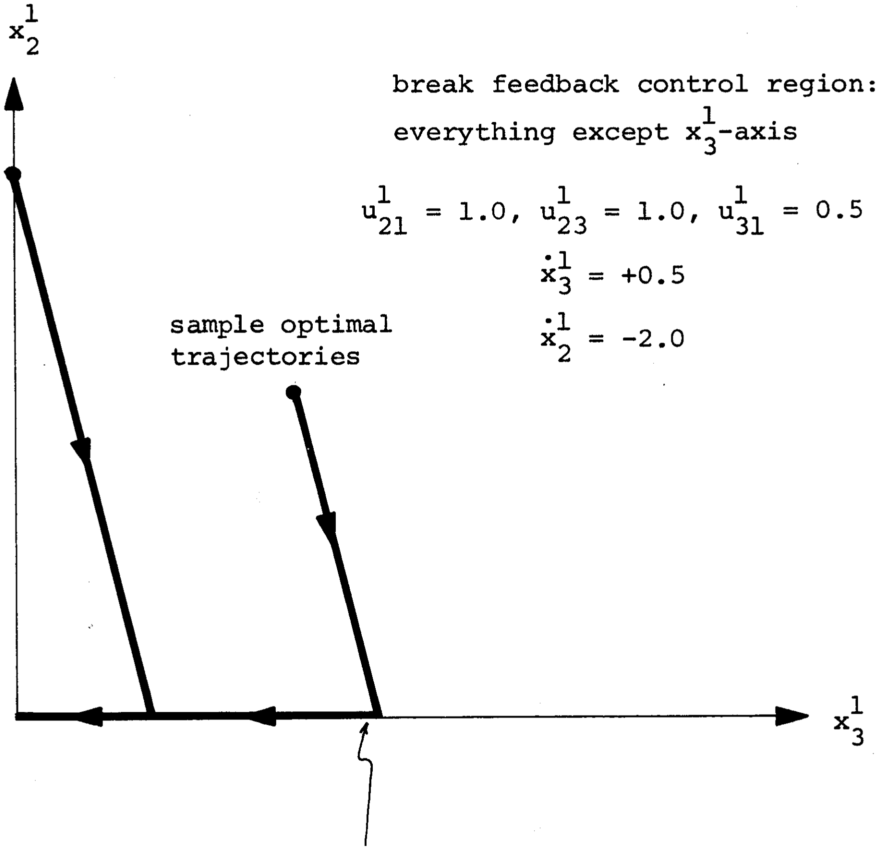


Figure 3.39 State-Costate Trajectory Pair for Example 3.7

$H_{f-2}(t_{f-1})$ to lie against the L_{f-1}-positive face of $\mathcal{Y}_{f-2}$ as depicted in Figure 3.38. Once x_2^1 leaves the boundary backward in time its costate travels with a forward slope of -2 as indicated in Figure 3.39. The Hamiltonian $H_{f-2}(\tau):z(\tau) = \lambda_2^1(\tau)y_2^1 + \lambda_3^1(\tau)y_3^1$ therefore achieves its maximum over $\mathcal{Y}_{f-2}$ at $Y_{f-2}:y_3^1 = -0.5$, $y_2^1 = 2.0$. Hence, the optimal slope of x_3^1 forward in time is now +0.5. The trajectory is pictured in Figure 3.39 where we see that x_3^1 returns to the boundary at the time τ_{f-2}, which is $3(t_f - t_{f-1})$ time units before t_{f-1}. The return of x_3^1 to the boundary marks a "dead end" of the backward trajectory since it is not possible to allow x_3^1 to re-enter the interior backward in time. This is because in order to do so it is necessary to have $\lambda_3^1(\tau_{f-2}^-) = \lambda_2^1(\tau_{f-2}^-)$ in order to have $H_{f-2}(\tau_{f-2}^-)$ lie on the appropriate face of $\mathcal{Y}_{f-2}$.

However, nothing is lost since the trajectory of Figure 3.39 may be used to construct the feedback solution for all initial conditions on x_3^1 and x_2^1. We simply treat τ_{f-2} as a break point and find that the entire space is filled up upon the construction of the break feedback region. See Figure 3.40. ● Example 3.7

We have presented a simple example for which a state variable returns to the boundary backward in time and demonstrated how the feedback space may be filled up. In this particular case, the point at which the state variable returns to the boundary represents a "dead end". However, it is not clear that this is true in general. In order



break feedback control region: everything except x_3^1-axis

$$u_{21}^1 = 1.0,\ u_{23}^1 = 1.0,\ u_{31}^1 = 0.5$$

$$\dot{x}_3^1 = +0.5$$

$$\dot{x}_2^1 = -2.0$$

non-break feedback control region: x_3^1 axis

$$u_{31}^1 = 0.5,\ u_{32}^1 = 1.0,\ u_{21}^1 = 1.0$$

$$\dot{x}_3^1 = -1.5$$

$$\dot{x}_2^1 = 0$$

Figure 3.40 Feedback Solution for Example 3.7

to fill up the space, it may be necessary (and therefore optimal) to allow state variables which have returned to the boundary backward in time to re-enter the interior. Once again, we require explicit knowledge of the $\underline{y}$-constraint figure in order to accommodate this problem as part of a computational algorithm.

Before leaving the discussion of this property, we point out that there exists some connection between the problem of non-global optimality and that of state variables returning to the boundary backward in time. We can infer this by examining the conclusions of Examples 3.4 and 3.7, which were performed for the same network topology and cost functional. In Example 3.4, we showed that it is not globally optimal for x_2^1 to strike the boundary last. This is consistent with the feedback solution of Example 3.7, which calls for x_3^1 to leave the boundary forward in time whenever x_2^1 has a positive value. It is not clear how this notion may be expanded to characterize general network problems.

3.4 Summary

In this chapter we have developed an algorithm which constructs the feedback solution to the dynamic routing problem when the inputs to the network are constant in time. The building blocks of the feedback space are convex polyhedral cones and we have proposed constructing these cones by a sequence of optimal trajectories run backward in time. These trajectories are fashioned to satisfy the necessary conditions, which are also sufficient.

Through the process of developing a rigorous algorithm we have discovered the fundamental characteristics of the feedback solution. Aside from its deceivingly neat organization into convex polyhedral conical regions, the feedback space is a complex and oftentimes confusing creature. This confirms the viewpoint commonly shared among optimal control theorists that feedback solutions are very difficult to find, especially for problems involving constraints on the state variables.

The complex algorithm for the construction of the feedback space developed in this chapter for the general network problem is difficult to implement in a computational scheme. However, it serves as a conceptual setting for proving the simplifications which result for problems involving single destination networks with unity weightings in the cost functional.

Chapter 4

FEEDBACK ROUTING WITH CONSTANT INPUTS AND UNITY WEIGHTINGS

4.1 Introduction

In Chapter 3 we presented an algorithm for the construction of the feedback solution to the general network dynamic routing problem with constant inputs. However, we found that four basic properties associated with the algorithm complicate the formulation of a computational scheme to implement the algorithm. In brief, these are:

1) Non-global optimality of certain sequences of state variables leaving the boundary backward in time
2) Non-uniqueness of leave-the-boundary costates
3) Subregions
4) Return of state variables to boundary backward in time.

In this chapter, we investigate these properties as they apply to problems involving <u>single destination networks with inputs constant in time and all unity weightings in the cost functional</u>. Throughout Chapter 3 we have promised that for this class of problems various simplifications result. Here we state and prove these simplifications using the geometrical point of view presented in Section 3.3.3.2.

The primary results are (in the order they are proved in Section 4.3):

- Theorem 4.1 The leave-the-boundary costate values are unique.
- Theorem 4.2 There is one subregion per region.
- Theorem 4.3 All solutions to the constrained optimization problem are also globally optimal.
- Theorem 4.4 There always exists an optimal control which does not call for the return of state variables to the boundary backward in time.

One additional result which even further simplifies the situation is:

- Lemma 4.6 No breaks in the optimal control between boundary junction times.

As a consequence of this last property, we need only solve the pointwise constrained linear program once between appointed boundary junction times. Moreover, the only kind of feedback regions to be constructed are those of the non-break type detailed in Section 3.3.3.3.

With these simplifications in hand, we proceed to construct the details of a computational scheme which realizes the backward constructive algorithm for this class of problems. Of particular interest is the use of parametric linear programming techniques to determine the L_p-positive face which provides the values of the leave-the-boundary costates. A computer program is developed to implement the scheme for a five-node network. However, the portion of the program which employs Chernikova's algorithm to calculate the entire set of solutions to the pointwise linear program is highly inefficient. This problem causes the calculation of the entire feedback solution to be impractical. Finally, we discuss some of the issues involved in extending the geometrical approach to general network problems.

4.2 Preliminaries

As we shall presently become involved in detailed geometrical analysis in the $\underline{y}$-space, we briefly review a few of the essential notions. First recall that the definition of the $\underline{y}$-constraint figure is:

$$\begin{aligned} &\mathcal{Y} \overset{\Delta}{=} \{\underline{y} \in \mathcal{R}^n | \underline{y} = -\underline{B}\,\underline{u} - \underline{a} \ \text{ and } \ \underline{u} \in \mathcal{U}\} \\ &\text{where } \quad \mathcal{U} \overset{\Delta}{=} \{\underline{u}: \underline{D}\,\underline{u} \le \underline{c} \ \text{ and } \ \underline{u} \ge \underline{0}\}. \end{aligned} \tag{4.1}$$

Next recall that the constrained optimization problem of Section 3.3.2.2 (Operation 1, part (a)) calls for the maximization of the Hamiltonian $H_p(\tau)$ over the restricted constraint figure $\mathcal{Y}_p$, where we define $\mathcal{Y}_p$ as follows:

$$\mathcal{Y}_p \overset{\Delta}{=} \{\underline{y} \in \mathcal{R}^{\sigma_p} | y \in \mathcal{Y} \ \text{ and } y_i^j = 0 \quad \forall\, x_i^j \in \mathcal{B}_p\}. \tag{4.2}$$

We remind the reader that $\mathcal{B}_p$ is the set of state variables on the boundary for $\tau \in [\tau_p, \tau_{p+1})$. Also, $\mathcal{I}_p$ is the set of state variables off the boundary for $\tau \in [\tau_p, \tau_{p+1})$, where card $(\mathcal{I}_p) = \sigma_p$. Now, the solution set to the constrained optimization problem on $\tau \in [\tau_p, \tau_{p+1})$ is denoted by Y_p. A member of Y_p is denoted by $\underline{y}_p$, and is referred to as a σ_p-dimensional operating point.

The following proposition regarding $\mathscr{Y}$ and $\mathscr{Y}_p$ are easy to prove, so that the details are eliminated here.

Proposition 4.1: $\mathscr{Y}$ is a closed and bounded convex polyhedron. The same is true of $\mathscr{Y}_p$.

The notions of extreme point and edge of a convex polyhedron are central in linear programming. Rigorous geometrical definitions of these notions may be found in Dantzig [1963]. Briefly, they are:

Definition 4.1: A point of a convex set is an extreme point of that set if no straight line segment of finite length completely in the set contains the point.

Definition 4.2: The edge of a convex polyhedron is the straight line segment joining two adjacent extreme points.

Now, from linear programming considerations and the details of our algorithm we know that Y_p is the convex polyhedron which is the face of tangency between the restricted Hamiltonian hyperplane $H_p(\tau)$ and the restricted constraint figure $\mathscr{Y}_p$ on $\tau \varepsilon [\tau_p, \tau_{p+1})$. Therefore, extreme points of the solution set Y_p are also extreme points of $\mathscr{Y}_p$. The following proposition is easy to prove.

Proposition 4.2: The solution set to the constrained optimization problem, Y_p, lies on the boundary of $\mathscr{Y}$. However, extreme points of $\mathscr{Y}_p$ are not necessarily extreme points of $\mathscr{Y}$.

Now, we present a proposition which provides the fundamental geometrical relationship between the constraint figures $\mathcal{Y}$ and $\mathcal{U}$. Once again, the proof is simple and is eliminated for the purpose of brevity.

Proposition 4.3: Each extreme point of $\mathcal{Y}$ is the image (through the mapping $\underline{y} = -\underline{B}\ \underline{u} - \underline{a}$) of exactly one extreme point of $\mathcal{U}$. However, extreme points of $\mathcal{U}$ may map into points that are not extremum of $\mathcal{Y}$ (such as boundary and interior points). Likewise, each edge of $\mathcal{Y}$ is the image of exactly one edge of $\mathcal{U}$. However, edges of $\mathcal{U}$ may map into points which are not edges of $\mathcal{Y}$.

Now, in order to describe the convex polyhedron $\mathcal{Z}$ we need only concern ourselves with its extreme points, or equivalently, with its edges. However, we do not have an explicit representation of $\mathcal{Y}$ in terms of linear inequalities, so that we must approach this problem somewhat indirectly. The approach shall be to talk about edges of $\mathcal{U}$, for which we do have a representation, and then to investigate the manner in which the mapping $\underline{y} = -\underline{B}\ \underline{u} - \underline{a}$ acts upon these edges. In particular, we shall be exploiting the special structure of the matrices $\underline{B}$, $\underline{C}$ and $\underline{D}$ (where $\underline{B}$, $\underline{D}$ and $\underline{C}$ appear as in (4.1)). We present here the argument as far as it applies to single destination networks. This argument is extended to cover multi-destination network models in Section 4.5, where the subsequent application of the argument is not quite as clear as it is here.

We begin by examining the nature of the edges of $\mathcal{U}$. The point of view that we take to achieve this task is the algebraic interpretation of linear programs. This interpretation is basic in the formulation of algorithms to solve linear programs, and as such can be found in any good text on the subject (such as Dantzig [1963]). The basic notions are:

(i) the canonical form of a system of linear equations,

(ii) a corresponding basic feasible solution,

and (iii) a pivot from one basic feasible solution to another, performed by exchanging a basic variable for a non-basic variable while maintaining canonical form.

Now, extreme points of a linear constraint set are characterized algebraically as corresponding to basic feasible solutions of the representative system of linear equations in canonical form. Also, a pivot from one basic feasible solution to another corresponds geometrically to moving along an edge from the extreme point represented by the first basic feasible solution to the extreme point represented by the second basic feasible solution. This is the important property which we shall exploit in our attempt to describe edges.

We now examine the set of equations which describes $\mathcal{U}$ for single destination networks. The capacity constraint over a link (i,k) has now the simple form

$$0 \leq u_{ik} \leq C_{ik} \tag{4.3}$$

since for a single destination there is exactly one control variable per link. We notice immediately from (4.3) that $\mathcal{U}$ is a rectangular hyper-parallelepiped so that the algebraic description of the edges is simple to obtain. However, we shall now characterize an edge algebraically in a rigorous fashion in order to establish the basic approach for the more complicated multi-destination situation discussed in Section 4.5.

We begin by introducing the non-negative slack variable z_{ik} so that the inequality $u_{ik} \leq C_{ik}$ becomes

$$
\begin{gathered}
u_{ik} + z_{ik} = C_{ik} \\
z_{ik} \geq 0.
\end{gathered}
\tag{4.4}
$$

The complete set of equations of type (4.4) corresponding to $\underline{D}\,\underline{u} \leq \underline{c}$ therefore has the form:

$$
r\left\{\underbrace{\begin{matrix}u&&&&&&\\&u&&&&&\\&&\ddots&&&&\\&&&u_{ik}&&&\\&&&&\ddots&&\\&&&&&u&\\&&&&&&u\end{matrix}}_{r}\right.\quad\underbrace{\begin{matrix}+z&&&&&&\\&+z&&&&&\\&&\ddots&&&&\\&&&+z_{ik}&&&\\&&&&\ddots&&\\&&&&&+z&\\&&&&&&+z\end{matrix}}_{r}\quad\begin{matrix}=C\\=C\\\vdots\\=C_{ik}\\\vdots\\=C\\=C\end{matrix}
\tag{4.5}
$$

where we have eliminated all subscripts except for ik to avoid notational havoc. Recall that r is the dimension of the control vector and therefore is the dimension of the link collection L. Now, (4.5) is immediately in canonical form with respect to any set of r variables chosen (either u's or z's) to be basic. For instance, if we consider the row corresponding to C_{ik}, then we pick either u_{ik} or z_{ik} to be basic and set its value to C_{ik} and the value of the remaining variable to zero. Note that such basic solutions always have $u_{ik} \geq 0$ ($u_{ik} = 0$ or $u_{ik} = C_{ik}$) so that the lower bound in (4.3) is never violated when talking about basic solutions to (4.5).

Hence, every extreme point of $\mathcal{U}$ is represented by a point whose ik coordinate has value of either zero or C_{ik}. In terms of the network, this means that every link is operating either at zero or full capacity. Now, in order to perform a pivot on (4.5), we simply pick one variable which is currently basic at value C_{ik} (either u_{ik} or z_{ik}) and replace it in the basis by the remaining variable (either z_{ik} or u_{ik} respectively). Hence, <u>movement along an edge of $\mathcal{U}$ is equivalent to picking a coordinate (say ik) of the current extreme point and either increasing its value to C_{ik} (if it is currently zero) or decreasing its value to zero (if it is currently at C_{ik})</u>. The total number of edges emanating from any extreme point is therefore r.

We now are in a position to discuss the nature of edges of $\mathcal{Y}$, since by Proposition 4.3 edges of $\mathcal{Y}$ correspond to edges of $\mathcal{U}$. First we distinguish between the two basic types of control variables.

<u>Definition 4.3:</u> Suppose the single destination of the network is node d. Then a control variable of the type u_{id} is called a <u>direct</u>

control. A control variable of the type u_{ik}, $k \neq d$, is called an intermediate control. See Figure 4.1.

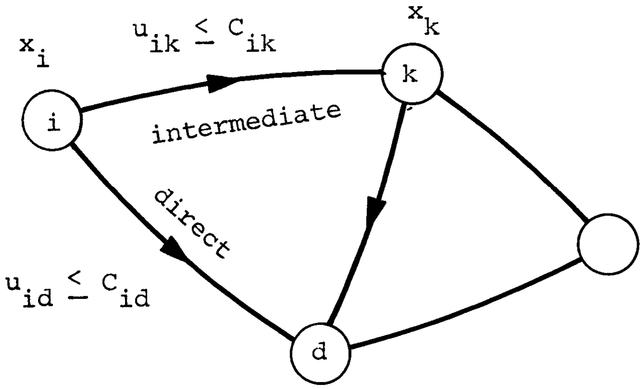


Figure 4.1 Direct and Intermediate Control Variables

Now, a direct control enters into exactly one component of the dynamic equation $\dot{\underline{x}} = \underline{B}\,\underline{u} + \underline{a}$ since there is no state variable associated with the destination node. The contribution is

$$\dot{x}_i = -u_{id} + \ldots + a_i \tag{4.6}$$

where we have eliminated the single destination superscript on the state and control variables and input. Also, an intermediate control enters into exactly two components of the dynamic equation as follows

$$\begin{aligned} \dot{x}_i &= -u_{ik} + \ldots + a_i \\ \dot{x}_k &= +u_{ik} + \ldots + a_k. \end{aligned} \tag{4.7}$$

Suppose now that the current operating point $\underline{y} = \underline{y}_p$ is an extreme point of $\mathcal{Y}$. We wish to characterize the types of edges of $\mathcal{Y}$ which may emanate from $\underline{y}_p$. From (4.6) and (4.7) we distinguish the following two types of edges of $\mathcal{Y}$:

<u>Type 1</u> - Exactly <u>one</u> coordinate of $\underline{y}$ changes from its value at y_p. This edge corresponds to an edge of $\mathcal{U}$ for which <u>a direct control and its slack</u> change places in the basis. From equations (3.162) and (4.6) we see that the two possibilities are:

$$\begin{cases} u_{id} = 0 \\ z_{id} = C_{id} \end{cases} \xrightarrow{\text{pivot}} \begin{cases} u_{id} = C_{id} \\ z_{id} = 0 \end{cases} \Rightarrow \Delta y_i = C_{id} \qquad (4.8)$$

$$\begin{cases} u_{id} = C_{id} \\ z_{id} = 0 \end{cases} \xrightarrow{\text{pivot}} \begin{cases} u_{id} = 0 \\ z_{id} = C_{id} \end{cases} \Rightarrow \Delta y_i = -C_{id} \qquad (4.9)$$

where Δy_i indicates the change in that ith coordinate which occurs when moving from one end to another of an edge. Notice that Δy_i is either plus or minus C_{id}.

<u>Type 2</u> - Exactly <u>two</u> coordinates of $\underline{y}$ change from their value at $\underline{y}_p$. This edge corresponds to an edge of for which <u>an intermediate control and its slack</u> change places in the basis. From equations (3.162) and (4.7) we see that the two possibilities are:

$$\begin{cases} u_{ik} = 0 \\ z_{ik} = c_{ik} \end{cases} \xrightarrow{\underline{\text{pivot}}} \begin{cases} u_{ik} = c_{ik} \\ z_{ik} = 0 \end{cases} \Rightarrow \begin{cases} \Delta y_i = c_{ik} \\ \Delta y_k = -c_{ik} \end{cases} \tag{4.10}$$

$$\begin{cases} u_{ik} = c_{ik} \\ z_{ik} = 0 \end{cases} \xrightarrow{\underline{\text{pivot}}} \begin{cases} u_{ik} = 0 \\ z_{ik} = c_{ik} \end{cases} \Rightarrow \begin{cases} \Delta y_i = -c_{ik} \\ \Delta y_k = c_{ik}. \end{cases} \tag{4.11}$$

where Δy_i and Δy_k indicate the changes in the ith and kth coordinate respectively which occur when moving from one end of an edge to another. Notice that Δy_i and Δy_k are both either plus or minus c_{ik}, and that $\Delta y_i = -\Delta y_k$.

<u>Notation 4.1</u>: An edge of Type 1 in which the change occurs in the coordinate y_i is represented as

$$\overset{\uparrow}{y_i}$$

and an edge of Type 2 in which the change occurs in the coordinates y_i and y_k is represented as

$$\underbrace{y_i \quad y_k}.$$

We now present a brief refresher on the geometrical point of view applied to the analysis of various properties associated with the backward construction algorithm. Assume that a current σ_p-dimensional operating point is $\underline{y}_p \in Y_p \subset \mathcal{Y}_p$ and the restricted Hamiltonian hyperplane is

$$H_p(\tau):z(\tau) = \sum_{x_i^j \varepsilon I_p} \lambda_i^j(\tau) y_i^j \qquad (4.12)$$

$$\tau \varepsilon [\tau_p, \tau_{p+1}).$$

We now choose to allow the state variables in L_p to leave the boundary backward in time at the appointed boundary junction time $t_p \varepsilon (\tau_p, \tau_{p+1})$. Note that τ_p may represent a control switch time or the boundary junction time t_{p+1} before t_p in backward time. Then the values of the leave-the-boundary costates $\lambda_i^j(t_p)$ for $x_i^j \varepsilon L_p$ are those for which the enlarged Hamiltonian hyperplane

$$H_{p-1}(t_p):z(t_p) = \sum_{x_i^j \varepsilon I_p} \lambda_i^j(t_p) y_i^j + \sum_{x_i^j \varepsilon L_p} \lambda_i^j(t_p) y_i^j \qquad (4.13)$$

contains an L_p-positive face of $\mathcal{Y}_{p-1}$.

