# OPTIMAL DISCRETE-LEVEL CONTROL

By

JAMES DAVID ENGELLAND

Bachelor of Science
University of Kansas
Lawrence, Kansas
1963

Master of Science
University of Kansas
Lawrence, Kansas
1964

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
July, 1970

Thesis
1970D
E57o
cop. 2

OKLAHOMA
STATE UNIVERSITY
LIBRARY
OCT 28 1970

OPTIMAL DISCRETE-LEVEL CONTROL

Thesis Approved:

Henry R. Sebesta
Thesis Adviser

Karl N. Reid

Lee Hannsberger

Charles M. Bacon

D. Durham
Dean of the Graduate College

763581

## ACKNOWLEDGMENTS


The total list of people to whom I owe thanks and appreciation for their help and encouragement during my doctoral program is too long to be enumerated here. However, I would like to list a few of these people who were of special importance:

Dr. H. R. Sebesta, who served as my major adviser. His interest, suggestions, and help on the dissertation were invaluable. He was never too busy to listen and give help.

Dr. K. N. Reid, who was my graduate committee chairman. It was Dr. Reid who, through friendly persuasion, helped me decide to return for my doctoral degree. Through his interest and concern, I received sufficient financial aid to complete my program.

Dr. E. L. Harrisburger and Dr. C. M. Bacon, the other members of my graduate committee. They both contributed significantly to my program by offering valuable criticism and guidance which was of great benefit to me.

My family, who deserve special mention. Sue, who shared every difficulty but always had enough faith, was always a patient listener and a willing helper. Our parents, who provided very real moral support, also deserve mention.

Finally, Miss Velda Davis and Mr. Eldon Hardy. Miss Davis prepared the manuscript and very ably served as an administrative assistant in processing it. Mr. Hardy made the excellent illustrations contained in the text of the dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

| Figure | | Page |
|---|---|---|

Figure | Page

# CHAPTER I

## INTRODUCTION

Modern technology has made high quality automatic control systems extremely important. Great strides have been made toward the realization of these "optimal or near optimal" systems through application of optimal control theory and the computer. However, there are two major shortcomings in the theory which must be eliminated before the control engineer can readily synthesize optimal or quasi-optimal controllers. These are:

1. Basic results of optimal control theory determine the optimal control ($U^*$) as a function of time (open-loop form). The control really is needed as a function of the plant state variables (closed-loop form) for easy implementation (see Figure 1). Although a digital computer can be used "on line" in the control system to solve the optimal control equations and determine the optimal control, it must do so rapidly compared to the system. (For example, the guidance computer used during the recent landing of the lunar module worked in this manner.) This means the computer must either be very fast or the plant relatively slow. Optimal control of this type is always expensive and for some fast systems impractical.

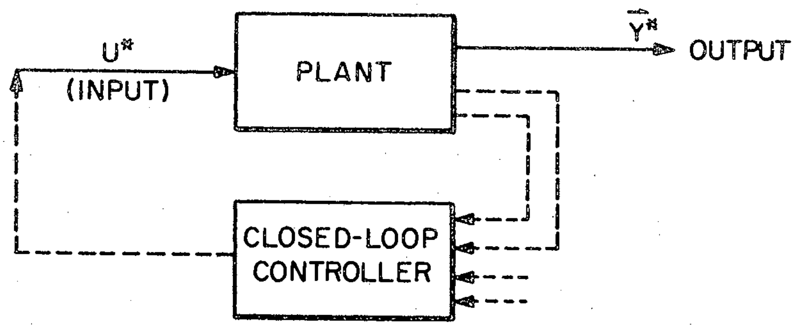



Figure 1. Basic Control System Schematic

2. If the control can be determined as a function of the state variables (presently possible only in special cases), it may still be extremely difficult to implement. The function may be so complicated that it is not realizable with hardware less sophisticated than a digital computer.

Clearly, there is an urgent need for new approaches to control synthesis which can either overcome these difficulties or avoid them. The intent of the present research was twofold: to propose and examine a new controller concept - Optimal Discrete Level Control; and to develop a controller synthesis method which avoids the aforementioned difficulties by use of the Optimal Discrete Level Control concept.