<u>Notation 4.2:</u> Let Y_p be a current set of operating points and L_p be the set of state variables to leave the boundary backward in time at time $t_p \varepsilon (\tau_p, \tau_{p+1})$. <u>Then an</u> L_p<u>-positive face of</u> $\mathcal{Y}_{p-1}$ <u>which is contained in</u> $H_{p-1}(t_p)$ <u>for some</u> $\lambda_i^j(t_p)$, $x_i^j \varepsilon L_p$, <u>is denoted</u> $F(Y_p, t_p, L_p)$. The collection of all such faces is denoted $\mathcal{F}(Y_p, t_p, L_p)$.

This completes our discussion of the geometrical preliminaries required in the proofs of the next section. Additionally, we shall have occasion to utilize some basic terminlogy and elementary results from graph theory. The relevant definitions and a simple result are presented here.

Definition 4.4: A *graph* is a set of *nodes* $\{y_1, \ldots, y_n\}$ and a set of ordered pairs $\underset{\smile}{y_i \quad y_j}$ called *edges*. We call y_i and y_j *extremities* of the edge.

Note that we have actually referred to the coordinates of $\underline{y}$ as the nodes. This shall be our usage. Edges of the form $\underset{\smile}{y_i \quad y_j}$ actually correspond in our discussion to edges of $\mathcal{Y}$ emanating from $\underline{y}_p$.

Definition 4.5: A *subgraph* of a given graph is a graph which contains a subset of the nodes and edges of the given graph.

Definition 4.6: A *chain* is a sequence of edges such that one of the extremities of each edge belongs to the preceding edge in the sequence and the other extremity to the following edge.

Definition 4.7: A graph is *connected* if two arbitrary distinct nodes are always connected by a chain.

Definition 4.9: A connected graph without loops is a *tree*.

Proposition 4.4: A tree of s nodes contains s-1 edges.

4.3 Special Properties of the Algorithm

We begin with the central theorem of this section. The proof of this theorem employs the geometrical point of view, and contains the basic notions which are utilized in subsequent proofs.

Theorem 4.1

For problems involving single destination networks, the values of the leave-the-boundary costates are unique at a given boundary junction time t_p.

Proof: We emphasize that in this proof t_p is a fixed potential boundary junction time, and begin by considering the case in which we wish to allow all the states remaining on the boundary to leave the boundary at t_p, that is, $L_p = \mathcal{B}_p$. It will be easy to generalize the result to cases for which $L_p \subset \mathcal{B}_p$ and $L_p \neq \mathcal{B}_p$.

We now divide the proof into two principal sections:

(a) Y_p contains an extreme point of $\mathcal{Y}$

(b) Y_p does not contain an extreme point of $\mathcal{Y}$.

(a) Y_p contains an extreme point of $\mathcal{Y}$.

The proof will proceed by contradiction. Suppose the values of the leave-the-boundary costates are not unique. Then according to the geometrical interpretation $\mathcal{F}(Y_p, t_p, \mathcal{B}_p)$ contains at least two distinct members. Call these $F'(Y_p, t_p, \mathcal{B}_p)$ and $F''(Y_p, t_p, \mathcal{B}_p)$. Each of these is a face of $\mathcal{Y}$ since $L_p = \mathcal{B}_p$ and $\mathcal{Y}_{p-1} = \mathcal{Y}$. Now, the enlarged Hamiltonian hyperplane is actually the n-1 dimensional global Hamiltonian hyperplane, so that we refer to it as $H(t_p)$. In accordance with the assumption that $F'(Y_p, t_p, \mathcal{B}_p)$ and $F''(Y_p, t_p, \mathcal{B}_p)$ are distinct, we must have that each is contained in a different global

Hamiltonian hyperplane. We denote these global Hamiltonian hyperplanes by $H'(t_p)$ and $H''(t_p)$ respectively and express them as

$$H'(t_p): z'(t_p) = \sum_{x_i \in \mathcal{I}_p} \lambda_i(t_p) y_i + \sum_{x_i \in \mathcal{B}_p} \lambda_i'(t_p) y_i \tag{4.14}$$

$$H''(t_p): z''(t_p) = \sum_{x_i \in \mathcal{I}_p} \lambda_i(t_p) y_i + \sum_{x_i \in \mathcal{B}_p} \lambda_i''(t_p) y_i \tag{4.15}$$

where we have eliminated the destination superscripting since we are considering single destination networks. Now, $Y_p \subseteq H_p(t_p)$ and from Theorem 3.5, part (a), we know that $H_p(t_p) \subset H'(t_p)$ and $H_p(t_p) \subset H''(t_p)$. Therefore, $Y_p \subset H'(t_p)$ and $Y_p \subset H''(t_p)$. The geometry associated with our assumption is depicted in Figure 4.2.

Consider now any operating point $\underline{y}_p \in Y_p$. Then by the above considerations we know that $\underline{y}_p \in H'(t_p)$ and $\underline{y}_p \in H''(t_p)$; that is, $\underline{y}_p$ satisfies both (4.14) and (4.15). Substituting $\underline{y}_p$ into both of these expressions we obtain:

$$z'(t_p) = z''(t_p). \tag{4.16}$$

We now make the further assumptions that $F'(Y_p, t_p, \mathcal{B}_p)$ and $F''(Y_p, t_p, \mathcal{B}_p)$ are adjacent faces of $\mathcal{Y}$. We may do this without any loss of generality because if there is more than one member of $\mathcal{F}(Y_p, t_p, \mathcal{B}_p)$ we may always pick two adjacent members. Now, according to the definition of $\mathcal{L}_p$-positive face, there must exist some point $\underline{y}^+$ such that

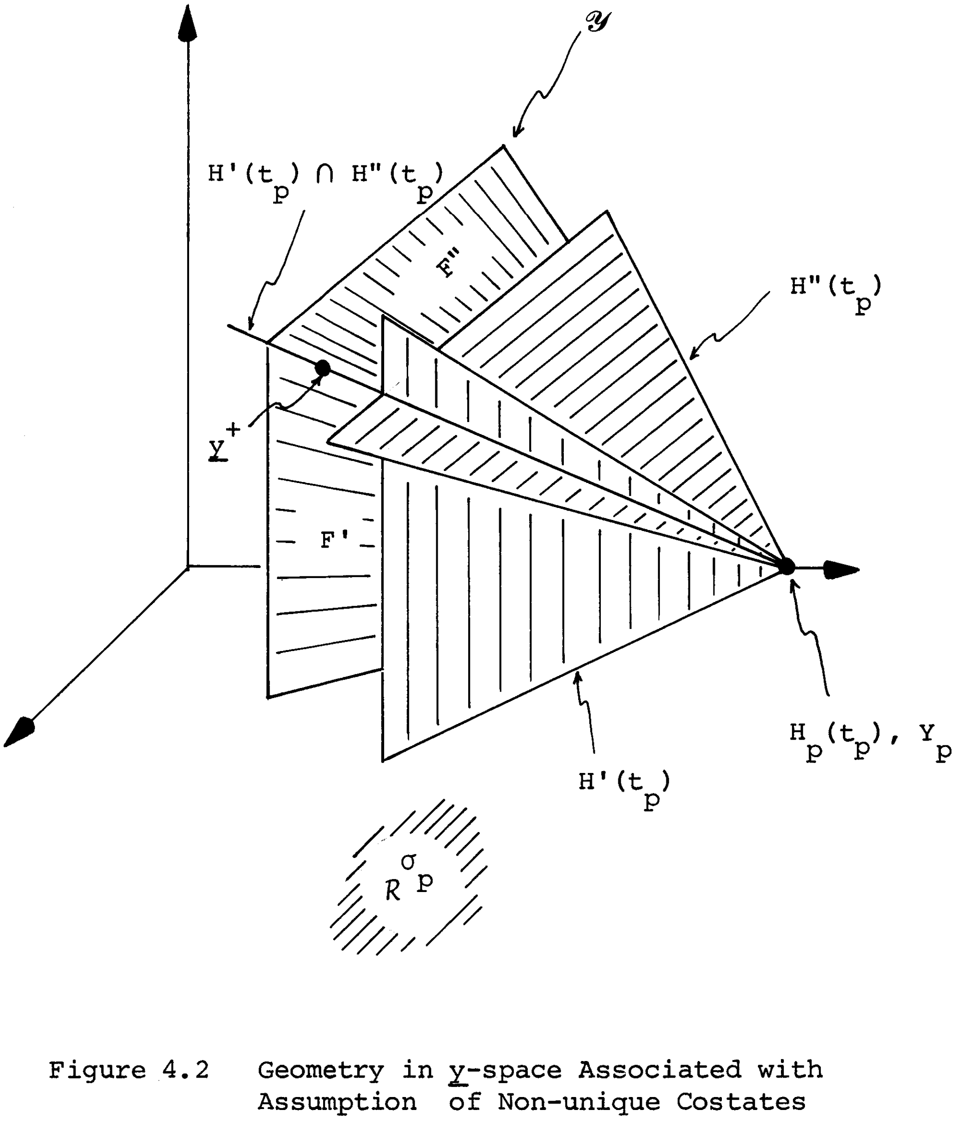


Figure 4.2 Geometry in $\underline{y}$-space Associated with Assumption of Non-unique Costates

$$\underline{y}^{+} \varepsilon H'(t_p) \cap H''(t_p) \tag{4.17}$$

where $\underline{y}^{+}$ is in the positive orthant of $\mathcal{R}^{\rho_p}$; that is,

$$y_i^{+} > 0 \quad \forall\, x_i \varepsilon \mathcal{L}_p = \mathcal{B}_p. \tag{4.18}$$

By subtracting (4.15) from (4.14) and taking into account (4.16), we conclude that any $\underline{y} \varepsilon H'(t_p) \cap H''(t_p)$ satisfies

$$\sum_{x_i \varepsilon \mathcal{B}_p} (\lambda_i'(t_p) - \lambda_i''(t_p))y_i = 0. \tag{4.19}$$

Therefore, $\underline{y}^{+}$ must satisfy

$$\sum_{x_i \varepsilon \mathcal{B}_p} (\lambda_i'(t_p) - \lambda_i''(t_p))y_i^{+} = 0. \tag{4.20}$$

Our assumption that $H'(t_p)$ and $H''(t_p)$ are distinct hyperplanes implies that $\lambda_j'(t_p) - \lambda_j''(t_p) \neq 0$ for at least one $x_j \varepsilon \mathcal{B}_p$. This together with (4.18) implies

$$(\lambda_j'(t_p) - \lambda_j''(t_p))y_j^{+} \neq 0 \quad \text{for some } x_j \varepsilon \mathcal{B}_p. \tag{4.21}$$

But (4.20) and (4.21) imply that there must exist some $x_k \varepsilon \mathcal{B}_p$ such that $\lambda_k'(t_p) - \lambda_k''(t_p) \neq 0$ where $\lambda_k'(t_p) - \lambda_k''(t_p)$ is of opposite sign from $\lambda_j'(t_p) - \lambda_j''(t_p)$. We now summarize the preceding observations:

> Positive Solution Condition. In order for the assumption of non-unique costates to be true, it is necessary that $\lambda_j'(t_p) - \lambda_j''(t_p) \neq 0$ and $\lambda_k'(t_p) - \lambda_k''(t_p) \neq 0$ for some $x_j \varepsilon \mathcal{B}_p$

and $x_k \in \mathcal{B}_p$. Moreover, it is necessary that $\lambda_j'(t_p) - \lambda_j''(t_p)$ is of opposite sign from $\lambda_k'(t_p) - \lambda_k''(t_p)$.

Let us now consider the region $F'(Y_p, t_p, \mathcal{B}_p) \cap F''(Y_p, t_p, \mathcal{B}_p)$. As we have assumed that the faces $F'(Y_p, t_p, \mathcal{B}_p)$ and $F''(Y_p, t_p, \mathcal{B}_p)$ are adjacent, their intersection must be non-empty. Also, since both $F'(Y_p, t_p, \mathcal{B}_p)$ and $F''(Y_p, t_p, \mathcal{B}_p)$ are n-1 dimensional faces of $\mathcal{Y}$, the intersection $F'(Y_p, t_p, \mathcal{B}_p) \cap F''(Y_p, t_p, \mathcal{B}_p)$ is an n-2 dimensional face of $\mathcal{Y}$; that is, it is an n-2 dimensional convex polyhedron.

Recall now that in this part of the proof we are assuming that Y_p contains an extreme point of $\mathcal{Y}$. Denote such an extreme point by $\underline{y}_e \in Y_p$. By Definition 3.9 of an $\mathcal{L}_p$-positive face, we have

$$\underline{y}_e \in F'(Y_p, t_p, \mathcal{B}_p) \tag{4.22}$$

and

$$\underline{y}_e \in F''(Y_p, t_p, \mathcal{B}_p). \tag{4.23}$$

Therefore, $\underline{y}_e \in F'(Y_p, t_p, \mathcal{B}_p) \cap F''(Y_p, t_p, \mathcal{B}_p)$. Since $\underline{y}_e$ is an extreme point of $\mathcal{Y}$, it is also an extreme point of the n-2 dimensional convex polyhedral face of $\mathcal{Y}$, $F'(Y_p, t_p, \mathcal{B}_p) \cap F''(Y_p, t_p, \mathcal{B}_p)$. Therefore, there is a minimum of n-2 independent edges of $F'(Y_p, t_p, \mathcal{B}_p) \cap F''(Y_p, t_p, \mathcal{B}_p)$ which emanate from $\underline{y}_e$. These are also edges of $\mathcal{Y}$. Now, we have

$$F'(Y_p, t_p, \mathcal{B}_p) \subset H'(t_p)$$

and (4.24)

$$F''(Y_p, t_p, \mathcal{B}_p) \subset H''(t_p)$$

in accordance with the geometrical interpretation. This implies that

$$F'(Y_p, t_p, \mathcal{B}_p) \cap F''(Y_p, t_p, \mathcal{B}_p) \subset H'(t_p) \cap H''(t_p) \qquad (4.25)$$

Hence, in order for our assumption of non-unique costates to be true the following condition must also hold:

> Edge Condition. If $\underline{y}_e \varepsilon Y_p$ is an extreme point of $\mathcal{Y}$ then there exists a minimum of n-2 linearly independent edges of $\mathcal{Y}$ emanating from $\underline{y}_e$ which are also contained in $H'(t_p) \cap H''(t_p)$.

We now begin the task of providing a contradiction to our assumption by demonstrating that the positive solution condition and the edge condition cannot hold simultaneously. In brief, we shall proceed by enumerating all possible types of edges which emanate from $\underline{y}_e$ and that satisfy the requirements of the positive solution condition and the edge condition. Our conclusion shall be that the maximum number of such edges in n-3, which is one short of the number required by the edge condition.

Now, the two basic types of edges of $\mathcal{Y}$ were described in Section 4.2 under the names "Type 1" and "Type 2". In brief, Type 1 edges involve a change in exactly one coordinate of $\underline{y}$ while Type 2 edges involve changes in exactly two coordinates of $\underline{y}$. We now wish to be more specific with regard to the type of state variables affected by the pivots, that is, whether they are members of $\mathcal{I}_p$ or $\mathcal{B}_p$. Our notation shall be that a pivot takes us from the current operating (extreme) point $\underline{y}_e$ to an adjacent extreme point $\underline{y}_e'$. We denote the i^{th} coordinate of $\underline{y}_e$ by y_i^e and the i^{th} coordinate of $\underline{y}_e'$ by y_i'.

(i) <u>Type 1 edge involving $x_j \varepsilon \mathcal{I}_p$</u>

According to (4.8)-(4.9) we have

$$\begin{cases} y_i' = y_i^e & \forall \ i \neq j \\ y_j' = y_j^e + \Delta y_j & \text{where } |\Delta y_j| = C_{jd} > 0. \end{cases} \tag{4.26}$$

The edge condition requires $\underline{y}' \varepsilon H'(t_p) \cap H''(t_p)$. Therefore, the point $\underline{y}'$ must satisfy the equations for $H'(t_p)$ and $H''(t_p)$. Substituting $\underline{y}'$ into (4.14) and (4.15) and using (4.16) we obtain

$$\lambda_j(t_p)(\Delta y_j) + \sum_{x_i \varepsilon \mathcal{B}_p} \lambda_i'(t_p) y_i' = 0 \tag{4.27}$$

$$\lambda_j(t_p)(\Delta y_j) + \sum_{x_i \varepsilon \mathcal{B}_p} \lambda_i''(t_p) y_i' = 0. \tag{4.28}$$

Taking note of the fact that $y_i' = 0$ for $x_i \varepsilon \mathcal{B}_p$ (since $y_i^e = 0$ for $x_i \varepsilon \mathcal{B}_p$) we obtain

$$\lambda_j(t_p)(\Delta y_j) = 0. \tag{4.29}$$

The necessary conditions of optimality (3.32) call for $\lambda_j(t_p) > 0$ for $x_j \in \mathcal{I}_p$. But since (4.26) specifies $|\Delta y_j| > 0$, equation (4.29) presents a contradiction. Therefore, this type of edge cannot satisfy the edge condition and therefore is not a candidate for our n-2 required edges.

(ii) Type 1 edge involving $x_j \in \mathcal{B}_p$

In this case we have

$$\begin{cases} y_i' = y_i^e & \forall\, i \neq j \\ y_j' = \Delta y_j & \text{where } |\Delta y_j| = c_{jd} > 0 \end{cases} \tag{4.30}$$

By (4.8)-(4.9) and the fact that $y_j^e = 0 \quad \forall\, x_j \in \mathcal{B}_p$. Once again, the point $\underline{y}'$ must satisfy the equations for both $H'(t_p)$ and $H''(t_p)$. Substituting $\underline{y}'$ into (4.14) and (4.15) and simplifying obtain

$$\lambda_j'(t_p)(\Delta y_j) = \lambda_j''(t_p)(\Delta y_j) = 0. \tag{4.31}$$

But $|\Delta y_j| > 0$ and (4.31) imply

$$\lambda_j'(t_p) = \lambda_j''(t_p).$$

Therefore, edges of this type are potential edges of $\mathcal{Y}$ with the stipulation that

$$\lambda_j'(t_p) - \lambda_j''(t_p) = 0. \tag{4.32}$$

(iii) Type 2 edge involving $x_j \varepsilon \mathcal{B}_p$, $x_k \varepsilon \mathcal{B}_p$

In this case (4.1) and (4.11) give

$$\begin{cases} y_i' = y_i^e & \forall \ i \neq j,k \\ y_j' = \Delta y_j & \\ y_k' = -\Delta y_j & \text{where } |\Delta y_j| = c_{jk} > 0. \end{cases} \tag{4.33}$$

As before, we substitute (4.33) into (4.14) and (4.15) and simplify to obtain

$$\begin{aligned} \lambda_j'(t_p) &= \lambda_k'(t_p) \\ \lambda_j''(t_p) &= \lambda_k''(t_p). \end{aligned} \tag{4.34}$$

(iv) Type 2 edge involving $x_j \varepsilon \mathcal{B}_p$, $x_k \varepsilon \mathcal{I}_p$

We have by (4.10) and (4.11) that

$$\begin{cases} y_i' = y_i^e & \forall \ i \neq j,k \\ y_j' = \Delta y_j & \\ y_k' = y_k^e - \Delta y_j & \text{where } |\Delta y_j| = c_{jk} > 0 \end{cases} \tag{4.35}$$

Substituting (4.35) into (4.14) and (4.15) and simplifying obtain

$$\lambda_j'(t_p) = \lambda_j''(t_p) = 0. \tag{4.36}$$

(v) Type 2 edge involving $x_j \varepsilon \mathcal{I}_p$, $x_k \varepsilon \mathcal{I}_p$

From (4.10) and (4.11) we have

$$\begin{cases} y_i' = y_i^e & \forall\ i \neq j,k \\ y_j' = y_j^e + \Delta y_j & \\ y_k' = y_k^e - \Delta y_j & \text{where } |\Delta y_j| = c_{jk} > 0. \end{cases} \tag{4.37}$$

Substituting (4.37) into (4.14) and (4.15) and simplifying obtain

$$\lambda_j(t_p) = \lambda_k(t_p). \tag{4.38}$$

We shall refer to the admissible edges represented by (ii)-(iv) above as "edges of type (ii)-(iv)" respectively.

We have now considered all possible types of edges emanating from $\underline{y}_e$ and found that the edge condition has implied certain relationships among the costates involved. Now, in our attempt to arrive at our contradiction we shall not attempt to explicitly enumerate all edges emanating from $\underline{y}_e$ for a specific constraint figure $\mathscr{Y}$. Instead, we shall consider all possible edges of types (ii)-(iv) above which may exist and show that there are no more than n-3 independent edges which also satisfy the positive solution condition. In order to facilitate the discussion concerning an arbitrary set of edges, we introduce the following graphical representation.

Definition 4.10: Consider the components of $\underline{y}$ $(y_1, \ldots, y_n)$ to be the nodes of a graph and the arrows and arcs of notation 4.1 to be edges of a graph. Then an edge graph is the collection of all such nodes and edges which represent a set of edges of $\mathscr{Y}$ emanating from $\underline{y}_e$.

Note that edges of Type (ii) ($\overset{\uparrow}{y}_i$ where $x_i \varepsilon \mathcal{B}_p$) are special in that they only involve one extremity node. We now divide the nodes of an edge graph into two mutually exclusive categories: those which correspond to state variables which are members of the boundary set $\mathcal{B}_p$ and those which correspond to state variables which are members of the interior set $\mathcal{I}_p$. As a shorthand notation we shall say that if $x_i \varepsilon \mathcal{B}_p$ then $y_i \varepsilon \mathcal{B}_p$, where $\mathcal{B}_p$ is now taken as a collection of nodes of the edges graph. Likewise, if $x_i \varepsilon \mathcal{I}_p$ we say that $y_i \varepsilon \mathcal{I}_p$. The use of the notations $\mathcal{B}_p$ and $\mathcal{I}_p$ should be clear from the context. A typical edge graph categorized in this fashion is presented in Figure 4.3, where we have represented components of $\underline{y}$ simply as y's.

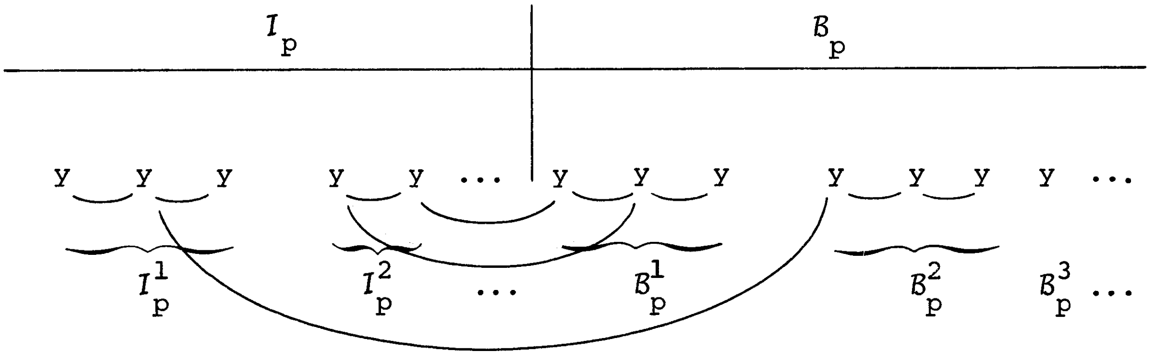


Figure 4.3 Typical Edge Graph

Using the notation we have established we now reiterate the four possible types of edges which may exist and the costate conditions which they imply:

$$\text{(ii)} \qquad \begin{matrix}\mathcal{B}_p\\ \uparrow \\ y_j\end{matrix} \longrightarrow \lambda'_j(t_p)-\lambda''_j(y_p) = 0 \tag{4.39}$$

$$\text{(iii)} \qquad \underparen{\overbrace{y_j \quad y_k}^{\mathcal{B}_p}} \longrightarrow \lambda'_j(t_p)-\lambda''_j(t_p) = \lambda'_k(t_p)-\lambda''_k(t_p) \tag{4.40}$$

$$\text{(iv)} \qquad \underparen{\overset{\mathcal{B}_p}{y_j} \quad \overset{\mathcal{I}_p}{y_k}} \longrightarrow \lambda'_j(t_p)-\lambda''_j(t_p) = 0 \tag{4.41}$$

$$\text{(v)} \qquad \underparen{\overbrace{y_j \quad y_k}^{\mathcal{I}_p}} \longrightarrow \lambda_j(t_p) = \lambda_k(t_p) \tag{4.42}$$

Definition 4.11: For a given edge graph, a $\mathcal{B}_p$-subgraph (denoted $\mathcal{B}_p^{\alpha}$, where the integer valued α ranges over the number of $\mathcal{B}_p$-subgraphs) consists of a set of y's in $\mathcal{B}_p$ and a set of edges for which the set of y's is disconnected from all other members of $\mathcal{B}_p$.

Definition 4.12: For a given edge graph, an $\mathcal{I}_p$-subgraph (denoted $\mathcal{I}_p^{\beta}$, where the integer β ranges over the number of $\mathcal{I}_p$-subgraphs) consists of a set of y's in $\mathcal{I}_p$ and a set of edges for which the set of y's is disconnected from all other members of $\mathcal{I}_p$.

Examples of $\mathcal{B}_p$- and $\mathcal{I}_p$-subgraphs are presented in Figure 4.3.

Definition 4.13: The subgraphs $\mathcal{B}_p^\alpha$ and $\mathcal{I}_p^\beta$ are said to be connected if some member of $\mathcal{B}_p^\alpha$ is connected to some member of $\mathcal{I}_p^\beta$.

For example, in Figure 4.3, $\mathcal{B}_p^1$ is connected to $\mathcal{I}_p^2$.

Definition 4.14: The $\mathcal{B}_p$-subgraph $\mathcal{B}_p^\alpha$ is said to be independent if it is not connected to any $\mathcal{I}_p$-subgraph.

In Figure 4.3, the $\mathcal{B}_p$-subgraph $\mathcal{B}_p^3$ is independent.

We now assume that all edges of any subgraph under discussion satisfy the edge condition. Therefore, the relations (4.39)-(4.42) are in force. We now apply these relations to make the following observations concerning members of $\mathcal{B}_p$- and $\mathcal{I}_p$- subgraphs:

- If $y_j \in \mathcal{B}_p^\alpha$ and $y_k \in \mathcal{B}_p^\alpha$ then

$$\lambda_j'(t_p) - \lambda_j''(t_p) = \lambda_k'(t_p) - \lambda_k''(t_p). \tag{4.43}$$

That is, for any members of the same $\mathcal{B}_p$-subgraph the values of the corresponding costate differences are the same.

- If $y_j \in \mathcal{I}_p^\beta$ and $y_k \in \mathcal{I}_p^\beta$ then

$$\lambda_j(t_p) = \lambda_k(t_p). \tag{4.44}$$

That is, for any members of the same $\mathcal{I}_p$-subgraph the values of the corresponding costates are the same.

- If $y_j \varepsilon \mathcal{B}_p^\beta$ and $y_k \varepsilon \mathcal{I}_p^\alpha$ <u>and</u> $\mathcal{I}_p^\beta$ and $\mathcal{B}_p^\alpha$ are connected then

$$\lambda_j'(t_p) - \lambda_j''(t_p) = 0. \qquad (4.45)$$

That is, each member of a $\mathcal{B}_p$-subgraph which is connected to any $\mathcal{I}_p$-subgraph has its corresponding costate difference equal to zero.

- If $y_j \varepsilon \mathcal{B}_p^\alpha$ and y_j has the type (ii) edge $\overset{\uparrow}{y_i}$, then

$$\lambda_i'(t_p) - \lambda_i''(t_p) = 0 \qquad \forall\, y_i \varepsilon \mathcal{B}_p^\alpha. \qquad (4.46)$$

That is, each member of a $\mathcal{B}_p$-subgraph which contains a type (ii) edge among any of its members has its corresponding costate difference equal to zero.

<u>Lemma 4.1:</u> In order that the positive solution condition be satisfied by the edges represented in a particular edge graph, it is necessary that there are at least two independent $\mathcal{B}_p$-subgraphs in the edge graph, each of which does not contain a type (ii) edge.

<u>Proof of Lemma 4.1:</u> The positive solution condition requires that we have an $x_j \varepsilon \mathcal{B}_p$ and an $x_k \varepsilon \mathcal{B}_p$ such that $\lambda_j'(t_p) - \lambda_j''(t_p) \neq 0$ and $\lambda_k'(t_p) - \lambda_k''(t_p) \neq 0$, and these costate differences must be of the opposite sign. Equation (4.43) tells us that the state variables which

satisfy this requirements must not correspond to coordinates of $\underline{y}$ which are members of the same $\mathcal{B}_p$-subgraph. Therefore, x_k and x_j must correspond to members of separate $\mathcal{B}_p$-subgraphs, say $\mathcal{B}_p^j$ and $\mathcal{B}_p^k$. Now, by (4.45) neither $\mathcal{B}_p^j$ or $\mathcal{B}_p^k$ can be connected to any $\mathcal{I}_p$-subgraph, and hence each is independent. Furthermore, (4.46) implies that neither $\mathcal{B}_p^j$ or $\mathcal{B}_p^k$ contain a type (ii) edge. ■ Lemma 4.1

We now interpret the property of linear dependence of edges into our graph theoretic approach.

Lemma 4.2: Any subgraph of an edge graph which contains a loop must represent a linearly dependent edge of $\mathcal{Y}$.

Proof of Lemma 4.2: It is easily seen that a two node subgraph with a loop represents two linearly dependent edges of $\mathcal{Y}$. Now, consider the three node subgraph with a loop in Figure 4.4. The edges are identified by circled integers. Suppose this subgraph represents three edges of $\mathcal{Y}$ emanating from $\underline{y}_e$. Let us now denote by $\Delta y_i^{\textcircled{j}}$ the change in the coordinate y_i which occurs when moving along the edge ⓙ from $\underline{y}_e$ to an adjacent extreme point of $\mathcal{Y}$. From (4.10) and (4.11) we then have

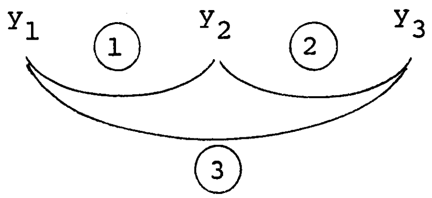


Figure 4.4 Three Node Subgraph with Loop

the following relationships:

$$\Delta y_1^{\textcircled{1}} = -\Delta y_2^{\textcircled{1}}$$

$$\Delta y_2^{\textcircled{2}} = -\Delta y_3^{\textcircled{2}}$$

$$\Delta y_1^{\textcircled{3}} = -\Delta y_3^{\textcircled{3}} \quad .$$

Let us now consider the three vectors in $\underline{y}$-space corresponding to edges ① , ② and ③. When we perform the determinant test for linear independence of these vectors, the above relationships yield:

$$\begin{vmatrix} \Delta y_1^{\textcircled{1}} & \Delta y_2^{\textcircled{1}} & 0 \\ 0 & \Delta y_2^{\textcircled{2}} & \Delta y_3^{\textcircled{2}} \\ \Delta y_1^{\textcircled{3}} & 0 & \Delta y_3^{\textcircled{3}} \end{vmatrix} = 0.$$

Therefore, edge ③ is linearly dependent upon edges ① and ② .

Now, consider a loop in the a subgraph of arbitrary size. See Figure 4.5.

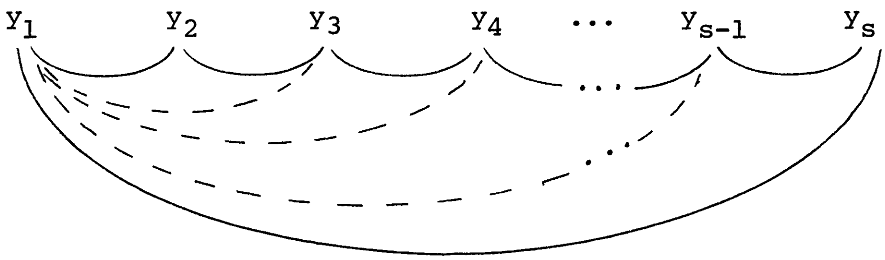


Figure 4.5 Loop in Subgraph of Arbitrary Size

From the previous argument concerning a three node loop we conclude that

$\underbrace{y_1 \quad y_3}$ is a linear combination of $\underbrace{y_1 \quad y_2}$ and $\underbrace{y_2 \quad y_3}$,

$\underbrace{y_1 \quad y_4}$ is a linear combination of $\underbrace{y_1 \quad y_3}$ and $\underbrace{y_3 \quad y_4}$,

$\vdots$

$\underbrace{y_1 \quad y_s}$ is a linear combination of $\underbrace{y_1 \quad y_{s-1}}$ and $\underbrace{y_{s-1} \quad y_s}$.

Therefore, the edge of $\mathscr{Y}$ represented by $\underbrace{y_1 \quad y_s}$ is a linear combination of those represented by $\underbrace{y_1 \quad y_2}$, $\underbrace{y_2 \quad y_3}$, ..., $\underbrace{y_{s-1} \quad y_s}$.

Lemma 4.2 ■

In counting linearly independent edges of an edge graph we shall be considering the following special subgraphs.

Definition 4.15: A connected subgraph of an edge graph is either a $\mathcal{B}_p$-subgraph, an $\mathcal{I}_p$-subgraph, or the union of a $\mathcal{B}_p$-subgraph and an $\mathcal{I}_p$-subgraph which are connected.

Lemma 4.3

(a) Any connected subgraph consisting of s nodes has s-1 linearly independent edges, not including any edges of type (ii).

(b) No more than one edge of type (ii) can be added to a connected subgraph representing linearly independent edges of $\mathscr{Y}$ and have the new set of edges represented remain linearly independent.

Proof of Lemma 4.3:

(a) If the connected subgraph contains a loop, then by Lemma 4.2 it contains at least one linearly dependent edge. Therefore, discard one edge per loop. The remaining subgraph is loop free and connected. By Definition 4.9 it is then a tree. Also, by Proposition 4.4 it contains s-1 edges. It is not difficult to see that all of the edges of a tree are linearly independent.

(b) By part (a) a connected subgraph with s nodes contains s-1 linearly independent edges. But we may have not more than s independent edges in a subspace of dimension s. Therefore, only one type (ii) edge may be added. ■Lemma 4.3

We now proceed to determine the maximum number of independent edges which satisfy Lemmas 4.1 and 4.3 (which embody the positive solution condition and the edge condition). The breakdown of $\mathcal{I}_p$ and $\mathcal{B}_p$ is as follows:

According to Lemma 4.1, there must be at least two independent $\mathcal{B}_p$-subgraphs in $\mathcal{B}_p$ each of which does not contain an edge of type (ii). Call these $\mathcal{B}_p$-subgraphs $\mathcal{B}_p'$ and $\mathcal{B}_p''$. Then by Lemma 4.3, the number of linearly independent edges of $\mathcal{Y}$ represented by these $\mathcal{B}_p$-subgraphs are cardinality $(\mathcal{B}_p')$-1 and cardinality $(\mathcal{B}_p'')$-1 respectively.

In addition, we may have some $\mathcal{B}_p$-subgraphs, each of which contains a type (ii) edge. By Lemma 4.3 each of these represents the maximum possible number of linearly independent edges of $\mathcal{Y}$, which is exactly the number of nodes in the subgraph. Denote the collection of all such subgraphs by $\mathcal{B}_p'''$. Then the number of linearly independent edges representedy by $\mathcal{B}_p'''$ is cardinality $(\mathcal{B}_p''')$.

Finally, we must consider all of the $\mathcal{I}_p$-subgraphs and the remaining $\mathcal{B}_p$-subgraphs. From Theorem 4.3 we conclude that these subgraphs will represent the maximum possible number of linearly independent edges of $\mathcal{Y}$ if each $\mathcal{B}_p$-subgraph is connected to an $\mathcal{I}_p$-subgraph by an edge of type (iv) to form a connected subgraph. Denote the union of all these subgraphs by $\mathcal{G}$. Then the number of linearly independent edges represented by $\mathcal{G}$ is cardinality $(\mathcal{G})-1$.

The above breakdown is depicted in Figure 4.6.

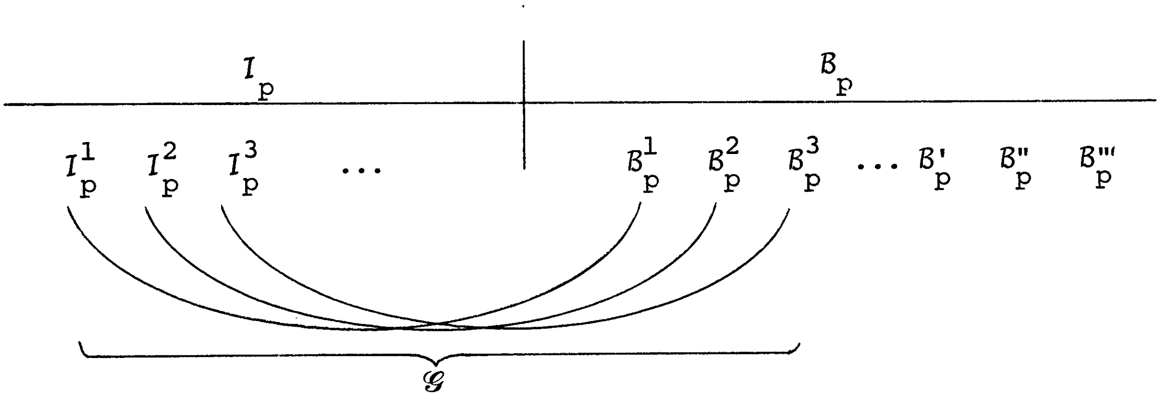

$\mathcal{I}_p$
$\mathcal{B}_p$
$\mathcal{I}_p^1$ $\mathcal{I}_p^2$ $\mathcal{I}_p^3$ ...
$\mathcal{B}_p^1$ $\mathcal{B}_p^2$ $\mathcal{B}_p^3$ ... $\mathcal{B}_p'$ $\mathcal{B}_p''$ $\mathcal{B}_p'''$
$\mathcal{G}$

Figure 4.6 Edge Graph Which Satisfies Positive Solution Condition and Edge Condition

We now tally up the total maximum number of linear independent edges of $\mathcal{Y}$ represented by the above:

$$\begin{array}{lll} \text{maximum no. of linearly independent edges} & = & \text{cardinality } (\mathcal{B}'_p + \mathcal{B}''_p + \mathcal{B}'''_p + \mathcal{G}) - 3 \\ & = & \text{cardinality } (\mathcal{B}_p \cup \mathcal{I}_p) - 3 \\ & = & n-3. \end{array} \qquad (4.47)$$

But the edge condition requires that we produce n-2 such edges. Hence, our original assumption that $\mathcal{F}(Y_p, t_p, \mathcal{B}_p)$ contains more than one element must be false if Y_p contains an extreme point of $\mathcal{Y}$.

(b) <u>Y_p does not contain an extreme point of $\mathcal{Y}$</u>

If Y_p does not contain an extreme point of $\mathcal{Y}$, then it must lie on some face of $\mathcal{Y}$ whose dimension is greater than the dimension of $H_p(t_p)$. This is so because if the face on which Y_p lies is of dimension less than or equal to that of $H_p(t_p)$, then Y_p must be that face, and hence contain an extreme point of $\mathcal{Y}$. Let the g-dimensional face of $\mathcal{Y}$ on which Y_p lies be denoted F_g, where $\sigma_p < g < n-1$. See Figure 4.7.

The geometry of linear programs tells us that any globally maximizing Hamiltonian hyperplane $H(t_p)$ which contains Y_p must also contain the face on which it lies, namely F_g. Hence $H(t_p)$ must contain the g-1 dimensional linear variety in which F_g lies, call it $H_g(t_p)$. Now $H_g(t_p)$ must contain $H_p(t_p)$, so that it has the form:

$$H_g(t_p): \sum_{x_i \varepsilon \mathcal{I}_p} \lambda_i(t_p) y_i + \sum_{x_i \varepsilon \mathcal{B}'} \lambda_i(t_p) y_i = \sum_{x_i \varepsilon \mathcal{I}_p} \lambda_i(t_p) y_i^0 \qquad (4.48)$$

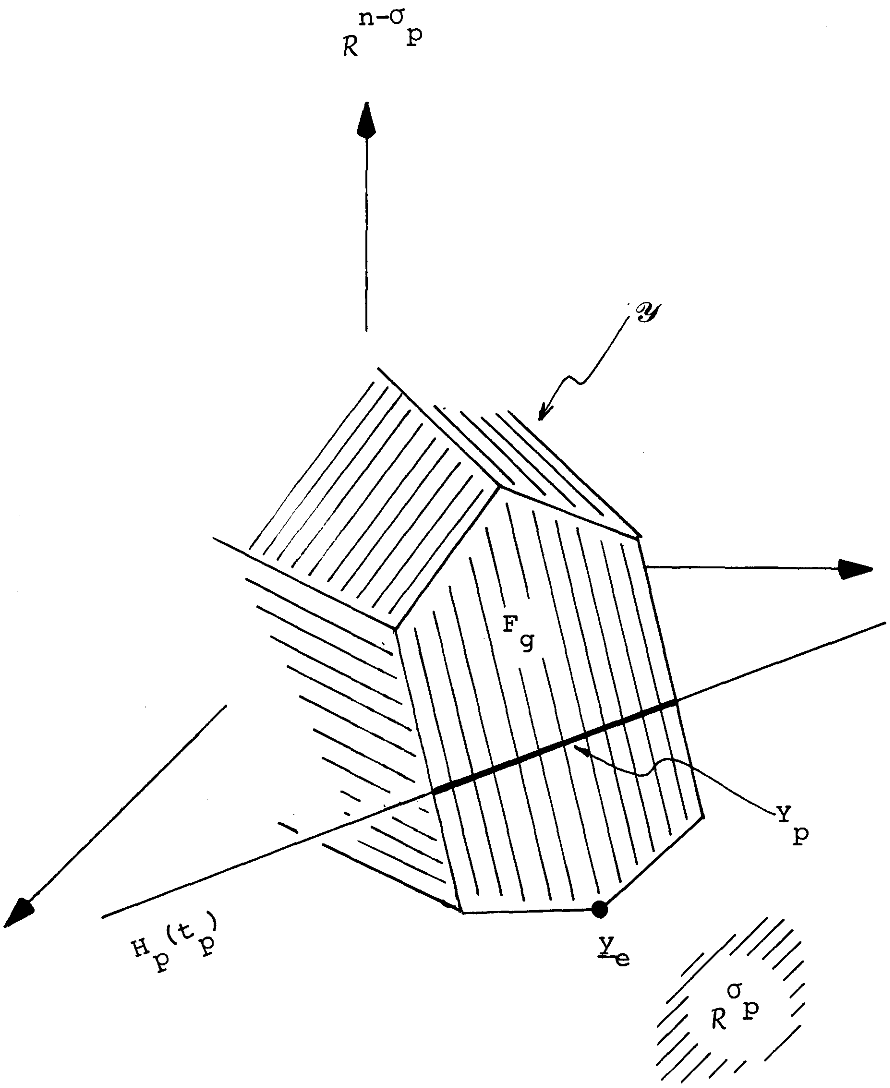


Figure 4.7 Situation in Which Y_p Does Not Contain Extreme Point of $\mathcal{Y}$.

where y_i^0 is the i^{th} component of any point in Y_p and $\mathcal{B}' \subset \mathcal{B}_p$. The values of $\lambda_i(t_p)$ for all $x_i \in \mathcal{B}'$ are now considered fixed. The global Hamiltonian hyperplane must have the form:

$$H(t_p): \sum_{x_i \in \mathcal{I}_p} \lambda_i(t_p) y_i + \sum_{x_i \in \mathcal{B}'_p} \lambda_i(t_p) y_i + \sum_{x_i \in \mathcal{B}''_p} \lambda_i(t_p) y_i$$

$$= \sum_{x_i \in \mathcal{I}_p} \lambda_i(t_p) y_i^0 \qquad (4.49)$$

where $\mathcal{B}'_p \cup \mathcal{B}''_p = \mathcal{B}_p$. We shall once again proceed by contradiction. We assume that $\mathscr{F}(Y_p, t_p, \mathcal{B}_p)$ contains two elements and we associate them with the distinct Hamiltonian hyperplanes $H'(t_p)$ and $H''(t_p)$ whose equations are given by

$$H'(t_p): \sum_{x_i \in \mathcal{I}_p} \lambda_i(t_p) y_i + \sum_{x_i \in \mathcal{B}'} \lambda_i(t_p) y_i + \sum_{x_i \in \mathcal{B}''} \lambda'_i(t_p) y_i$$

$$= \sum_{x_i \in \mathcal{I}_p} \lambda_i(t_p) y_i^0 \qquad (4.50)$$

and

$$H''(t_p): \sum_{x_i \in \mathcal{I}_p} \lambda_i(t_p) y_i + \sum_{x_i \in \mathcal{B}'} \lambda_i(t_p) y_i + \sum_{x_i \in \mathcal{B}''} \lambda''_i(t_p) y_i$$

$$= \sum_{x_i \in \mathcal{I}_p} \lambda_i(t_p) y_i^0. \qquad (4.51)$$

If we pick any extreme point of F_g, say $\underline{y}'_e$, we are assured that it lies in $H'(t_p) \cap H''(t_p)$ with $H'(t_p)$ and $H''(t_p)$ as given in (4.50) and (4.51).