## Optimal Discrete Level Control

Controllers can be placed into two major categories according to the type of signal they provide to the plant; continuous and pulsed. Although most investigative work has been done with continuous controllers, a growing number of the pulsed type are being used. The pulsed controller most commonly used is the relay (bang-bang) type. Typical control signals which would be produced by several commonly used types of pulsed controllers (pulse-width, pulse-frequency, and pulse-amplitude) are shown in Figure 2. Notice the basic difference between pulse width and pulse-frequency signals is that the former has constant amplitude pulses with variable on-off ratios, while the letter does not.

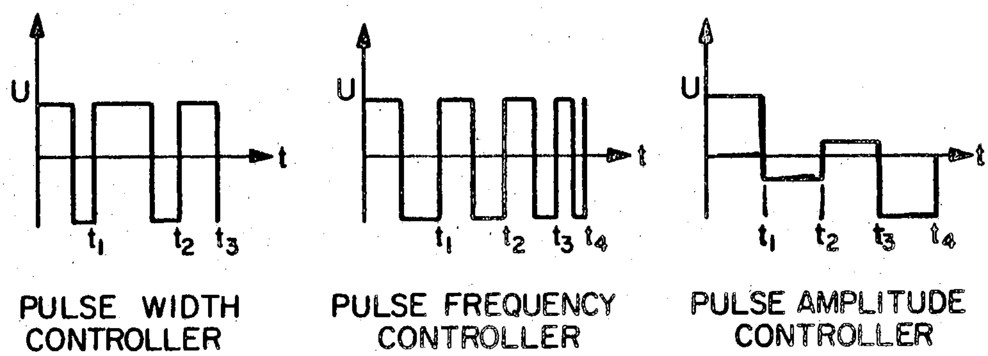



Figure 2. Typical Control Signals Produced by Pulsed Relay Controllers

A proposed new concept for switching controllers is that of optimal Discrete Level Control (hereafter denoted ODLC). This control philosophy can be defined by a single statement:

> The controller output (plant input) will be constrained to assume only specified discrete levels of amplitude and must switch between these levels in a sequence that minimizes a specified performance index.

Obviously, such a minimizing sequence will be optimal with respect to the performance index subject to the discreteness constraint. No other piecewise constant control can be found which produces a better performance. Figure 3 illustrates the type of control signal an ODLC controller with five specified levels might produce for some unspecified plant and performance index.

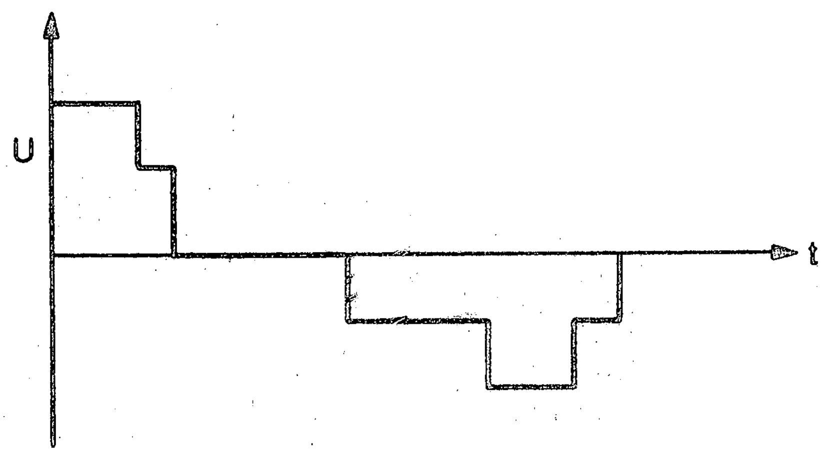



Figure 3. Typical Five Level ODLC Signal

This control sequence U (the output of the controller) is the input to the plant or system which causes it to behave in the desired optimal manner.

ODLC, as a concept, is new, but it should be pointed out that there are two familiar types of optimal controllers which could justifiably be redefined as being special cases of ODLC controllers. They are time-optimal and fuel-optimal controllers for linear and certain other plants.[1]

---

[1]These "certain other" plants are those in which the control signal enters linearly into the system equations.

These two special cases do not occur as a result of the control philosophy outlined above, but as a consequence of the special nature of their performance indices. In both special cases, the performance index itself causes the control to assume a piecewise constant nature to achieve optimal performance.

Time-optimal control, where the measure of optimality is the time needed to drive the system variables to some desired state, requires that maximal controller effort be exerted at all times. This means that only the "full positive" and "full negative" levels are used, and switching will occur as needed between these two levels, as illustrated in Figure 4. This is obviously a special case of two level ODLC.

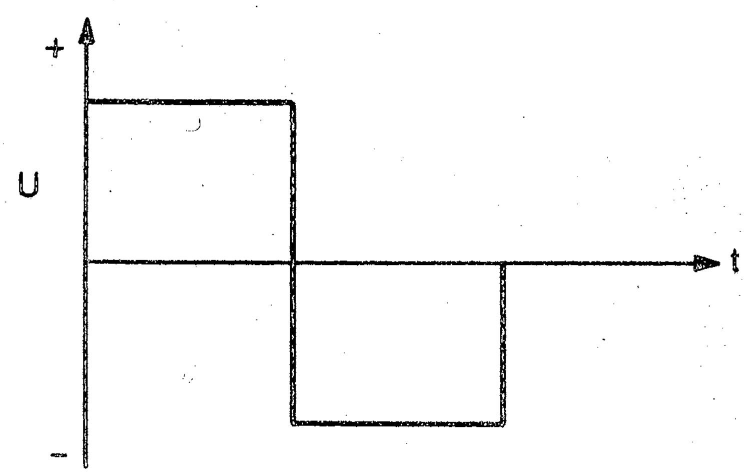



Figure 4. Typical Time-Optimal Control Sequence

Fuel-optimal control, where the measure of optimality is a linear combination of the time and effort (absolute value of U) needed to drive the system variables to a desired state requires three levels of control. These levels are full positive, full negative, and null (off). The resulting optimal control typically has the character illustrated by Figure 5. It also is a special case of ODLC, but of three levels rather than two.

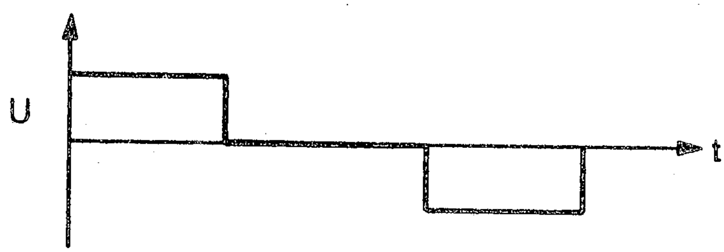



Figure 5. Typical Fuel-Optimal Control Sequence

The two above examples are ODLC because of their very special performance indices. But, it would be wrong to assume that all or even many pulsed controllers are ODLC. The very nature of pulsed controllers normally implies restrictions which make them no more than sub-optimal compared to ODLC. For example, fixed-rate sampled data controllers are inherently limited to constant pulse-width control signals which make them sub-optimal. Pulse-frequency (requiring the presence of a variable base frequency) and pulse-amplitude (requiring restrictions similar to fixed rate sampled-data) systems also cannot be classified as being ODLC because of their respective requirements just listed.

ODLC control signals can be shown to be functionally dependent only on the system variables.[2] This ODLC "control law" can be stated simply in terms of switching (hyper) surfaces in state space. Figure 6 is an example of the switching surface (curve) for time optimal control of a simple second order system. In Figure 6, the control will always be "full negative" if the system states are located right of the switching curve, and "full positive" on the left. Similarly, optimal switching surfaces will exist for all ODLC controllers. This is an important fact because it means that ODLC controllers can be operated closed-loop (functionally dependent only on the current values of the system states.)[3] Realization of an ODLC controller can, thus, be accomplished by hardware (or a computer) which functionally approximates these switching (hyper) surfaces.