Also, $\underline{y}_e'$ is an extreme point of $\mathcal{Y}$. Therefore, we may proceed as in part (a) with $\underline{y}_e$ replaced by $\underline{y}_e'$, $\mathcal{I}_p$ replaced by $\mathcal{I}_p \cup \mathcal{B}'$ and $\mathcal{B}_p$ replaced by $\mathcal{B}_p''$. The statements of the positive solution condition and the edge condition are analogous to those of part (a). We are again unable to produce more than n-3 edges of the requires type, thus contradicting the assumption that $\mathcal{F}(Y_p, t_p, \mathcal{B}_p)$ contains more than a single element. The case in which Y_p does not contain an extreme point of $\mathcal{Y}$ is complete.

The proof is therefore complete for the case in which $\mathcal{L}_p = \mathcal{B}_p$. We have demonstrated the uniqueness of $F(Y_p, t_p, \mathcal{B}_p)$. If $\mathcal{L}_p \subset \mathcal{B}_p$, but $\mathcal{L}_p \neq \mathcal{B}_p$, then

$$F(Y_p, t_p, \mathcal{L}_p) = F(Y_p, t_p, \mathcal{B}_p) \cap \mathcal{R}^{\sigma_p+\rho_p}$$

where the basis vectors of $\mathcal{R}^{\sigma_p+\rho_p}$ are the elements of $\mathcal{I}_{p-1}$. Therefore, $F(Y_p, t_p, \mathcal{L}_p)$ is the unique element of $\mathcal{F}(Y_p, t_p, \mathcal{L}_p)$.

■ Theorem 4.1

In view of relation (4.52) and the fact that the constructive algorithm calls for every state variable in $\mathcal{B}_p$ to leave the boundary backward in time, we shall be interested in $F(Y_p, t_p, \mathcal{B}_p)$ in order to determine all the leave-the-boundary costate values at the potential boundary junction time $t_p \in (-\infty, t_{p+1})$. For the purpose of the current discussion let us replace t_p by the running time variable $\tau \in (-\infty, t_{p+1})$. Then our geometric interpretation of the necessary conditions has the global Hamiltonian hyperplane $H(\tau)$ lying on the <u>unique</u> (by Theorem 4.1) $\mathcal{L}_p$-positive face $F(Y_p, \tau, \mathcal{B}_p)$ at every time $\tau \in (-\infty, t_{p+1})$. Also, we introduce one additional piece of notation.

<u>Notation 4.2</u>

$$\zeta_p \triangleq \text{cardinality } (\mathcal{B}_p) \tag{4.52}$$

Let us now consider the case in which Y_p contains an extreme point of $\mathcal{Y}$. Call this point $\underline{y}_e$. Now, in order for all ρ_p state variables in $\mathcal{B}_p$ to leave the boundary backward in time at τ, we must be able to have $y_i > 0$ for all $x_i \in \mathcal{B}_p$. Such points must therefore lie on $F(Y_p, \tau, \mathcal{B}_p)$. But at $\underline{y}_e$ we have $y_i = 0$ for all $x_i \in \mathcal{B}_p$ and the only edges which involve changes in y_i for $x_i \in \mathcal{B}_p$ are those of types (ii), (iii) and (iv) detailed in the proof of Theorem 4.1. These observations suggest the following condition:

> <u>Independent Edge Condition</u>. At any time τ, the unique $\mathcal{L}_p$-positive face $F(Y_p, \tau, \mathcal{B}_p)$ contains ζ_p independent edges emanating from $\underline{y}_e$ of either type (ii),,(iii) or (iv) detailed in the proof of Theorem 4.1.

For reference, we now list the edges of types (ii), (iii) and (iv) and detail what the requirement that the edges lie in $H(\tau)$ implies about the associated costates. First, we note that we are dealing with a unique global Hamiltonian which contains $F(Y_p, \tau, \mathcal{B}_p)$ at time τ

and express this as

$$H(\tau): \sum_{x_i \varepsilon \mathcal{I}_p} \lambda_i(\tau) y_i + \sum_{x_i \varepsilon \mathcal{B}_p} \lambda_i(\tau) y_i = \sum_{x_i \varepsilon \mathcal{I}_p} \lambda_i(\tau) y_i^e \tag{4.53}$$

$$\tau \ \varepsilon \ (-\infty, t_{p+1})$$

where y_i^e is the i^{th} component of $\underline{y}$ evaluated at the extreme point $\underline{y}_e$. Utilizing the fact that every edge under consideration must lie in the Hamiltonian hyperplane of (4.53) we obtain the following costate relationships corresponding to edges of types (ii), (iii) and (iv):

$$\text{(ii)} \quad \overset{\mathcal{B}_p}{\overset{\uparrow}{y_j}} \quad \longrightarrow \lambda_j(\tau) = 0. \tag{4.54}$$

$$\text{(iii)} \quad \underbrace{\overbrace{y_j \quad y_k}^{\mathcal{B}_p}} \longrightarrow \lambda_j(\tau) = \lambda_k(\tau) \tag{4.55}$$

$$\text{(iv)} \quad \underbrace{\overset{\mathcal{B}_p}{y_j} \quad \overset{\mathcal{I}_p}{y_k}} \longrightarrow \lambda_j(\tau) = \lambda_k(\tau) \tag{4.56}$$

We now shall establish what the edge graph will look like at a given time τ in light of the independent edge condition. We make the following claim and then justify it:

<u>Edge Graph Characterization</u>. Every $\mathcal{B}_p$-subgraph contains an edge of type (ii) or is connected to an $\mathcal{I}_p$-subgraph

with an edge of type (iv).

The justification for the above assertion is as follows: $\mathcal{B}_p$ contains ζ_p members, and by the independent edge condition it must contain ζ_p independent edges of type (ii), (iii) or (iv). Now, any $\mathcal{B}_p$-subgraph containing ζ'_p members, where $\zeta'_p \leq \zeta_p$, can contain no more than ζ'_p-1 independent edges of type (iii) by Lemma (4.1), part (a). However, by part (b) of that lemma any $\mathcal{B}_p$-subgraph of ζ'_p members may contain no more than ζ'_p independent edges. Therefore, every $\mathcal{B}_p$-subgraph must contain a number of edges equal to the number of its members. This may only be achieved by either adding an edge of type (ii) or connecting the $\mathcal{B}_p$-subgraph to an $\mathcal{I}_p$-subgraph by an edge of type (iv).

We denote a $\mathcal{B}_p$-subgraph which contains a type (ii) edge by $\overset{\uparrow}{\mathcal{B}}{}^{\gamma}_p$, where the integer γ ranges over the number of such subgraphs. The edge graph corresponding to this situation is pictured in Figure 4.8.

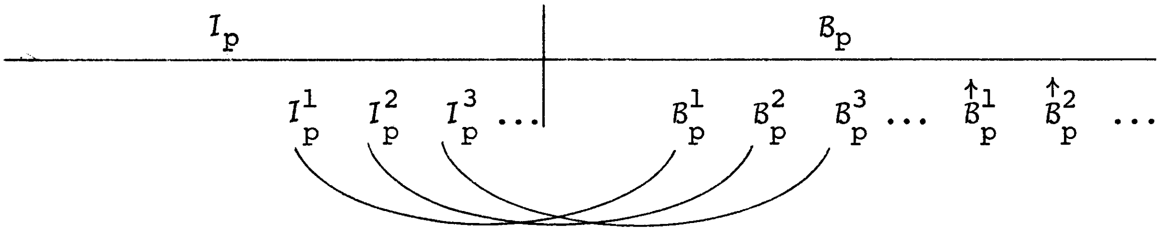

$\mathcal{I}_p$
$\mathcal{B}_p$
$\mathcal{I}^1_p$ $\mathcal{I}^2_p$ $\mathcal{I}^3_p$ ...
$\mathcal{B}^1_p$ $\mathcal{B}^2_p$ $\mathcal{B}^3_p$ ... $\overset{\uparrow}{\mathcal{B}}{}^1_p$ $\overset{\uparrow}{\mathcal{B}}{}^2_p$ ...

Figure 4.8 Edge Graph Which Satisfies Edge Graph Characterization

For convenience, we have labelled connected $\mathcal{B}_p$- and $\mathcal{I}_p$-subgraphs with the same superscript in Figure 4.8. Before continuing, we characterize edges of type (v) in order to round out our understanding of the costate implications of the edge graph.

$$\text{(v)} \quad \overbrace{\underbrace{y_j \quad y_k}_{}}^{\mathcal{I}_p} \longrightarrow \quad y_j(\tau) = y_k(\tau) \qquad (4.57)$$

We now return to consideration of some $y_i \in \mathcal{B}_p$. From the edge graph characterization above we see that y_i is either a member of some $\mathcal{B}_p^{\alpha}$ or some $\overset{\uparrow}{\mathcal{B}}{}_p^{\gamma}$. We now examine these two cases:

- If $y_i \in \overset{\uparrow}{\mathcal{B}}{}_p^{\gamma}$, then from (4.54) and (4.55) we have

$$\lambda_i(\tau) = 0. \qquad (4.58)$$

- If $y_i \in \mathcal{B}_p^{\alpha}$, then from (4.55)-(4.57) we have

$$\lambda_i(\tau) = \lambda_j(\tau) \qquad (4.59)$$

for any $y_j \in \mathcal{I}_p^{\alpha}$, where $\mathcal{I}_p^{\alpha}$ is the $\mathcal{I}_p$-subgraph connected to $\mathcal{B}_p^{\alpha}$.

We now present two lemmas which together characterize the dynamic costate behavior associated with single destination network problems with all unity weightings in the cost functional. This characterization will constitute the basic principle in the proofs of the subsequent results of this chapter.

Lemma 4.4: Consider single destination network problems with all unity weightings in the cost functional. Suppose all of the costates corresponding to state variables on boundary arcs achieve their leave-the-boundary values. Then every costate satisfies exactly one of the following conditions between boundary junctions:

$$\dot{\lambda}_i(\tau) = \lambda_i(\tau) = 0 \tag{4.60}$$

or

$$\dot{\lambda}_i(\tau) = -1 \text{ and } \lambda_i(\tau) > 0 \tag{4.61}$$

$$\forall\, \tau \in (-\infty, t_{p+1}).$$

Proof:

We begin by pointing out that the supposition that all of the costates corresponding to state variables on boundary arcs can achieve their leave-the-boundary values shall be shown to be valid in the discussion following Lemma 4.5. Also, we make the assumption of Section 3.3.2.2 that once a state variable enters an interior arc it remains in the interior backward in time. In Theorem 4.4 we shall show this assumption to be valid for the class of problems currently under consideration.

Consider the case in which some $\underline{x_i \in I_p}$. Then by the assumption that x_i remains in I_p as time runs to minus infinity and by (3.32) we have

$$\dot{\lambda}_i(\tau) = -1 \quad \forall\, \tau \in (-\infty, t_{p+1}).$$

We must also have $\lambda_i(t_{p+1}) > 0$, so $\lambda_i(\tau)$ satisfies (4.61).

On the other hand, consider the case in which some $\underline{x_i \in \mathcal{B}_p}$. Since we are considering the leave-the-boundary values of the costates corresponding to state variables on the boundary, our geometric interpretation has $H(\tau)$ lying on the unique $\mathcal{L}_p$-positive face $F(Y_p, \tau, \mathcal{B}_p)$ at every time $\tau \in (-\infty, t_{p+1})$. See Section 3.3.2.2. Recall that Y_p is the current set of operating points. Uniqueness is insured by Theorem 4.1. Note that we are considering the global Hamiltonian hyperplane as in Theorem 4.1. Now, the values of the leave-the-boundary costates at τ are determined by the face $F(Y_p, \tau, \mathcal{B}_p)$ on which $H(\tau)$ lies. But as τ runs from t_{p+1} to minus infinity $H(\tau)$ rotates due to changing values of $\lambda_j(\tau)$ for $x_j \in \mathcal{I}_p$. Therefore, we are basically concerned with investigating if $F(Y_p, \tau, \mathcal{B}_p)$ changes with time, and determining what is implied about the leave-the-boundary costate values. To this end, we now divide the remainder of the proof into two sections as in the proof of Theorem 4.1: (a) Y_p contains an extreme point of $\mathcal{Y}$ and (b) Y_p does not contain an extreme point of $\mathcal{Y}$.

(a) $\underline{Y_p \text{ contains an extreme point of } \mathcal{Y}}$

Denote the extreme point by $\underline{y}_e$. Then by the independent edge condition stated earlier, $F(Y_p, \tau, \mathcal{B}_p)$ has ζ_p independent edges emanating from $\underline{y}_e$. These edges are represented by an edge graph that has the format of Figure 4.8. The leave-the-boundary values of the costates at τ are determined by the edge graph at τ through (4.58) and (4.59).

We therefore wish to examine the time varying behavior of the edge graph.

By the edge graph characterization we know that the edge graph must maintain the format of Figure 4.7 as a function of time. We must consider both the situation in which the edge graph does not change with time and that in which it does change with time. The latter situation must correspond to the disappearance of a set of edges and the appearance of an equal number of edges elsewhere. We shall refer to the time at which such an edge change occurs as an edge switch time, denoted by τ_s. Also, the times just before and after the switch (in backward time) are denoted τ_s^+ and τ_s^- respectively.

We now return to our discussion of some state $x_i \varepsilon \mathcal{B}_p$ and examine the behavior of $\lambda_i(\tau)$ as determined by the edge graph behavior as a function of τ. The situations to be considered are:

1) No switch in edge occurs.

2) A switch in edge occurs at τ_s and y_i is in some $\mathcal{B}_p^{\gamma_1}$ before the switch (that is, at $\tau = \tau_s^+$).

3) A switch in edge occurs at τ_s and y_i is in some $\mathcal{B}_p^{\uparrow\alpha_1}$ before the switch (that is, at $\tau = \tau_s^+$).

We now consider the above situations one at a time:

1) No switch in edge occurs.

If $y_i \varepsilon \mathcal{B}_p^{\gamma_1}$ for some subgraph $\mathcal{B}_p^{\gamma_1}$, then $\lambda_i(\tau) = 0$ for all $\tau \varepsilon (-\infty, t_{p+1})$ by (4.58). In this case condition (4.60) of the lemma statement is satisfied. On the other hand, suppose $y_i \varepsilon \mathcal{B}_p^{\alpha_1}$ for some subgraph $\mathcal{B}_p^{\alpha_1}$. Then $\lambda_i(\tau) = \lambda_j(\tau)$ for any $y_j \varepsilon \mathcal{I}_p^{\alpha_1}$, where $\mathcal{I}_p^{\alpha_1}$ is the subgraph connected to $\mathcal{B}_p^{\alpha}$. Therefore, condition (4.61) of the lemma statement is satisfied.

2) A switch in edge occurs at τ_s and $y_i \varepsilon \mathcal{B}_p^{\gamma_1}$ at $\tau = \tau_s^+$.

The possibilities for this case are:

● $\tau = \tau_s^+ : y_i \varepsilon \mathcal{B}_p^{\uparrow\gamma_1}$

$\tau = \tau_s^- : y_i \varepsilon \mathcal{B}_p^{\uparrow\gamma_1}$

Here we are considering the situation in which the edge switch causes the composition of $\mathcal{B}_p^{\uparrow\gamma_1}$ to change but that y_i remains a member. Now, from (4.58) we have

$$\begin{aligned} \lambda_i(\tau) &= 0 \qquad \forall\, \tau \;\varepsilon\; (\tau_s^+, t_{p+1}) \\ \lambda_i(\tau) &= 0 \qquad \forall\, \tau \;\varepsilon\; (-\infty, \tau_s^-) \end{aligned} \tag{4.62}$$

so that λ_i remains equal to zero on $(-\infty, t_{p+1})$.

● $\tau = \tau_s^+ : y_i \varepsilon \mathcal{B}_p^{\gamma_1}$ $\qquad \gamma_1 \neq \gamma_2$

$\tau = \tau_s^- : y_i \varepsilon \mathcal{B}_p^{\gamma_2}$

Once again, we apply (4.58) to obtain

$$\begin{aligned} \lambda_i(\tau) &= 0 \qquad \forall\, \tau \;\varepsilon\; (\tau_s^+, t_{p+1}) \\ \lambda_i(\tau) &= 0 \qquad \forall\, \tau \;\varepsilon\; (-\infty, \tau_s^-) \end{aligned} \tag{4.63}$$

so that λ_i remains equal to zero on $(-\infty, t_{p+1})$.

- $\tau = \tau_s^+ : y_i \in \overset{\uparrow}{\mathcal{B}}{}_p^{\gamma_1}$

  $\tau = \tau_s^- : y_i \in \overset{\uparrow}{\mathcal{I}}{}_p^{\alpha}$

Equations (4.58) tells us that

$$\lambda_i(\tau) = 0 \qquad \forall\, \tau \in (\tau_s^+, t_{p+1}) \tag{4.64}$$

and (4.59) says

$$\lambda_i(\tau) = \lambda_j(\tau) > 0 \qquad \forall\, \tau \in (-\infty, \tau_s^-) \tag{4.65}$$

where $y_j \in \mathcal{I}_p^{\alpha}$. But (4.64) and (4.65) imply that λ_i experiences a jump of negative sign forward in time across τ_s. But this contradicts the fact that the costates may only experience positive jumps forward in time by equation (3.36). Therefore, this kind of edge switch cannot occur.

Summarizing the above situations, we see that if y_i is in some $\overset{\uparrow}{\mathcal{B}}{}_p^{\gamma_1}$ at any time τ on $(-\infty, t_{p+1})$ then λ_i remains zero on this interval. This behavior suits equation (4.60) of the lemma statement.

2) A switch in edge occurs at τ_s and $y_i \in \mathcal{B}_p^{\alpha_1}$ at $\tau = \tau_s^+$.

We begin with the following observation which results directly from the fact that the Hamiltonian hyperplane $H(\tau)$ rotates continuously in time:

<u>Edge Switch Condition:</u> Let τ_s be the time at which the edge graph switch occurs. Suppose the edge e^+ is replaced

by the edge e^- at τ_s. Then both e^+ and e^- are in the edge graph at τ_s.

This situation is illustrated in Figure 4.9, where we depict the hyperplane just before and after the switch of edges.

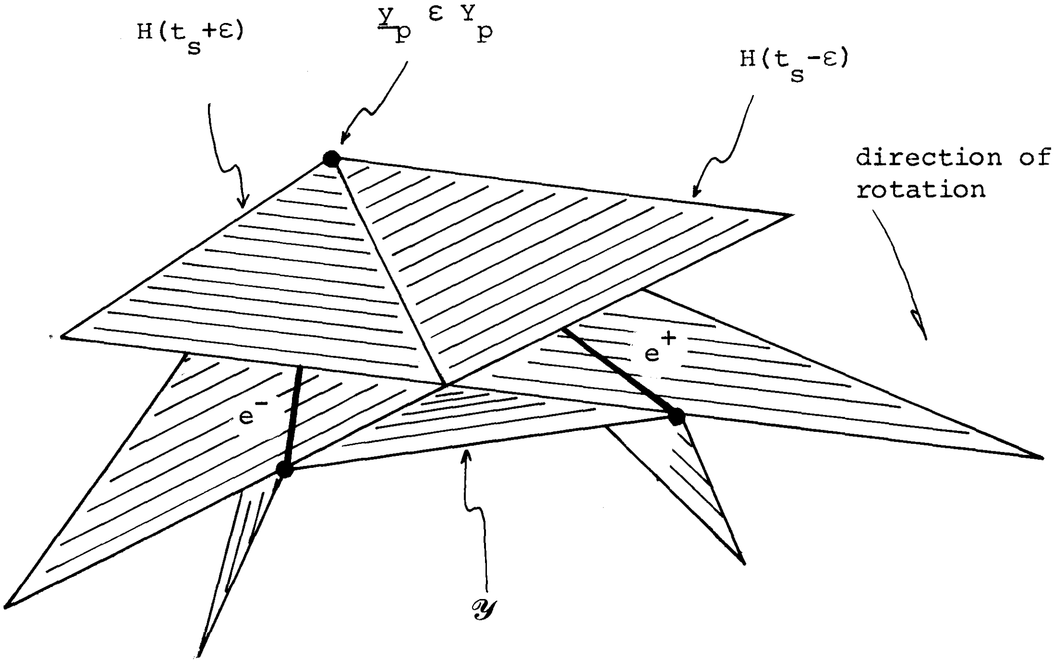


Figure 4.9 Continuously Rotating Hamiltonian Hyperplane

The possibilities for this case are:

- $\tau = \tau_s^+ : y_i \in \mathcal{B}_p^{\alpha_1}$

  $\tau = \tau_s^- : y_i \in \mathcal{B}_p^{\alpha_1}$

This corresponds to the situation in which a member is added or removed from $\mathcal{B}_p^{\alpha_1}$. From (4.55)-(4.57) we have

$$\lambda_i(\tau) = \lambda_j(\tau) \qquad \forall\, \tau \in (\tau_s^+, t_{p+1})$$

$$\lambda_i(\tau) = \lambda_j(\tau) \qquad \forall\, \tau \in (-\infty, \tau_s^-)$$

For any $y_j \in \mathcal{I}_p^{\alpha_1}$. Therefore

$$\lambda_i(\tau) = \lambda_j(\tau) \qquad \forall\, \tau \in (-\infty, t_{p+1}). \tag{4.66}$$

With the assumption that states on interior arcs remain on interior arcs as τ runs to minus infinity, we have from (3.32) that

$$\begin{aligned} &\lambda_j(t_{p+1}) > 0 \\ \text{and} \quad &\dot{\lambda}_j(\tau) = -1 \qquad \forall\, \tau \in (-\infty, t_{p+1}). \end{aligned} \tag{4.67}$$

Equations (4.66) and (4.67) imply

$$\begin{aligned} &\lambda_i(\tau) > 0 \\ \text{and} \quad &\dot{\lambda}_i(\tau) = -1 \qquad \forall\, \tau \in (-\infty, t_{p+1}). \end{aligned} \tag{4.68}$$

- $\tau = \tau_s^+ : y_i \in \mathcal{B}_p^{\alpha_1}$

  $\tau = \tau_s^- : y_i \in \mathcal{B}_p^{\alpha_2}$ $\qquad \alpha_1 \neq \alpha_2$

Then from (4.55)-(4.57) we have

$$\begin{aligned} \lambda_i(\tau) &= \lambda_{j_1}(\tau) \qquad \forall\, \tau \in (\tau_s^+, t_{p+1}) \\ \lambda_i(\tau) &= \lambda_{j_2}(\tau) \qquad \forall\, \tau \in (-\infty, \tau_s^-) \end{aligned} \tag{4.69}$$

for any $y_{j_1} \varepsilon I_p^{\alpha_1}$ and $y_{j_2} \varepsilon I_p^{\alpha_2}$. But the <u>edge switch condition</u> says that at the time of switch τ_s, $y_i \varepsilon B_p^{\alpha_1}$ <u>and</u> $y_i \varepsilon B_p^{\alpha_2}$. Consequently, by (4.69) we have

$$\lambda_i(\tau_s) = \lambda_{j_1}(\tau_s) = \lambda_{j_2}(\tau_s). \tag{4.70}$$

Once again invoking the assumption that states that interior arcs remain on interior arcs for all $\tau \varepsilon (-\infty, t_{p+1})$, we have

$$\dot{\lambda}_{j_1}(\tau) = \dot{\lambda}_{j_2}(\tau) = -1 \quad \forall \tau \varepsilon (-\infty, t_{p+1}) \tag{4.71}$$

and

$$\lambda_{j_1}(t_{p+1}) > 0.$$

Equations (4.69)-(4.71) imply

$$\begin{aligned} &\lambda_i(\tau) > 0 \\ &\dot{\lambda}_i(\tau) = -1 \quad \forall \tau \varepsilon (-\infty, t_{p+1}). \end{aligned} \tag{4.72}$$

$$\bullet\ \tau = \tau_s^+ : y_i \ \varepsilon \ B_p^{\alpha_1}$$
$$\uparrow$$
$$\tau = \tau_s^- : y_i \ \varepsilon \ B_p^{\gamma}$$

The edge switch condition says that both of the above edges are in effect at switch time τ_s. Under this circumstance, equation (4.58) tells us that

$$\lambda_i(\tau_s) = 0 \tag{4.73}$$

and (4.59) says

$$\lambda_i(\tau_s) = \lambda_j(\tau_s) > 0 \tag{4.74}$$

for some $y_j \in \mathcal{I}_p^{\alpha_1}$.

Since (4.73) and (4.74) are in contradiction, this kind of edge switch cannot occur.

Summarizing the above situations, we see that if y_i is in some $\mathcal{B}_p^{\alpha_1}$ at any time $\tau \in (-\infty, t_{p+1})$, then $\dot{\lambda}_i(\tau) = -1$ and $\lambda_i(\tau) > 0$ for all $\tau \in (-\infty, t_{p+1})$. This behavior suits equation (4.61) of the lemma statement. Hence, the lemma is proved if Y_p contains an extreme point of $\mathcal{Y}$.

(b) <u>Y_p does not contain an extreme point of $\mathcal{Y}$</u>

We take the same approach here as in part (b) of the proof of Theorem 4.1. The solution set Y_p lies on some face of $\mathcal{U}$, call it F_g, of dimension g, where $\sigma_p \leq g \leq n-1$. Refer to Figure 4.7. Following the reasoning of that section, we now have the global Hamiltonian

$$H(\tau): \sum_{x_i \in \mathcal{I}_p} \lambda_i(\tau) y_i + \sum_{x_i \in \mathcal{B}'_p} \lambda_i(\tau) y_i + \sum_{x_i \in \mathcal{B}''_p} \lambda_i(\tau) y_i$$

$$= \sum_{x_i \in \mathcal{I}_p} \lambda_i(\tau) y_i \tag{4.75}$$

where $\mathcal{B}'_p \cup \mathcal{B}''_p = \mathcal{B}_p$. If we now pick any extreme point of F_g, say $\underline{y}'_e$, we may proceed as in part (a) of this proof with $\underline{y}_e$ replaced by $\underline{y}'_e$, $\mathcal{B}_p$ replaced by $\mathcal{B}''_p$ and the edges corresponding to $\mathcal{B}'_p$ <u>fixed</u> (since these are the edges of the invariant face F_g upon which Y_p lies). We are again able to show that in the face of all possible switches in edges, the costates behave either as in (4.60) or (4.61). ■ Lemma 4.4

Now, statements (4.60) and (4.61) of Lemma 4.4 imply that the costate trajectories on $(-\infty, t_{p+1})$ appear as in Figure 4.10.

Lemma 4.5: Consider single destination network problems with all unity weightings in the cost functional. Suppose all of the costates corresponding to state variables on boundary arcs achieve their leave-the-boundary values. Then the costates do not experience jumps across boundary junction times.

Proof:

As in the case of Lemma 4.4 we point out that the supposition that all of the costates corresponding to state variables on boundary arcs can achieve their leave-the-boundary values is shown to be valid in the discussion following this proof.

Consider now the boundary junction time $t_p \in (-\infty, t_{p+1})$ at which the state variables in L_p leave the boundary backward in time. Then for $\tau \in [t_p, t_{p+1})$ the set of state variables on the boundary is B_p, the set of states in the interior is I_p and the $\underline{y}$-space solution set is Y_p. Likewise, for $\tau \in (-\infty, t_p)$ those sets are B_{p-1}, I_{p-1} and Y_{p-1} respectively.

Now, consider some $\underline{x_i \in I_p}$. Then by (3.22) λ_i does not experience a jump at t_p.

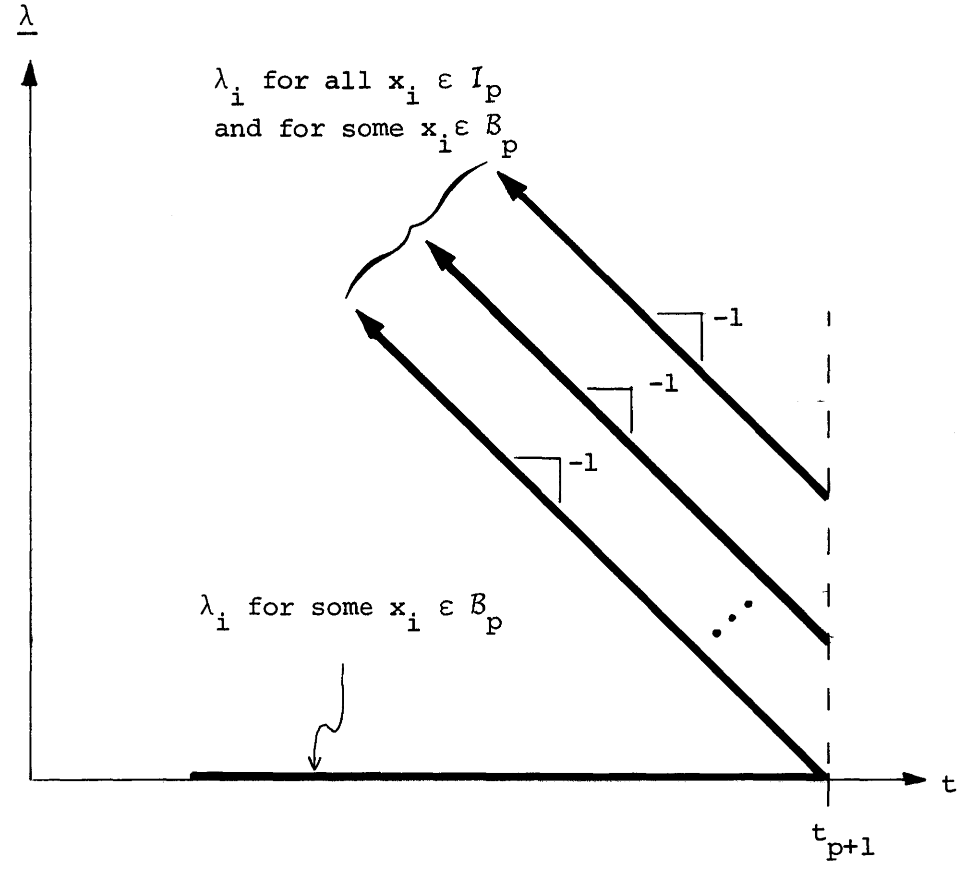

λ
λ_i for all $x_i \in \mathcal{I}_p$
and for some $x_i \in \mathcal{B}_p$
-1
-1
-1
λ_i for some $x_i \in \mathcal{B}_p$
t
t_{p+1}

Figure 4.10 Costate Trajectories Implied by Lemma 4.4

Consider now some $x_i \in \mathcal{B}_p$. If $x_i \in \mathcal{L}_p$, then by hypothesis $\lambda_i(t_p)$ is at its leave-the-boundary value. Therefore, no jump in λ_i is experienced when x_i leaves the boundary. On the other hand, suppose $x_i \in \mathcal{B}_p/\mathcal{L}_p$, that is, $x_i \in \mathcal{B}_{p-1}$. In this case λ_i is determined by the face $F(Y_{p-1}, t_p, \mathcal{B}_{p-1})$ on which $H(t_p)$ lies. We now claim that

$$F(Y_{p-1}, t_p, \mathcal{B}_{p-1}) = F(Y_p, t_p, \mathcal{B}_p). \qquad (4.76)$$

To verify (4.76) we first note that by the continuity of the Hamiltonian, the solution set Y_{p-1} lies on $F(Y_p, t_p, \mathcal{B}_p)$. Also, by definition $F(Y_p, t_p, \mathcal{B}_p)$ is the $\mathcal{B}_p$-positive face of $\mathcal{Y}$ with respect to Y_p and $H(t_p)$. Therefore, since $\mathcal{B}_{p-1} \subset \mathcal{B}_p$, $F(Y_p, t_p, \mathcal{B}_p)$ must also be the $\mathcal{B}_{p-1}$-positive face of $\mathcal{Y}$ with respect to Y_{p-1} and $H(t_p)$. The preceding statement is equivalent to (4.76). As a consequence of (4.76) we have that λ_i for $x_i \in \mathcal{B}_{p-1}$ does not change across t_p. This completes the proof that no costate jumps across t_p. ■ Lemma 4.5

We conclude from Lemma 4.5 that across the boundary junction t_p all the costates continue to propagate backward in time as in Figure 4.10 *except* for the following cases:

(i) $\lambda_i(\tau) = 0$ for $\tau \in (t_p, t_{p+1})$ and x_i is leaving the boundary at t_p. In this case $\dot{\lambda}_i(\tau) = -1$ for $\tau \in (-\infty, t_p)$.

(ii) $\lambda_i(\tau) = 0$ for $\tau \in (t_p, t_{p+1})$ and x_i remains on the boundary but the leave-the-boundary value of $\lambda_i(\tau)$,

$\tau \in (-\infty, t_p)$, dictates that $\lambda_i(\tau) = \lambda_j(\tau)$ for some λ_j satisfying case (i) above. In this case $\dot{\lambda}_i(\tau) = \dot{\lambda}_j(\tau) = -1$ for $\tau \in (-\infty, t_p)$.

The picture of the costate behavior across the boundary junction time is presented in Figure 4.11. Note that the set of x_i satisfying (i) and (ii) above are not dependent upon the boundary junction time t_p.

We now combine the results of Lemmas 4.4 and 4.5 into a general description of the costate trajectories. Consider any backward optimal trajectory corresponding to some sequence of state variables leaving the boundary backward in time at a series of boundary junction times. Then if all boundary costates achieve their leave-the-boundary value, then each costate λ_i satisfies the following trajectory:

$$\begin{aligned} \dot{\lambda}_i(\tau) &= \lambda_i(\tau) = 0 \qquad && \forall\, \tau \in (t_i, t_f) \\ \dot{\lambda}_i(\tau) &= -1 && \forall\, \tau \in (-\infty, t_i) \end{aligned} \tag{4.77}$$

where t_i is one of the boundary junction times between (and possibly including) t_f and the time at which x_i leaves the boundary. A typical set of costate trajectories for a backward optimal trajectory is depicted in Figure 4.12.

In Lemmas 4.4 and 4.5 we make the supposition that all boundary costates can achieve their leave-the-boundary values. We now verify that supposition. In (4.77) we describe the trajectory which must be

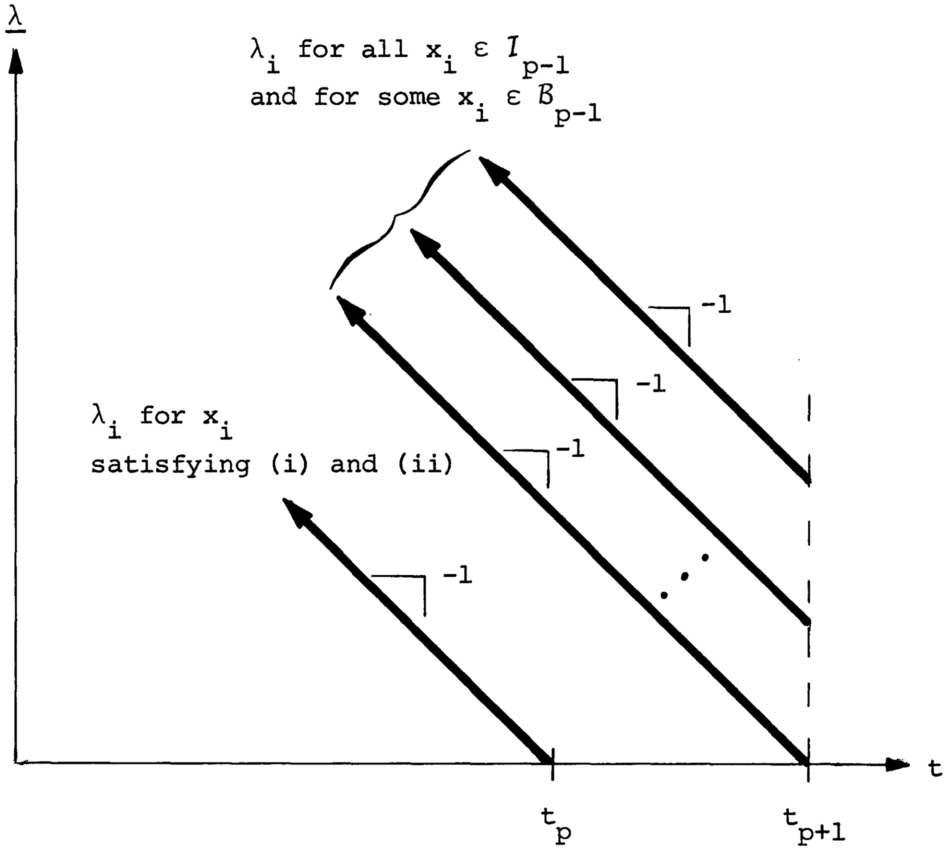

λ
λ_i for all $x_i \in \mathcal{I}_{p-1}$
and for some $x_i \in \mathcal{B}_{p-1}$
λ_i for x_i
satisfying (i) and (ii)
-1
-1
-1
-1
t
t_p
t_{p+1}

Figure 4.11 Costate Trajectories Implied by Lemma 4.5

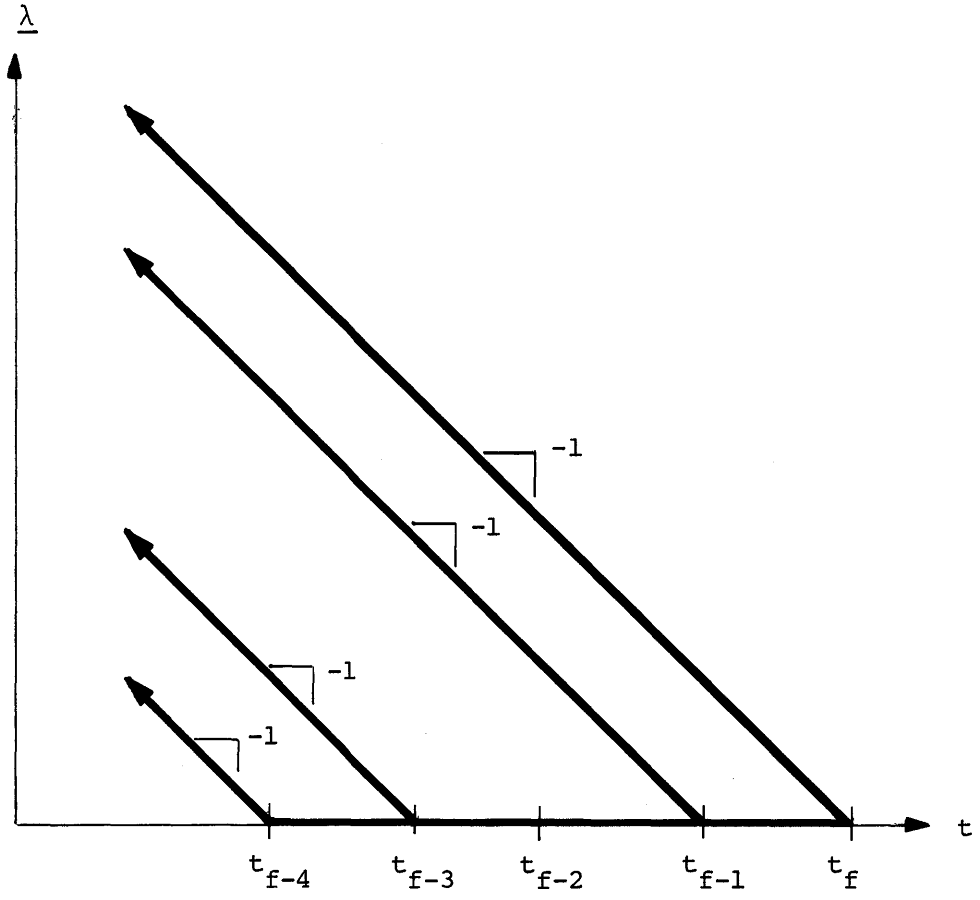


Figure 4.12 Typical Set of Costate Trajectories

satisfied by any costate in order that all boundary costates are maintained at their leave-the-boundary values. The required trajectories bear the following relationships between boundary and interior arcs (see (4.77)):

$$\forall\ x_i(\tau) = 0 \qquad \left\{\begin{array}{l} \dot{\lambda}_i(\tau) = -1 \\ \text{or} \\ \dot{\lambda}_i(\tau) = 0 \end{array}\right. \tag{4.78}$$

$$\forall\ x_i(\tau) > 0 \qquad \dot{\lambda}_i(\tau) = -1 \tag{4.79}$$

Since $\alpha_i = 1$ for all x_i, then (4.78) and (4.79) satisfy the necessary conditions (3.32)-(3.34). Therefore, the leave-the-boundary values of the costates can be achieved by costate trajectories which satisfy the necessary conditions.

Lemma 4.6: The set of optimal controls does not switch between boundary junction times; that is, there are no break points between boundary junctions.

Proof: In accordance with the necessary conditions (including those satisfying the constrained minimization), all optimal controls must be a solution to the global minimization

$$\underline{u}^*(\tau) = \text{ARG}\ \underset{\underline{u}\epsilon\mathcal{U}}{\text{MIN}}\ \underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau) \tag{4.80}$$

pointwise in τ for all $\tau \ \epsilon\ [t_0, t_f]$. Consider the typical control variable u_{ik} for a single destination network. Since u_{ik} enters into either one or two state dynamic equations in the fashion of equations (4.6) and (4.7) respectively, then the term of $\underline{\lambda}^T(\tau)\dot{\underline{x}}(\tau)$ which involves u_{ik} is

$$(\lambda_k(\tau) - \lambda_i(\tau))u_{ik}(\tau) \qquad (4.81)$$

where we take $\lambda_k = 0$ if node k is the destination. The constraint associated with u_{ik} is simply $0 \leq u_{ik} \leq C_{ik}$. The optimal control is then

$$u^*_{ik}(\tau) = \begin{cases} C_{ik} & \text{if } \lambda_k(\tau) - \lambda_i(\tau) < 0 \\ 0 & \text{if } \lambda_k(\tau) - \lambda_i(\tau) > 0 \\ [0, C_{ik}] & \text{if } \lambda_k(\tau) - \lambda_i(\tau) = 0 \end{cases} \qquad (4.82)$$

$$\forall\ (i,k) \in L.$$

The optimal control therefore depends only on $\mathrm{SGN}(\lambda_k(\tau) - \lambda_i(\tau))$. But from Lemma 4.4 we see that $\mathrm{SGN}(\lambda_k(\tau) - \lambda_i(\tau))$ remains constant on $(-\infty, t_{p+1})$ if no state variables leave the boundary on that interval; that is, $\mathrm{SGN}(\lambda_k(\tau) - \lambda_i(\tau))$ is constant between boundary junction times. Consequently, the set of optimal controls does not switch between boundary junctions. ■ Lemma 4.6

From Lemma 4.6 we know that for the class of problems under consideration <u>all feedback control regions are of the non-break variety</u>.

<u>Theorem 4.2</u>

For single destination network problems with all unity weightings in the cost functional, there is one subregion per region with respect to any set of state variables leaving the boundary.

Proof:

Consider the feedback control region $\mathscr{R}_p$ constructed from the optimal solution on $(-\infty, t_{p+1})$. Also, let L_p be the set of state variables which we choose to allow to leave from $\mathscr{R}_p$ backward in time, where $L_p \subset B_p$. According to Lemma 4.6, the optimal trajectory associated with the state variables in L_p leaving $\mathscr{R}_p$ backward in time,does not experience a break as time runs to minus infinity. Denote by $\Omega_{-\infty}$ the single set of optimal controls corresponding to the trajectory. Then according to Definition 3.8 we must show that the state variables in L_p leave with the same set of optimal controls $\Omega_{-\infty}$ from every point in $\mathscr{R}_p$. This is equivalent to showing that the set of optimal controls $\Omega_{-\infty}$ associated with the state variables in L_p leaving the boundary is the same for every boundary junction time $t_p \in (-\infty, t_{p+1})$. Note that we need not be concerned with the common breakwall stipulation of Definition 3.8 since there are no breakwalls.

We begin by noting that

$$\Omega_{-\infty} = \Omega^G \cap \Omega^{B_{p-1}} \tag{4.83}$$

where

$\Omega_{-\infty}$ is the set of optimal controls with which the state variables in L_p leave $\mathscr{R}_p$ backward in time and that keep the state variables in B_{p-1} on the boundary,

Ω^G is the globally optimal set of controls associated with the state variables in L_p leaving $\mathscr{R}_p$ backward in time,

and

$\Omega^{\mathcal{B}_{p-1}}$ is the set of all controls which keep the state variables in $\mathcal{B}_{p-1}$ on the boundary.

Now, $\Omega^{\mathcal{B}_{p-1}}$ is time invariant. Therefore, if we can show that Ω^G is the same for any $t_p \varepsilon (-\infty, t_{p+1})$, then the desired result is obtained. To this end, recall that Ω^G is given componentwise by (4.82) where $\tau \varepsilon (-\infty, t_p)$. From Lemmas 4.4 and 4.5 we conclude that $\mathrm{SGN}(\lambda_k(\tau) - \lambda_i(\tau))$ is the same for all $\tau \varepsilon (-\infty, t_p)$ and all $t_p \varepsilon (-\infty, t_{p-1})$. This observation coupled with (4.82) give us the desired result.

■ Theorem 4.2

Theorem 4.3

For problems involving single destination networks with all unity weightings in the cost functional, any solution to the constrained optimization problem (of Operation 1, part (a)) is also a globally minimizing solution.

Proof: Let us assume that Y_p is a set of σ_p-dimensional operating points which maximizes the restricted Hamiltonian hyperplane $H_p(\tau)$ with respect to the restricted constraint figure $\mathscr{Y}_p$ at time τ. In Section 3.3.2.2 we identified the following demonstration of global optimality of Y_p at τ: Find values of $\lambda_i(\tau)$ for $x_i \in \mathcal{B}_p$ such that $H(\tau)$ is maximized over $\mathcal{Y}$ at Y_p, that is, values which rotate the global Hamiltonian hyperplane $H(\tau)$ about $H_p(\tau)$ until $H(\tau)$ becomes tangent to $\mathcal{Y}$ at Y_p. In the discussion following Lemma 4.5 it was established that values of the boundary costates can always be found which bring $H(\tau)$ tangent to $F(Y_p, \tau, \mathcal{B}_p)$. Consequently, $H(\tau)$ may be brought tangent to $\mathcal{Y}$ since $F(Y_p, \tau, \mathcal{B}_p)$ is a face of $\mathcal{Y}$. ■ Theorem 4.3

The following theorem is the final result of this section. It is equivalent to saying that there always exists a solution which does <u>not</u> call for the return of state variables to the boundary backward in time. Hence, the assumption made in the beginning of the proof of Lemma 4.4 is valid.