At this point, several important characteristics of ODLC can be listed. First, since the ODLC controller is an optimal signal-producing device, its system controlling properties should be excellent. Second, in many cases, exact solutions to, or adequate approximations of ODLC switching surfaces will be very easy to achieve. Thus, ODLC synthesis may lead to a simple (and inexpensive) controller which optimally (for the approximations, near-optimally) controls its plant. A third important property of the Discrete Level Controller is noise immunity. Because of the quantized nature of the control signal control quality will be immune to all but extremely large levels of amplitude noise.

---

[2]This functional dependence is shown in detail in Chapter III.

[3]Contrast this to an open-loop control law where the input is specified solely as a function of time, and must be recalculated for each new operating condition encountered by the system.

This makes ODLC very attractive for use in noisy environments.

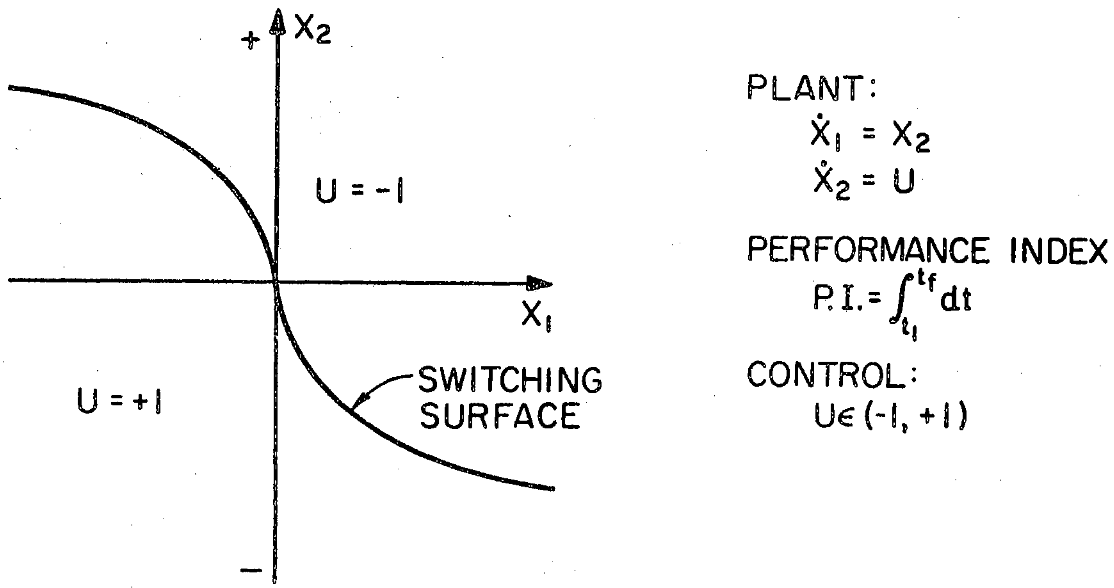



Figure 6. Optimal Switching Curve in State Space

A number of applications where the above would be useful can be envisioned but one special class of applications deserve special mention. This class is those systems in which the control servos are inherently discrete level. A prominent example of this would be the maneuvering thrusters on rockets and space vehicles which must operate in an on-off sequence. Clearly then, ODLC is an attractive type of system for space-age applications.

## Research Objectives

The thrust of the research was to explore some of the characteristics of the proposed concept for Optimal Discrete Level Control. The first goal was to establish the validity of using optimal control theory in finding the ODLC switching surfaces. The second goal was to determine the feasibility of implementing the optimal switching (hyper) surfaces or their approximations. Another goal was to develop synthesis methods suitable for design of systems using optimal discrete level control.

The scope of research was limited to investigation of the optimal (and near optimal) discrete level control of plants of the type

$$\frac{d\underline{X}}{dt} = \underline{f}(\underline{X}, \underline{U}, t),$$

where $\underline{f}$ is an $n \times 1$ vector function of the $\underline{n}$ component state vector $\underline{X}$ and the m control signals $\underline{U}$. The functions $\underline{f}$ are assumed to possess continuous partial derivatives with respect to $\underline{X}$ through the first derivative.