<u>Theorem 4.4</u>

For problems involving single destination networks with all unity weightings in the cost functional, there always exists an optimal control which has $\dot{x}_i \leq 0$ for all x_i.

<u>Proof:</u> Suppose that x_i leaves the boundary backward in time at t_{p+1} as a member of some L_{p+1}. Then x_i leaves the boundary with some optimal slope

$$\dot{x}_i^* = \underline{b}_i \underline{u}_p + a_i < 0 \qquad (4.84)$$

where $\underline{b}_i$ is the i^{th} row of $\underline{B}$ and $\underline{u}_p$ is an optimal control which applies on $(-\infty, t_{p+1})$ if no other state variables leave the boundary backward in time. We shall now show that $\dot{x}_i^*$ given by (4.84) remains optimal on $-\infty, t_{p+1})$ even if other state variables leave the boundary backward in time on that interval. This property holds no matter what sequence of states leave the boundary. Suppose that the state variables in L_p leave the boundary at some boundary junction time $t_p < t_{p+1}$. See Figure 4.13. Let us now examine the optimal values of the controls which enter into $\dot{x}_i^*$. Consider the typical control variables u_{ik} and $u_{\ell i}$, where u_{ik} represents those control variables on links outgoing from node i and $u_{\ell i}$ represents control variables on links incoming to node i. Then the terms to be minimized corresponding to these control variables are

$$(\lambda_k(\tau) - \lambda_i(\tau))u_{ik}$$

and (4.85)

$$(\lambda_i(\tau) - \lambda_\ell(\tau))u_{\ell i}$$

$$\text{where } \tau \in (t_p, t_{p+1}).$$

Now since $x_i(\tau) > 0$ for $\tau \in (t_p, t_{p+1})$ then $\lambda_i(\tau) > 0$ for $\tau \in (t_p, t_{p+1})$. We begin by considering u_{ik} and the following two possibilities:

(i) $\lambda_k(\tau) - \lambda_i(\tau) \neq 0$ for $\tau \in [t_p, t_{p+1})$.

Then the optimal value of u_{ik} on $[t_p, t_{p+1})$ is

$$u_{ik}^* = \begin{cases} C_{ik} & \text{if } (\lambda_k(\tau) - \lambda_i(\tau)) < 0 \\ 0 & \text{if } (\lambda_k(\tau) - \lambda_i(\tau)) > 0 \end{cases} \tag{4.86}$$

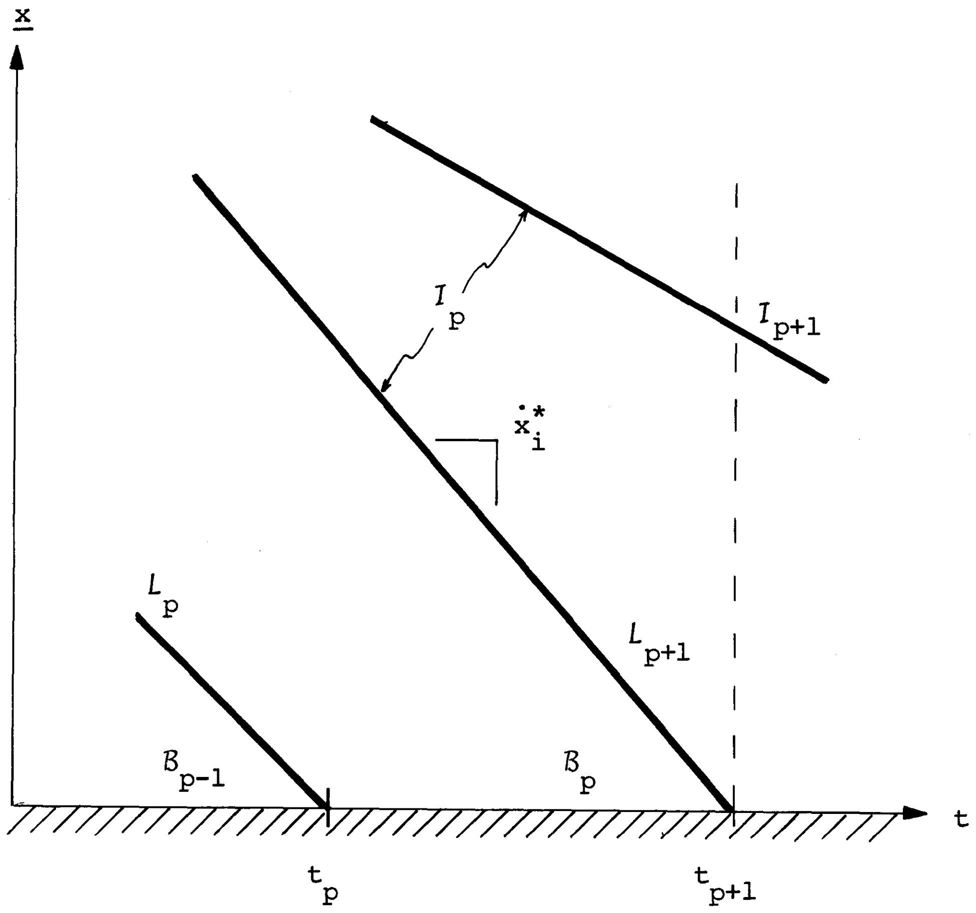


Figure 4.13 State Trajectories for Proof of Theorem 4.4

$$\tau \in [t_p, t_{p+1}).$$

But from (4.77) we know that $SGN(\lambda_k(\tau) - \lambda_i(\tau))$ is constant for $\tau \in (-\infty, t_{p+1})$ irregardless of which state variables leave the boundary on that interval. Therefore, the control in (4.85) is optimal on $\tau \in (-\infty, t_{p+1})$ irregardless of which state variables leave the boundary on that interval.

(ii) $\lambda_k(\tau) - \lambda_i(\tau) = 0$ for $\tau \in [t_p, t_{p+1})$

Then the optimal value of u_{ik} on $[t_p, t_{p+1})$ is any value $0 \leq u_{ik} \leq C_{ik}$ which together with all the other controls keeps the state variables in $\mathcal{B}_p$ on the boundary. But by (4.77) and the fact that $\lambda_i(\tau) > 0$ for $\tau \in (t_p, t_{p+1})$ we conclude that $\lambda_k(\tau) - \lambda_i(\tau) = 0$ for all $\tau \in (-\infty, t_{p+1})$ irregardless of which states leave the boundary on that interval. Therefore, if we consider the interval $[t_{p-1}, t_p)$, then the optimal value of u_{ik} on $[t_{p-1}, t_p)$ is any value $0 \leq u_{ik} \leq C_{ik}$ which together with all the other costates keeps the state variables in $\mathcal{B}_{p-1}$ on the boundary. But since $\mathcal{B}_{p-1} \subset \mathcal{B}_p$, any optimal control on $[t_p, t_{p+1})$ which keeps the state variables in $\mathcal{B}_p$ on the boundary must also be an optimal control on $[t_{p+1}, t_p)$ which keep the state variables in $\mathcal{B}_{p-1}$ on the boundary. Therefore, by induction on all possible time intervals backward in time corresponding to state variables leaving the boundary we may conclude that the optimal value of u_{ik} on $[t_p, t_{p+1})$ is optimal for all $\tau \in (-\infty, t_{p+1})$.

Arguments (i) and (ii) above also pertain to $u_{\ell i}$. Consequently, for any control variable which enters into the equation for $\dot{x}_i$ we have that any optimal value for $\tau \in [t_p, t_{p+1})$ is also an optimal value for all $\tau \in (-\infty, t_{p+1})$. We conclude that $\dot{x}_i^*$ is optimal for all $\tau \in (-\infty, t_{p+1})$. ■ Theorem 4.4

4.4 Computational Algorithm

In light of the results of Section 4.3, we are able to construct a computer algorithm which realizes the backward constructive algorithm of Chapter 3 for the special case under consideration. We begin by making the following observations:

- By Theorem 4.1, the leave-the-boundary value of the costates are unique and given by the coefficients of the global Hamiltonian hyperplane which lies on the $\mathcal{B}_p$-positive face. Therefore, all we need do to determine these costates at some boundary junction time t_p is to rotate $H(t_p)$ about $H_p(t_p)$ until it touches $F(Y_p, t_p, \mathcal{B}_p)$. We shall demonstrate a technique for this shortly.

- Suppose we have a set of state variables $\mathcal{I}_p$ off the boundary as we are working our way backward in time on an optimal trajectory. Then the rule for the execution of the basic step (see Section 3.3.2.2) calls for allowing the members of $\mathcal{L}_p$ to range over all possible subsets of $\mathcal{B}_p$ at the boundary junction time t_p. Now, from Theorem 4.2 we realize that t_p may be any time on $(-\infty, t_{p+1})$ so that we may arbitrarily take

$t_p = t_{p+1} - 1$. Furthermore, Theorem 4.3 tells us that each of the sets of state variables leaving the boundary corresponds to a globally optimal solution, so that the test for global optimality (Section 3.3.3.2, Operation 1, part (b)) is not required.

- Consider a particular set of state variables L_p leaving the boundary at some boundary junction time t_p. Then the initial set of controls corresponding to this situation persists to time equals minus infinity without a break according to Lemma 4.6. Therefore, we need only perform the constrained optimization problem once on $(-\infty, t_p)$, so that we may arbitrarily take the time at which it is performed as t_{p-1}. Moreover, every feedback control region constructed is of the non-break variety of Section 3.3.3.3.

- Theorem 4.4 tells us that no state variable is ever required by optimality to increase forward in time. Therefore, if we consider only those trajectories for which $\dot{x}_i^j(t) \leq 0$ for all x_i^j and all t, then we shall never have state variables returning to the boundary backward in time.

Now, the second observation above says that in order to be certain that we will fill up the state space with optimal controls, we must allow all possible combinations of state variables to leave the boundary in all possible orders. Furthermore, we are guaranteed that all of these sequences of state variables leaving the boundary will correspond to optimal trajectories. As pointed out in Section 3.3.2.2, by allowing all possible combinations of state variables to leave the boundary we are being conservative and will probably be "writing over" previously constructed feedback control regions on occasion. At any rate, by the final observation above we may list a priori all of the appropriate sequences where the departure of the state variables from the boundary may occur in intervals of one unit of time.

Example 4.1

Consider a network with three state variables x_1, x_2 and x_3. Then the sequences are (state variables enclosed in the same parentheses leave the boundary simultaneously):

t_{f-2}	t_{f-1}	t_f	t_{f-2}	t_{f-1}	t_f	t_{f-2}	t_{f-1}	t_f
x_1	x_2	x_3	x_3	x_1	x_2	-	x_3	(x_1,x_2)
x_2	x_1	x_3	-	(x_1,x_3)	x_2	-	x_2	(x_1,x_3)
-	(x_1,x_2)	x_3	x_3	x_2	x_1	-	x_1	(x_2,x_3)
x_1	x_3	x_2	x_2	x_3	x_1	-	-	(x_1,x_2,x_3)
			-	(x_2,x_3)	x_1			

A typical trajectory appears as in Figure 4.14.

● Example 4.1

The complete algorithm consists of performing all of the possible backward trajectories as in Example 4.1 and constructing the associated non-break feedback control regions. In order to describe the computa-

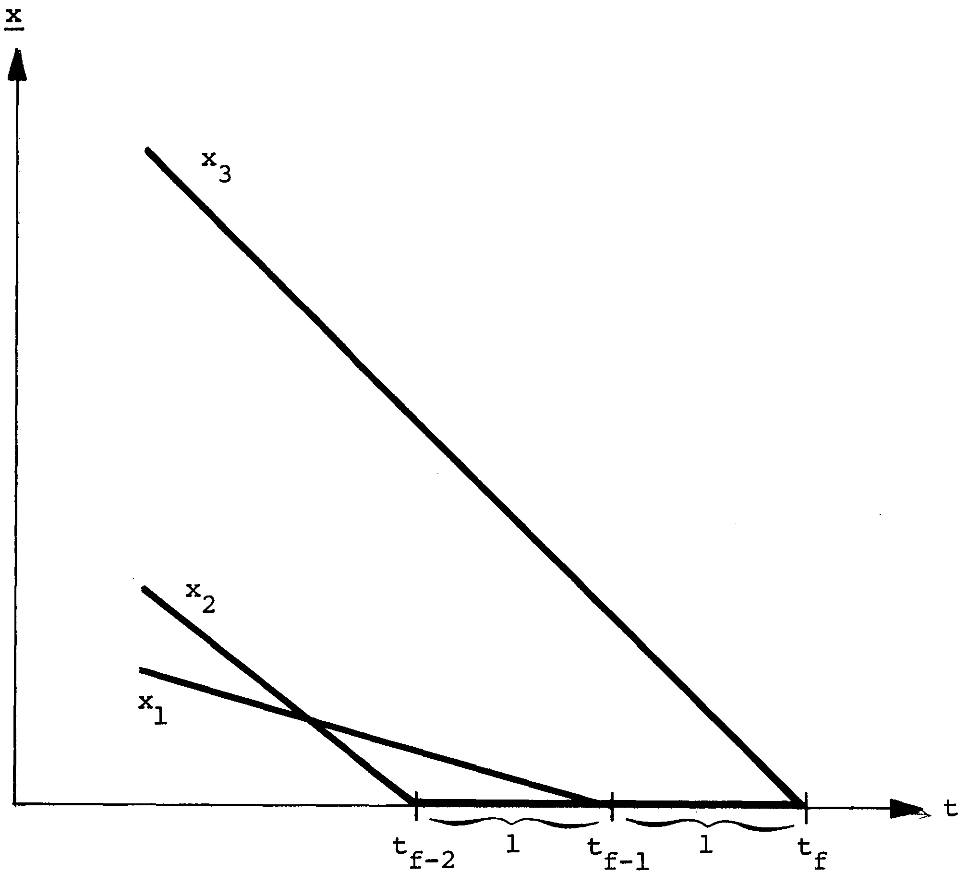


Figure 4.14 Typical Backward Trajectory for Example 4.1

tional algorithm we will follow one particular optimal trajectory backward in time from t_f for a few steps. In the course of the discussion we will demonstrate the method by which the leave-the-boundary costates may be determined.

Several Steps of the Computational Algorithm for the Construction of the Feedback Space

- Step 1
  - Operation 1

(i) Start at t_f.

(ii) Pick $\mathcal{L}_f \subset \mathcal{B}_f$. Then $\mathcal{I}_{f-1} = \mathcal{L}_f$. See Figure 4.15.

(iii) By Corollary 3.2, $\lambda_i(t_f) = 0 \quad \forall\, x_i \in \mathcal{L}_f$.

(iv) Arbitrarily set $t_f - t_{f-1} = 1$.

(v) $\lambda_i(t_{f-1}) = 1 \quad \forall\, x_i \in \mathcal{L}_f$.

(vi) Solve the following constrained optimization problem for all extremal solutions:

$$\underline{u}^* = \underset{\underline{u}\in\mathcal{U}}{\text{ARG MIN}} \sum_{x_i \in \mathcal{I}_{f-1}} \lambda_i(t_{f-1})\underline{b}_i^T\underline{u}$$

$$= \underset{\underline{u}\in\mathcal{U}}{\text{ARG MIN}} \sum_{x_i \in \mathcal{I}_{f-1}} (1)\underline{b}_i^T(\underline{u}) \tag{4.87}$$

$$\text{subject to} \quad \underline{b}_i^T\underline{u} \le -a_i \quad \forall\, x_i \in \mathcal{I}_{f-1} \tag{4.88}$$

$$\underline{b}_i^T\underline{u} = -a_i \quad \forall\, x_i \in \mathcal{B}_{f-1} \tag{4.89}$$

where $\underline{b}_i^T$ is the i^{th} row of $\underline{B}$. Denote this solution

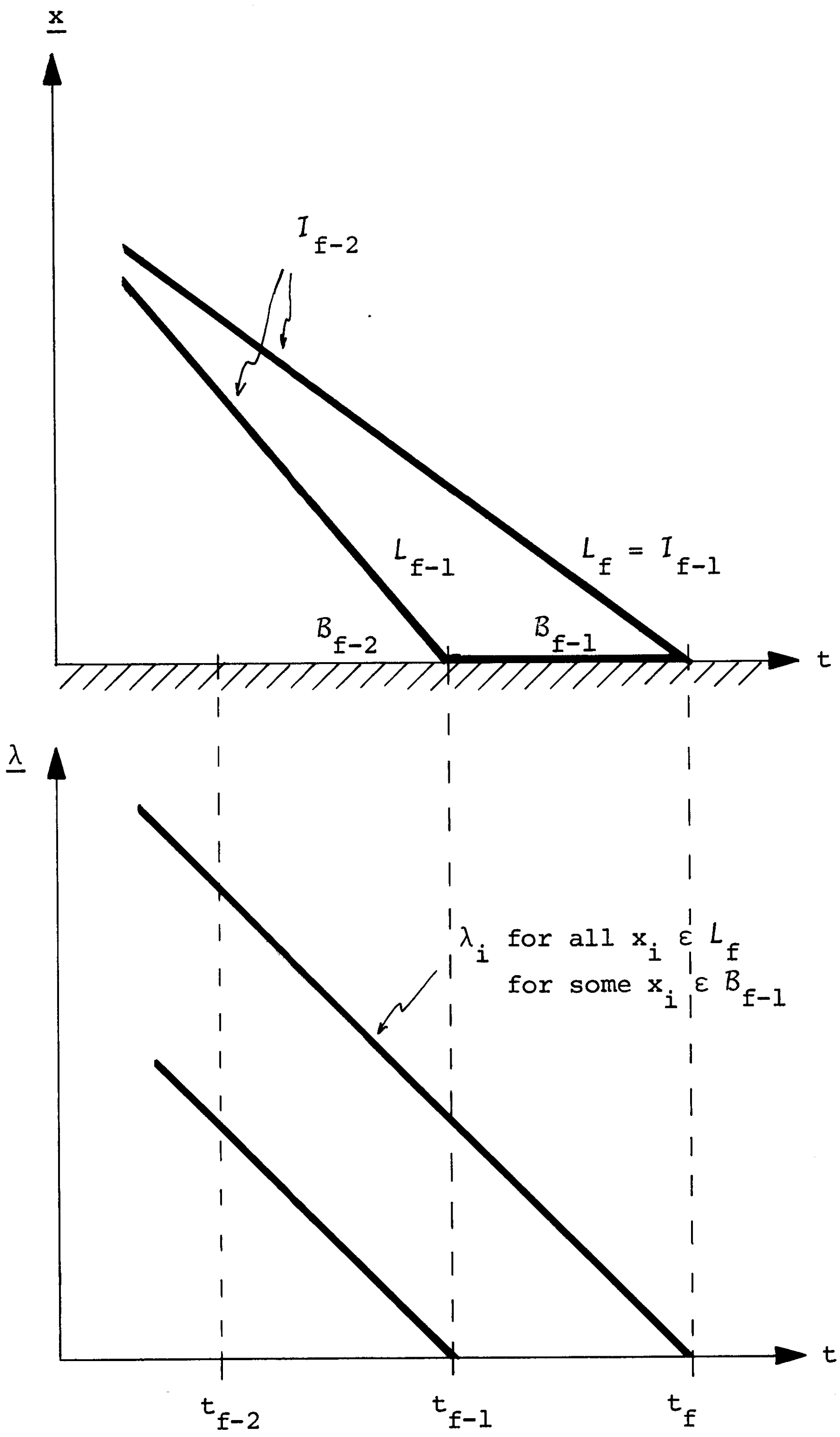

x
$\mathcal{I}_{f-2}$
L_{f-1}
$L_f = \mathcal{I}_{f-1}$
$\mathcal{B}_{f-2}$
$\mathcal{B}_{f-1}$
t
λ
λ_i for all $x_i \varepsilon L_f$
for some $x_i \varepsilon \mathcal{B}_{f-1}$
t
t_{f-2}
t_{f-1}
t_f

Figure 4.15 State and Costate Trajectories for Several Steps of the Computational Algorithm

set by Ω_{f-1}. We have used the notation Ω_{f-1} instead of the usual $\Omega_{-\infty}$ to distinguish this non-break optimal control set from those subsequently obtained in this discussion. By Theorem 4.3 we know that Ω_{f-1} is globally optimal.

- Operation 2

(i) Form the set of rays V_{f-1} which corresponds to Ω_{f-1} in the sense of Definition (3.10).

(ii) Construct the convex polyhedral cone

$$\mathscr{R}_{f-1} = \mathrm{Co}(\underline{0} \cup V_{f-1})/\underline{0}. \tag{4.90}$$

$\mathscr{R}_{f-1}$ is a non-break feedback control region with associated optimal control set Ω_{f-1}.

At this point we prepare for Step 2 as follows:

- Determination of the Leave-the-Boundary Costates

We shall require the values of $\lambda_i(t_{f-1})$ for all $x_i \in \mathcal{B}_{f-1}$. These are the values of the coefficients of the hyperplane $H(t_{f-1})$ which cause it to lie on the face $F(Y_{f-1}, t_{f-1}, \mathcal{B}_{f-1})$ of $\mathscr{Y}$, where Y_{f-1} corresponds to Ω_{f-1} obtained in Step 1, Operation 1, part (iv) above. By (4.77) we know that

$$\left.\begin{array}{ll} & \lambda_i(t_{f-1}) = 0 \\ \text{or} & \lambda_i(t_{f-1}) = t_f - t_{f-1} = 1 \end{array}\right\} \forall\, x_i \in \mathcal{B}_{f-1}.$$

In order to determine the appropriate value, we rotate the global

Hamiltonian hyperplane $H(t_{f-1})$ about the hyperplane $H_{f-1}(t_{f-1})$ until it touches the face of $F(Y_{f-1}, t_{f-1}, \mathcal{B}_{f-1})$ while holding $\lambda(t_{f-1})$ constant for all $x_i \in \mathcal{L}_f$. We may achieve this rotation by the following algorithm. Denote by $x_1, x_2, \ldots, x_{\zeta_{f-1}}$ all the members of $\mathcal{B}_{f-1}$.

(i) Set k = 1.

(ii) Release the constraint $\dot{x}_k = 0$ temporarily by adding a slack variable $z_k \geq 0$ to the appropriate equation of (4.89) in the following fashion:

$$\underline{b}_k^T \underline{u} + z_k = -a_k \tag{4.91}$$

The variable z_k is now a non-basic decision variable of the linear program (4.87)-(4.89). By (4.91) we have

$$z_k = -\dot{x}_k \tag{4.92}$$

Note that $\dot{x}_k = 0$ for any solution to (4.87)-(4.89).

(iii) Adjoin the term

$$\lambda_k(t_{f-1})\dot{x}_k = -\lambda_k(t_{f-1})z_k \tag{4.93}$$

to the cost function (4.87). The new linear program is now

$$\underline{u}^* = \text{ARG} \min_{\underline{u} \in \mathcal{U}} \left[\sum_{x_i \in \mathcal{I}_{f-1}} (1)\underline{b}_i^T\underline{u} - \lambda_k(t_{f-1})z_k \right] \tag{4.94}$$

$$\text{subject to} \quad \underline{b}_i^T\underline{u} \leq -a_i \qquad \forall\, x_i \in \mathcal{L}_f \tag{4.95}$$

$$\underline{b}_i^T\underline{u} = -a_i \qquad \forall\, x_i \in \mathcal{B}_{f-1}|x_k \tag{4.96}$$

$$\underline{b}_k^T\underline{u} + z_k = -a_k. \tag{4.97}$$

(iv) Set $\lambda_k(t_{f-1}) = 0$.

Therefore, any solution to (4.87)-(4.89) is also a solution to (4.94)-(4.97).

(v) Start with any solution to (4.87)-(4.89). Holding t_{f-1} fixed, solve (4.94)-(4.97) parametrically in $\lambda_k(t_{f-1})$ for $\lambda_k(t_{f-1})$ increasing linearly. Stop when a change in extreme point causes z_k to enter the basis with a strictly positive value (a degenerate pivot in the parametric program may cause z_k to enter the basis with zero value). When z_k is strictly positive then $\dot{x}_k$ is strictly negative. We have therefore caused the hyperplane $H(t_{f-1})$ to lie on $F(Y_{f-1}, t_{f-1}, x_k)$.

(vi) Return to the original solution of (4.87)-(4.89) chosen in (v) by performing reverse pivots from those executed in the parametric program.

(vii) Set $k = k+1$.

(viii) If $k > \zeta_{f-1}$ go to (x).

(ix) Go to (ii).

(x) End.

Upon the completion of the above algorithm we have caused the global Hamiltonian hyperplane $H(t_{f-1})$ to lie on $F(Y_{f-1}, t_{f-1}, x_1)$, $F(Y_{f-1}, t_{f-1}, x_2), \ldots, F(Y_{f-1}, t_{f-1}, x_{\zeta_{f-1}})$. But by Theorem 4.1 we know that $\mathscr{Y}(Y_{f-1}, t_{f-1}, \mathcal{B}_{f-1})$ contains the single member $F(Y_{f-1}, t_{f-1},$

$\mathcal{B}_{f-1}$). Therefore, $H(t_{f-1})$ must be lying upon $F(Y_{f-1}, t_{f-1}, \mathcal{B}_{f-1})$ and $\lambda_1(t_{f-1}), \lambda_2(t_{f-1}), \ldots, \lambda_{\zeta_{f-1}}(t_{f-1})$ must be the appropriate values of the leave-the-boundary costates at t_{f-1}.

- Step 2
  - Operation 1

(i) Start at boundary junction time t_{f-1}.

(ii) Pick $\mathcal{L}_{f-1} \subset \mathcal{B}_{f-1}$. Then $\mathcal{I}_{f-2} = \mathcal{L}_f \cup \mathcal{L}_{f-1}$. See Figure 4.14.

(iii) $\lambda_i(t_{f-1}) = t_f - t_{f-1} = 1 \quad \forall\, x_i \in \mathcal{L}_f$

$\lambda_i(t_{f-1})$ = (leave-the-boundary value determined at t_{f-1}) $\forall\, x_i \in \mathcal{L}_{f-1}$.

(iv) Arbitrarily set $t_{f-2} - t_{f-1} = 1$.

(v) $\lambda_i(t_{f-2}) = 2 \quad \forall\, x_i \in \mathcal{L}_f$

$\lambda_i(t_{f-2}) = \lambda_i(t_{f-1}) + 1 \quad \forall\, x_i \in \mathcal{L}_{f-1}$.

(vi) Solve the following constrained optimization problem for all extremal solutions:

$$\underline{u}^* = \underset{\underline{u}\in\mathcal{U}}{\text{ARG MIN}} \sum_{x_i \in \mathcal{I}_{f-2}} \lambda_i(t_{f-2})\underline{b}_i^T\underline{u} \tag{4.98}$$

$$\text{subject to} \quad \underline{b}_i^T\underline{u} \leq -a_i \quad \forall\, x_i \in \mathcal{I}_{f-2} \tag{4.99}$$

$$\underline{b}_i^T\underline{u} = -a_i \quad \forall\, x_i \in \mathcal{B}_{f-2} \tag{4.100}$$

Denote this solution set by Ω_{f-2}. By Theorem 4.3 Ω_{f-2} is globally optimal.

- Operation 2

(i) Form the set of rays V_{f-2} which corresponds to Ω_{f-2} in the sense of Definition (3.10).

(ii) Construct the convex polyhedral cone

$$\mathscr{R}_{f-2} = \text{Co}(\mathscr{R}_{f-1} \cup V_{f-2})/\mathscr{R}_{f-1} \qquad (4.101)$$

where $\mathscr{R}_{f-1}$ is given by (4.90). Then $\mathscr{R}_{f-2}$ is a non-break feedback control region with optimal control set Ω_{f-2}.

- Determination of the Leave-the-Boundary Costates

We determine the leave-the-boundary values of $\lambda_i(t_{f-2})$ for all $x_i \varepsilon \mathcal{B}_{f-2}$ using the parametric programming scheme detailed following Step (1).

The extension of the above steps to include the general step at boundary junction time t_p is a trivial exercise in notation and shall not be presented here. All of the linear programming manipulations called for in the algorithm are readily performed by available linear programming packages with the exception of the calculation of all extremal solutions. A program utilizing Chernikova's algorithm (described in Appendix A) was devised for this purpose.

Example 4.2

A computer program was devised to implement the above algorithm for a fully connected five node single destination network (a total of 4 state variables and 16 control variables). For this situation, complete execution of the algorithm calls for approximately 190 steps. All operations in the program performed with acceptable efficiency (with regard to speed of computation and storage requirements) with the notable exception of the program which calculates all of the extremal solutions. The poor efficiency of this program rendered the complete execution of the algorithm an impracticality. Example 4.2

4.5 Extension to General Network Problems

In this section we indicate how the geometrical approach used in this chapter may possibly be applied to the further development of the backward constructive algorithm for problems involving multi-destination networks with arbitrary weightings in the cost functional. In particular, we must cope with the four complicating properties discussed in Section 3.3.4. We present in Section 3.3.3.2 the geometrical interpretation in $\underline{y}$-space of three of these problems (global optimality, leave-the-boundary costates, subregions). For those cases, the discussion centers around the constraint figure $\mathcal{Y}$, for which we do not have an explicit representation. In particular, we talk about a particular class of faces of $\mathcal{Y}$ which lie adjacent to current operating points - these are the so-called "L_p-positive faces."

Currently, we possess no method for obtaining explicit knowledge of these faces for the general multi-destination network problem. Let us now describe our dilemma in more specific terms. Assume for the sake of argument that the current set of operating points is Y_p. Then for any potential boundary junction time $t_p \varepsilon (-\infty, t_{p+1})$ we wish to know the set of faces $\mathscr{F}(Y_p, t_p, \mathcal{B}_p)$; actually, we require the hyperplane in which each of these faces lies. Now, further assume that Y_p contains an extreme point of $\mathscr{Y}$, call it $\underline{y}_e$. Then some subset of the set of edges of $\mathscr{Y}$ emanating from $\underline{y}_e$ determine the required hyperplanes. The questions now becomes: (i) what is the set of edges of y emanating from y_e and (ii) which edges among these represent the required hyperplanes?

We begin our examination of question (i) by extending to general multi-destination networks the description of edges presented in Section 4.2 for single destination networks. Once again, we examine the form of the control constraint figure $\underline{D}\ \underline{u} \le \underline{C}$. For the link connecting node i to node k the appropriate constraint equation is

$$u_{ik}^{1}+u_{ik}^{2}+\dots+u_{ik}^{i-1}+u_{ik}^{i+1}+\dots+u_{ik}^{k}+\dots+u_{ik}^{N}+z_{ik} = C_{ik} \qquad (4.102)$$

$$z_{ik} \geq 0$$

where we have adjoined the non-negative slack variable z_{ik}. In (4.102) the control variable u_{ik}^{k} is termed a direct control and all other control variables are referred to as indirect controls (Definition 4.3). Note that there is exactly one direct control per link. Now, the complete set of equations of the type (4.102) corresponding to $\underline{D}\ \underline{u} \le \underline{C}$ has the form:

$$r\left\{\underbrace{\begin{matrix} u+u+\cdots \\ & u+u+\cdots \\ && \ddots \\ &&& u^{1}_{ik}+\cdots+u^{N}_{ik} \\ &&&& \ddots \\ &&&&& u+u+\cdots \end{matrix}}_{m}\;\underbrace{\begin{matrix} +z \\ & +z \\ && \ddots \\ &&& +z_{ik} \\ &&&& \ddots \\ &&&&& +z \end{matrix}}_{r}\;\begin{matrix} = C \\ = C \\ \vdots \\ = C_{ik} \\ \vdots \\ = C \end{matrix}\right. \tag{4.103}$$

where we have eliminated all subscripts except for ik to avoid notational havoc. Recall that r is the number of links and m is the dimension of the control vector. Now, notice that each control variable enters into exactly one equation in (4.103); hence, (4.103) is immediately in canonical form with respect to any set of r variables chosen (either u's or z's) to be basic. For instance, if we consider the row corresponding to C_{ik}, then we pick z_{ik} or some control u^{j}_{ik} $(j \neq i)$ to be basic and set its value equal to C_{ik} and the value of the remaining variables to zero. We do this for each row. Note that since $\underline{C} \geq 0$, every basic solution to (4.103) automatically has $\underline{u} \geq 0$.

Therefore, every extreme point of $\mathcal{U}$ is represented algebraically by a point whose ik coordinate has value of either zero or C_{ik}. Movement along an edge of $\mathcal{U}$ is equivalent to picking one variable which is currently basic at value C_{ik} (either some u^{j}_{ik}, $j \neq i$, or z_{ik}) and replacing it in the basis by one of the remaining variables. The possible points which may occur are:

1) direct control exchanges with slack on link ik
2) intermediate control exchanges with slack on link ik
3) direct control exchanges with indirect control on link ik
4) indirect control exchanges with indirect controls on link ik.

Notice that there is no pivot between two direct controls since there is exactly one direct control per link. Now, the edges corresponding to pivots (1) and (2) above are those of Type 1 and Type 2 detailed in Section 4.2. We now characterize the following types of edges of $\mathcal{Y}$ emanating from $\underline{y}_e$ corresponding to pivots (3) and (4) above:

<u>Type 3</u> - Exactly <u>three</u> coordinates of $\underline{y}$ change from their values at $\underline{y}_e$. This edges corresponds to an edge of $\mathcal{U}$ for which a <u>direct control and an indirect control</u> change places in the basis. From equations (3.38) and (3.162) we see that the possibilities are:

$$\left\{\begin{array}{l} u_{ik}^{k} = C_{ik} \\ u_{ik}^{j} = 0 \end{array}\right. \quad \xrightarrow{\text{pivot}} \quad \left\{\begin{array}{l} u_{ik}^{k} = 0 \\ u_{ik}^{j} = C_{ik} \end{array}\right. \longrightarrow \left\{\begin{array}{l} \Delta y_{i}^{k} = -C_{ik} \\ \Delta y_{i}^{j} = C_{ik} \\ \Delta y_{k}^{j} = -C_{ik} \end{array}\right.$$

$$\left\{\begin{array}{l} u_{ik}^{k} = 0 \\ u_{ik}^{j} = C_{ik} \end{array}\right. \quad \xrightarrow{\text{pivot}} \quad \left\{\begin{array}{l} u_{ik}^{k} = C_{ik} \\ u_{ik}^{j} = 0 \end{array}\right. \longrightarrow \left\{\begin{array}{l} \Delta y_{i}^{k} = C_{ik} \\ \Delta y_{i}^{j} = -C_{ik} \\ \Delta y_{k}^{j} = C_{ik} \end{array}\right.$$

where Δy_{i}^{k}, Δy_{i}^{j} and Δy_{k}^{j} indicate the change in the respective coordinates of $\underline{y}$ when moving from one end to another of an edge.

Type 4 - Exactly four coordinates of $\underline{y}$ change from their values at $\underline{y}_e$. This edge corresponds to an edge of $\mathcal{U}$ for which two indirect controls change places in the basis. From equations (3.38) and (3.162) we see that the possibilities are:

$$\begin{cases} u_{ik}^j = c_{ik} \\ u_{ik}^\ell = 0 \end{cases} \xrightarrow{\text{pivot}} \begin{cases} u_{ik}^j = 0 \\ u_{ik}^\ell = c_{ik} \end{cases} \rightarrow \begin{cases} \Delta y_i^\ell = c_{ik} \\ \Delta y_i^j = -c_{ik} \\ \Delta y_k^\ell = -c_{ik} \\ \Delta y_k^j = c_{ik} \end{cases}$$

$$\begin{cases} u_{ik}^j = 0 \\ u_{ik}^\ell = c_{ik} \end{cases} \xrightarrow{\text{pivot}} \begin{cases} u_{ik}^j = c_{ik} \\ u_{ik}^\ell = 0 \end{cases} \rightarrow \begin{cases} \Delta y_i^\ell = -c_{ik} \\ \Delta y_i^j = c_{ik} \\ \Delta y_k^\ell = c_{ik} \\ \Delta y_k^j = -c_{ik} \end{cases}$$

We therefore have a characterization of all four possible types of edges of $\mathcal{Y}$. Since the maximum number of coordinates of $\underline{y}$ which change when moving along any given edge is four, the constraint figure $\mathcal{Y}$ for multidestination networks is not significantly more complex than that for single destination networks (where the maximum number is two). One should be able to exploit this fact to achieve a better understanding of the nature of the complicating properties of the algorithm for general multidestination network problems. For instance, it should not be difficult to establish an upper bound on the number of members of $\mathscr{F}(Y_p, t_p, \mathcal{L}_p)$. This is equivalent to placing an upper bound on both the number of extreme points in the leave-the-boundary costate set and the number of subregions in $\mathcal{R}_p$ with respect to $\mathcal{L}_p$. However, in order to implement the algorithm one must possess explicit knowledge of the set of edges of the members of $\mathscr{F}(Y_p, t_p, \mathcal{L}_p)$ which emanate from $\underset{\sim}{y}_e$. A technique must therefore be devised to identify those edges of $\mathcal{U}$ which correspond to the required edges of $\mathscr{Y}$.

Chapter 5

CONCLUSION

5.1 Discussion

In this section we attempt to place in perspective the accomplishments of the preceding chapters. In Chapter 1 we express the desire to develop a dynamic procedure for message routing in a store-and-forward data communication network. As an initial step toward achieving this goal a linear dynamic state space model is presented in Chapter 2 which represents the flow and storage of messages in such a network. Although the model is sufficiently general to represent a stochastic user demand environment, we choose to consider the simplifying case of deterministic inputs for initial analytical studies. We associate with the model a linear integral cost functional which is the weighted total delay experienced by all the messages as they travel through the network until all reach their respective destinations. The desired minimization of the cost functional results in a linear optimal control problem with linear state and control variable inequality constraints. We seek a feedback solution to this problem.

We now digress a moment to discuss the basic approach. Attempts to apply optimal control theory to problem areas in which other techniques have proven inadequate have become common practice since the inception of the field in about 1960. Not coincidentally, this timing corresponds to the beginning of the era of computers, a tool which is relied upon by both theorists and practitioners of optimal control. Over the years optimal control has enjoyed a number of successes, most notably in the areas of application of Linear-Quadratic-Gaussian (LQG) control and filtering. However, it has also become apparent that "optimal" in the mathematical sense does not always imply "best" in some more general sense, so that optimal control theory has a great many detractors. See for example Horowitz [1975]. One complaint is that in minimizing a single objective, optimal solutions often ignore and sometimes aggravate other properties of the system performance. Also, optimal solutions are often obtained and implemented at great computational expense. Finally, if the objective curve is relatively flat near the optimum, then the marginal benefit obtained from the optimal solution over some more easily obtained sub-optimal solution may not justify the additional expense. One must therefore carefully weight the disadvantages of an application of optimal control with respect to the advantages. The fundamental advantages for our problem are described in Sections 1.2 and 1.3. We shall point out some disadvantages in the course of this discussion.

Returning now to the optimal message routing problem, the solution which we seek calls for all of the message storage in the network to go to zero at the final time. It is clear that this requirement does not correspond to the usual data network mode of operation in which there is a continuous input and flow of messages through the network. However, a particular case for which this requirement would be suitable is when one desires to relieve network congestion locally in time by getting rid of current message backlogs stored in the nodes. In this case, the inputs to the networks would be regulated so that the backlogs could be emptied with the available link capacity. Values of constant inputs for which this is possible are given in Theorem 3.3.

Having formulated the optimal control problem, the remainder of the thesis is devoted to developing a technique for finding the feedback solution for the case in which the inputs are constant in time. Of course, the constant input assumption further restricts the applicability of this approach to real network problems. However, one can conceive of situations in which the inputs are regulated at constant values, such as the backlog emptying procedure described above. From the optimal control viewpoint, constant inputs appear to provide us with the minimum amount of structure required to characterize and construct the feedback solution with reasonable effort.

The backward constructive algorithm is described in principle for multidestination networks, although it is not currently understood how this algorithm may be implemented to obtain the solution for this general class of problems. However, the complicating aspects of the algorithm disappear when we consider problems involving single destination networks with all unity weightings in the cost functional. Our ability to completely solve only this special class of problems is not surprising when we consider that for static maximum flow problems the general multi-commodity problem remains unsolved. It stands to reason that a dynamic multi-destination problem must be quite difficult!

Throughout the development we have exploited the total linearity of the problem (in dynamics, cost and constraints) to obtain many of our results. In particular, we have frequently employed concepts in linear programming both in the synthesis of the backward constructive technique and in the development of the computational algorithm. It is felt that this approach may prove helpful in other optimal control problems with a high degree of linearity.

Perhaps the primary disadvantage associated with the feedback solution is the amount of computer storage which will be required to implement it. The feedback control regions must be specified by a set of linear inequalities which in general may be extremely large, and the optimal controls within these regions must also be specified. On the bright side, however, we note that if the particular network for which we are providing the feedback solution contains one or more large computers, the required storage may possibly be made available. Another disadvantage is the amount of computation required in the original construction of the feedback solution. In particular, the program developed to determine all the extremum solutions to a linear program (Appendix A) requires a substantial amount of computation time and storage.

5.2 Summary of Results and Related Observations

5.2.1 Results

In this section we describe briefly the primary results of the thesis in the order in which they appear.

In Section 3.2.1 we develop the necessary conditions of optimality associated with the linear optimal control problem with linear state and control variable inequality constraints. We also prove the important result that the necessary conditions are also sufficient. Note that these results are valid for general deterministic inputs.

Starting in Section 3.2.2 we assume that all the inputs to the network are constant in time. In the same section we state and prove a necessary and sufficient condition for the controllability of any initial state to zero. Utilizing the necessary conditions of optimality, we prove in Section 3.2.3 that the feedback regions of interest are convex polyhedral cones in the state space.

Section 3.3 is devoted to a general description of the algorithm for the backward construction of the feedback space. The steps of the algorithm are detailed in Section 3.3.2.2. In Section 3.3.3.1 we show how parametric linear programming may be employed to efficiently solve the pointwise linear program in time. The geometrical interpretation is introduced in Section 3.3.3.2 and shown to be useful in analyzing several of the difficult problems associated with the algorithm. The construction of feedback control regions is detailed in Sections

3.3.3.3 and 3.3.3.4 respectively.

We show by example in Section 3.3.4.1 that for this problem there may be non-unique costates corresponding to state variables on interior arcs.

Chapter 4 is devoted to problems involving single destination networks with all unity weightings in the cost functional. In Section 4.3 we confront those properties which complicate the general algorithm and show that they present no problem for the class of problems under discussion. All proofs utilize the geometrical interpretation introduced in Section 3.3.3.2. In particular, we show that all costates corresponding to state variables on the boundary are unique. We also prove that the control does not break between boundary junction times. In Section 4.4 we present a complete computer implementable algorithm for the construction of the feedback solution for this special case.

5.2.2 Related Observations

Although the approach which we have taken may be limited in practical applicability, the results which we have obtained can provide valuable insight into the dynamic message routing problem. In this section we discuss several issues from this point of view.

We begin with a discussion of the state information required to implement the optimal control. As pointed out in Section 2.5, by specifying a feedback control function of the type $\underline{u}(t) = u(t, \underline{x}(t))$ we are assuming that total information regarding the storage state $\underline{x}(t)$

is available at each node of the network for control decisions at that node. Such a scheme requires a centralized method of observing and communicating the state of the system at every point in time. However, in practice one may wish to consider control strategies which do not require total knowledge of the state at every time. For instance, it may be desirable to have a decentralized strategy that makes control decisions based on local state information only.

Such considerations lead us to examine the nature of the state information requirements associated with the solution we have obtained in the specific case of constant inputs. We have found that the structure of the solution is such that regions in the state space of common optimal control are convex polyhedral cones. This situation is depicted conceptually for a two dimensional example in Figure 5.1.

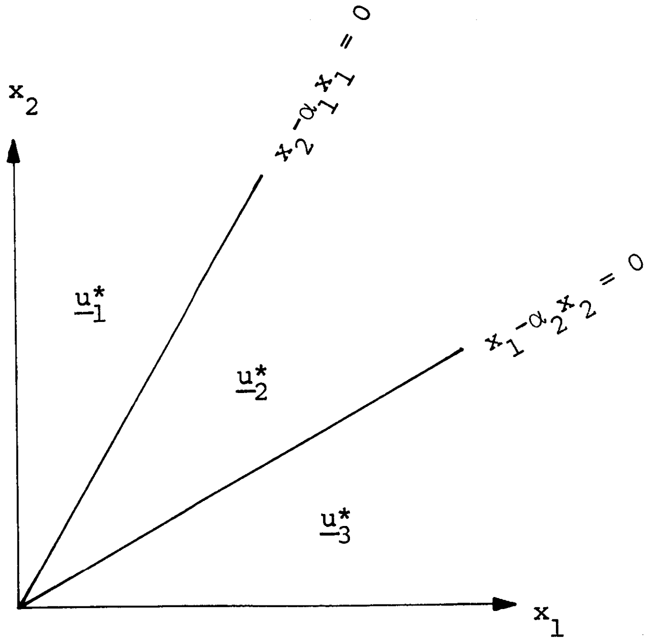


Figure 5.1 Typical Form of Feedback Solution

We see from Figure 5.1 that the optimal control is always a function of both x_1 and x_2. A completely decentralized scheme is therefore not possible. However, due to the nature of the optimal control regions, we do not require exact values of x_1 and x_2 but only their <u>relative</u> values. For example, we need only know that $\frac{x_2}{x_1} \geq \alpha_1$ and $x_1 > 0$ to conclude that $\underline{u}_1^*$ is the appropriate optimal control. This relationship may hold true for relatively long periods of time, so that periodic rather than continuous updating of the ratio $\frac{x_2}{x_1}$ should prove to be rather good. In fact, based upon the current state and optimal control it is easy to predict (due to linearity) the time at which the state will move from the current feedback control region into an adjacent feedback control region in the absence of outside disturbances or variations in the system parameters. Therefore, one may wish to update the state information (and the optimal control, if called for) at these predicted switch times. Although this scheme would not cut down on the amount of state information required, it drastically reduces the frequency with which it must be supplied.

For the purpose of our next discussion we refer to the network problem of Example 3.1. From Figure 3.9 we see that if the state lies in the region between the lines $x_2-3x_1 = 0$ and $x_1-3x_2 = 0$ (not including these lines) then any admissible values of the control variables u_{12} and u_{21} are optimal. Therefore, looping of messages between nodes 1 and 2 of Figure 3.7 may be optimal. However, such looping is not desirable in data communication networks for obvious reasons. This problem is an example of peripheral considerations which may be ignored by optimal solutions with limited objectives. In order to alleviate this situation, one must modify the problem statement appropriately. For this case, optimal solutions may tend to avoid looping if we incorporate a term linear in the control, say $\underline{\pi}^T\underline{u}$, into the cost functional. The idea here is to discriminate against frivolous use of the links. Fortunately, this modification does not complicate the solution technique presented in this thesis since the only change from before is that the pointwise minimization is now

$$\underline{u}^*(\tau) = \underset{\underline{u}\in\mathcal{U}}{\text{ARG MIN}} \; [(\underline{\lambda}(\tau)\underline{B} + \underline{\pi})^T\underline{u}].$$

5.3 Contributions

In this thesis we expand the areas of application of optimal control theory to include the important class of problems involving message routing in data communication networks. As in many previous applications, we find that this approach may provide significant advantages over previous techniques used to attack the problem at hand. In this case, we obtain the dynamic and closed-loop properties lacking in the queueing theory approach. The concept of applying optimal control theory to the dynamic routing problem was pioneered by Segall [1976] and this is the first work to provide a detailed solution to the optimal control problem. It is hoped that the success of this effort will stimulate further investigation into this approach so that its full potential may be realized and understood with respect to other techniques.

From the control theory point of view, it is commonly known that optimal feedback solutions are in general very difficult to determine. This is particularly true for problems involving state variable inequality constraints, where one can at best hope to find such solutions only for special problems. We have found a complete feedback solution to a meaningful linear optimal control problem with linear state and control variable constraints and consider this achievement to be a significant contribution to the theory of deterministic optimal control. Of particular interest is our novel combination of necessary conditions, dynamic programming and linear programming. In addition, we have developed a theory based on geometrical concepts in linear programming for the analysis of linear optimal control problems with linear constraints.

5.4 Suggestions for Further Work

We view this work as an initial investigation into the feasibility of applying optimal control theory to problems of dynamic routing in data communication networks. As such, we realize that many more questions remain unanswered than have been satisfactorily resolved.

The primary challenge is to gain an understanding of the method by which the backward constructive algorithm for general network problems may be implemented on the computer. From our arguments in Chapter 4 it is clear that the geometrical interpretation is a powerful tool in understanding the various properties associated with the algorithm. Although we know that the special properties of Chapter 4 do not hold in general, we feel that the geometrical interpretation may be used to identify such properties as do hold. This possibility is discussed in Section 4.5. Hopefully these properties may be adapted to a computer implementable algorithm.

An important area for significant advance is in the development of an efficient algorithm which computes all of the extreme point solutions to a linear program. As mentioned in Section 3.3.3.1 this is a classical problem in linear programming and to the author's knowledge there is no algorithm available which is efficient enough for our purposes. Perhaps a good place to start is by attempting to exploit the special structure of single destination networks to construct an efficient algorithm for that case. At any rate, the desirability of the backward constructive technique will be greatly enhanced upon finding a satisfactory solution to this problem.

Finally, of course, there remains to extend the feedback solution first to general deterministic inputs and ultimately to the most realistic case of stochastic inputs. It is clear that both of these problems are extremely difficult due to the state variable inequality constraints. Segall [1976] presents a sufficient condition of optimality in the stochastic case when the inputs are modelled as mutually independent jump processes with positive jumps only. However, the condition is expressed in terms of the partial differential equation of the optimal cost to go and solution by numerical integration is out of the question due to excessively high dimensionality. Therefore, alternate solutions must be investigated.

Appendix A

FINDING ALL EXTREME POINT SOLUTIONS TO THE CONSTRAINED OPTIMIZATION PROBLEM

Operation 1, part (a), of the backward constructive algorithm calls for finding all of the extreme point solutions to the constrained optimization problem. At a fixed instant of time this problem is the linear program (the time variable has been dropped):

$$\min z \tag{A.1}$$

where

$$z = \sum_{x_i^j \varepsilon \mathcal{I}_{p-1}} \lambda_i^j \underline{b}_i^j \underline{u} \tag{A.2}$$

subject to

$$\mathcal{U}' \left\{ \begin{array}{ll} D\,\underline{u} \leq C & \text{(A.3)} \\ \underline{b}_i^j \underline{u} = -a_i^j \qquad \forall\, x_i^j \,\varepsilon\, \mathcal{B}_{p-1} & \text{(A.4)} \\ \underline{u} \geq \underline{0} & \text{(A.5)} \end{array} \right.$$

where $\underline{b}_i^j$ is the row of $\underline{B}$ corresponding to x_i^j. Suppose we have utilized a standard linear program routine employing the simplex algorithm to find a single extreme point solution to (A.1)-(A.5). Call this solution $\underline{u}^*$. Then if we denote $z^* = \sum_{x_i^j \varepsilon \mathcal{I}_{p-1}} \lambda_i^j \underline{b}_i^j \underline{u}^*$, the hyperplane

$$z^* = \sum_{x_i^j \in I_{p-1}} \lambda_i^j \underline{b}_i^j \underline{u} \qquad \text{(A.6)}$$

is tangent to the convex polyhedron $\mathcal{U}'$ at $\underline{u}^*$ and every point contained in the face of tangency is a solution to (A.1)-(A.5). Consequently, in order to find all extreme point solutions to (A.1)-(A.5) we must find all vertices of the convex polyhedron:

$$\mathcal{U}'' \left\{ \begin{array}{l} \mathcal{U}' \\ \displaystyle\sum_{x_i^j \in I_{p-1}} \lambda_i^j \underline{b}_i^j \underline{u} = z^* \end{array} \right. \qquad \text{(A.7)}$$

Rubin [1975] demonstrates that the problem of finding all the vertices of an m-dimensional convex polyhedron can easily be formulated as the problem of finding all the edges of an m+1-dimensional convex polyhedral cone. Furthermore, Chernikova [1964],[1965] has presented algorithms for finding all the edges of a convex polyhedral cone represented by a set of homogeneous linear equations and inequalities. Bloom [1976] has programmed Chernikova's algorithm as a FORTRAN subroutine and reports on computational experience gained using it. He finds that for problems of reasonably large size the amount of computation and memory required to execute the algorithm is excessive. At the root of the problem is the extreme numerical sensitivity of the algorithm to the value of z*. If z* is slightly too large, then the hyperplane (A.6) does not intersect $\mathcal{U}'$, and hence $\mathcal{U}''$ is empty. If z* is slightly too small, the hyperplane (A.6) intersects $\mathcal{U}'$ below the desired face of tangency and therefore many extraneous extreme points may be generated. For these reasons Chernikova's algorithm appears unsuitable for our application.

Appendix B

EXAMPLE OF NON-BREAK AND BREAK FEEDBACK CONTROL REGIONS

We present here a simple example to illustrate the two types of feedback control regions. The network is shown in Figure B-1 and the cost functional is taken to be

$$J = \int_{t_0}^{t_f} (x_{12}(t) + x_{32}(t) + x_{23}(t))dt. \tag{B.1}$$

The $\underline{y}$-constraint figure for this network is presented in Figure B-2. We shall be interested in the operating points ① - ④ , and the coordinates are indicated in the order y_{12}, y_{32} and y_{23}. The controls corresponding to these points are listed in Figure B-3. The particular backward optimal trajectory which is depicted in Figure B-4 has x_{12} leaving boundary first backward in time, x_{32} leaving next backward in time and finally x_{32} leaves the boundary. The regions corresponding to controls ①,② and ④ are non-break feedback control regions. The region corresponding to control ③ (below the breakwall $x_{12} - 3x_{32} - \frac{3}{2}x_{23} = 0$) is a break feedback control region.

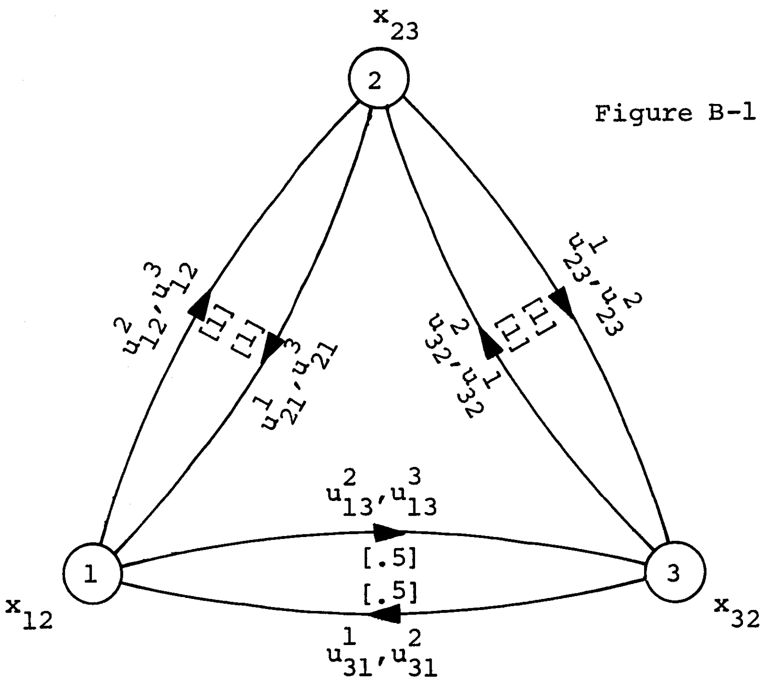


Figure B-1

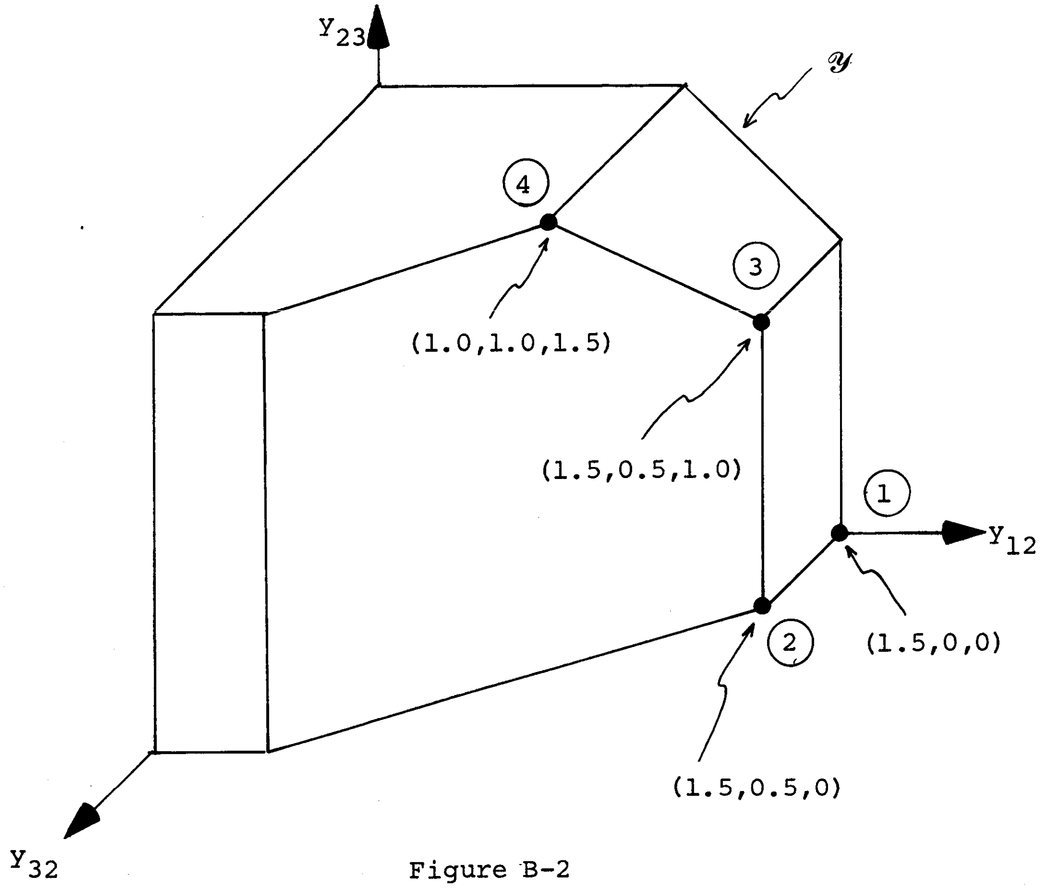


Figure B-2

	u^2_{12}	u^3_{12}	u^1_{21}	u^3_{21}	u^2_{13}	u^3_{13}	u^1_{31}	u^2_{31}	u^1_{23}	u^3_{23}	u^1_{32}	u^2_{32}
①	1	0	0	0	.5	0	0	0	0	0	0	.5
②	1	0	0	0	.5	0	0	0	0	0	0	1
③	1	0	0	0	.5	0	0	0	0	1	0	1
④	1	0	0	.5	0	.5	0	0	0	1	0	1

Figure B-3

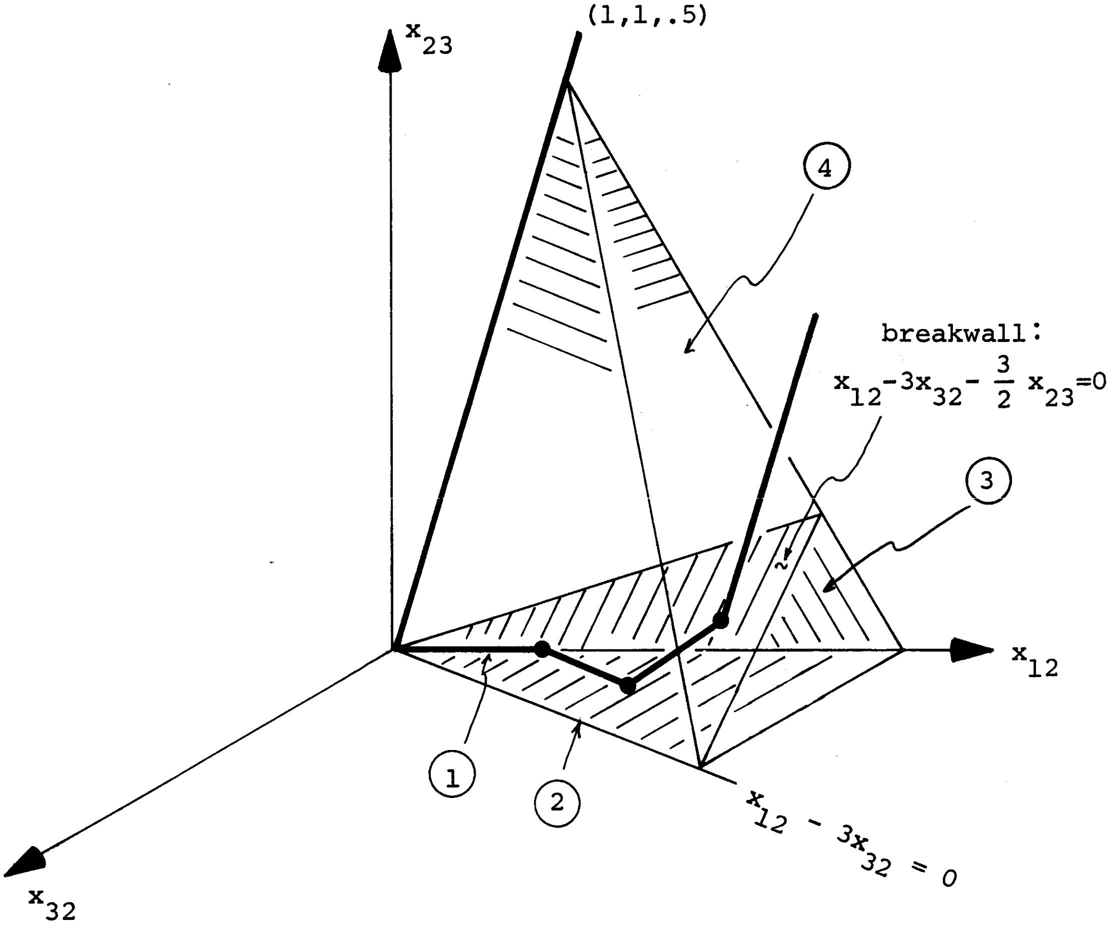


Figure B-4

REFERENCES

Balinski, M.L., "An Algorithm for Finding All the Vertices of Convex Polyhedral Sets", J. Soc. Indust. Appl. Math, Vol. 9, No. 1, March, [1961].

Berkovitz, L.D., "On Control Problems with Bounded State Variables", J. Math. Anal. Appl. 5, 488-501 [1962].

Bloom, J., "A Program for Chernikova's Algorithm", working paper, Electronic Systems Laboratory, M.I.T., March [1976].

Bryson, A.E., Denham, W.F., and Dreyfus, S.E., "Optimal Programming Problems with Inequality Constraints I: Necessary Conditions for Extremal Solutions", AIAA J. 1, No. 11, 2544-2550 [1963].

Cantor, D.G. and Gerla, M., "Optimal Routing in a Packet-Switched Computer Network", IEEE Transactions on Computers, Vol. C-23, No. 10, 1062-1069, Oct. [1974].

Chernikova, N.V., "Algorithm for Finding a General Formula for the Nonnegative Solutions of a System of Linear Equations", U.S.S.R. Computational Mathematics and Mathematical Physics, 4, 151-158 [1964].

Chernikova, N.V., "Algorithm for Finding a General Formula for the Nonnegative Solutions to a System of Linear Inequalities," U.S.S.R. Computational Mathematics and Mathematical Physics, 5, 228-233 [1965].

Cullum, J., "Discrete Approximations to Continuous Optimal Control Problems", SIAM J. Control 7, 32-49 [1969].

Dantzig, G.B., Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey [1963].

Denham, W.F., and Bryson, A.E., "Optimal Programming with Inequality Constraints II:Solution by Steepest Ascent", AIAA J. 2, No. 1, 25-34 [1964].

Gamkrelidze, R.V., "Optimal Processes with Restricted Phase Coordinates", Izv. Akad. Nauk SSSR, Ser. Mat 24, 315-354 [1968].

Gershwin, S.B., Private communication, M.I.T. [1976].

Ho, Y.C. and Bretani, P.B., "On Computing Optimal Control with Inequality Constraints", J. SIAM Control 1, No. 3, 319 [1963].

Horowitz, I.M., and Shaked, U., "Superiority of Transfer Function Design over State-Variable Methods in Linear Time-Invariant Feedback System Design," IEEE J. Auto. Control, 84-97, Feb [1975].

Jacobson, D.H., Lele, M.M., and Speyer, J.L., "New Necessary Conditions of Optimality for Control Problems with State-Variable Inequality Constraints", J. of Math. Anal. and Appl., Vol. 35, No. 2, 255-284, August [1971].

Kapur, K.C. and Van Slyke, R.M., "Cutting Plane Algorithms and State Space Constrained Linear Optimal Control Problems", J. of Computer and Systems Sciences, 4, 570-605 [1970].

Kleinrock, L., Communication Nets: Stochastic Message Flow and Delay, McGraw Hill [1964].

Kleinrock, L., Queueing Systems, Volume 2: Computer Applications, John Wiley & Sons, New York [1976].

Lasdon, L.S., Warren, A.D., Rice, R.K., "An Interior Penalty Method for Inequality Constrained Optimal Control Problems," IEEE Trans. Auto. Control, Vol. AC-12, 4, August [1967].

Manne, Ho, "Nested Decomposition for Dynamic Models", Mathematical Programming, p. 121, 6, No. 2, [1974].

Mattheiss, T.H., "An Algorithm for Determining Irrelevant Constraints and All the Vertices in Systems of Linear Inequalities", Operations Research, Vol. 21, No. 1, 247, Jan.-Feb. [1973].

Maurer, H., "On Optimal Control Problems with Bounded State Variables and Control Appearing Linearly", working paper, Mathematisches Institut der Universität Würzburg, [1975].

McShane, E.J., "On Multipliers for Lagrange Problems", Am. J. Math. 61, 809-839 [1939].

Pontryagin, L.S., "Optimal Control Processes", Uspekhi Mat. Nauk. 14, 3-20 [1959].

Rubin, D.C., "Vertex Generation and Cardinality Linear Programs", Operations Research, 23, 555-564 [1975].

Rustin, R., editor, Computer Networks, Prentice-Hall Inc., Englewood Cliffs, N.J., [1972].

Segall, A., "The Modeling of Adaptive Routing in Data-Communication Networks", IEEE Trans. on Comm., Spec. Issue on Comp. Comm. [1976].

Speyer, J.L., "Optimization and Control of Nonlinear Systems with Inflight Constraints", Ph.D. Thesis, Harvard U., Feb. [1968].

Valentine, F.A., "The Problem of Lagrange with Differential Inequalities as Added Side Conditions," Contributions to the Theory of Calculus of Variations, 1933-1937 (U. of Chicago Press, Chicago), 405-447 [1937].

BIOGRAPHY

Franklin Howard Moss was born April 20, 1949 in Baltimore, Maryland, son of Samuel and Rose Moss. He attended public secondary schools in Baltimore County and City, graduating from Baltimore Polytechnic Institute in June, 1967.

He matriculated at Princeton University in September 1967 where his major course of study was Aerospace and Mechanical Sciences. During the summer of 1969 he was employed by NASA as an engineering assistant at the Goddard Space Flight Center. He was an undergraduate member of the Aerospace Systems Laboratory and was employed there as research staff during the summer of 1970. In June 1971 he was graduated from Princeton with a B.S.E. degree, cum laude, and was elected to Sigma Xi.

Mr. Moss entered M.I.T. in September 1971 with a NASA traineeship in the Department of Aeronautics and Astronautics. He was awarded the S.M. degree in June 1972 with a thesis entitled "Nerva Injection and Cooldown Guidance for Planetary Orbit". He began studies in September 1972 toward the Doctorate in the Department of Aeronautics and Astronautics at M.I.T., concentrating in the areas of estimation and control. From June 1972 to June 1975 he was associated with the Charles Stark Draper Laboratory in the various roles of summer staff, research assistant and Draper Fellow. In June 1975 he became a research assistant in the Electronic Systems Laboratory, where he performed research leading to his dissertation in the area of data communication networks.

Mr. Moss was married to Jane May Levin of New York City in August 1972, and they have a son, Ilan Marcus, born September 3, 1975.