Specific research objectives were divided into two categories:

1. Investigation of the characteristics of discrete level controllers. Efforts in this area included:
   a. derivation of necessary conditions for ODLC sequences,
   b. determination of necessary restrictions to performance specifications for ODLC,
   c. definition of the types of systems in which ODLC can be used.

Work in this category was done to establish the basic control concepts.

2. Development of analytical methods for synthesis of discrete level controllers. There were two areas of primary concern here:
   a. development of computer methods to locate points on the optimal switching surfaces, (This turned

into an area of major effort since existing solution methods were determined to be inapplicable.)

b. investigation of appropriate surface fitting techniques and methods of selection of functional forms to be used for approximation of the switching surfaces.

This second area of study was to develop analytical procedure for the synthesis of closed-loop discrete level controllers.

## Topical Preview

The organization of this thesis roughly corresponds to the chronological order of research steps carried out in the investigation. Chapter II is a review of related work.

In Chapter III, validity of Pontryagin's Maximum Principle for obtaining necessary conditions for ODLC is verified. Special emphasis is given to determination of the characteristics of switching surfaces for both autonomous and non-autonomous systems.

Development of synthesis procedures for determining and approximating optimal switching (hyper) surfaces is carried out in Chapter IV. A discussion of several surface fitting methods is included.

Several example problems which illustrate some of the characteristics of ODLC surfaces and closed-loop controllers are detailed in Chapter V. Both linear and non-linear plants, and autonomous and non-autonomous systems are considered.

Chapter VI presents a discussion of principal results obtained from the investigation, particularly that covered in Chapter V.

Finally, Chapter VII summarizes the research and presents some pertinent conclusions.

# CHAPTER II

## LITERATURE SURVEY

Since the concept of Optimal Discrete Level Control as developed herein is new, there are no references in the literature dealing directly with the subject. There are, however, several papers which have dealt with subjects that are closely enough related to merit examination. In this chapter, the more significant of these are discussed.

### General Results on Optimal Pulsed Controllers

Kirk (11) has dealt with the problem of obtaining necessary conditions for optimal pulse-width modulation control. His basic approach is to view the system as a variable rate sampled data system. Calculus of variations is used to obtain these conditions. As with standard calculus of variations methods, he obtains not only algebraic necessary conditions but also a new set of differential equations, equal in number to the original plant equations, which must be solved simultaneously with the originals. As usual, the boundary conditions are split between initial and final time. The numerical technique used for solution of the resulting optimal control problem is an iterative steepest-descent algorithm. The most important aspect of Kirk's work is the implications contained in the problem definition. While Kirk restricts the plant inputs to $\pm 1$ and 0, he uses a rather general performance index. This means, in actuality, the systems he terms pulse-width-modulated are a special class of three level ODLC. There are several difficulties arising in Kirk's approach which limit its usefulness for ODLC.

1. The technique is limited to fixed final times - not free final times. (In all fairness, this shortcoming should be fairly easy to rectify.)
2. The numerical technique used was developed for three-level controls of the form $\pm 1, 0$. Because of this lack of generality, it would be difficult to satisfactorily extend the procedure to higher order cases.

Of considerable interest is the fact that the necessary conditions for the general ODLC problem developed in this paper reduce to those of Kirk for his special case. Since his problem is a special case of ODLC, this should occur, which helps verify the correctness of both analytical treatments.

In their paper, Nardizzi and Bekey (15) develop necessary conditions for optimal, combined pulse-width, pulse-amplitude control. On the surface, their optimal pulsed-controller work would appear to have direct bearing on Optimal Discrete Level Control since they also show that Kirk's pulse-width control is simply a special case of their problem. However, this is not the case, since they, in general, allow the input amplitudes to be infinitely variable. The only Discrete Level case to which their conditions can be applied is this special case of pulse-width modulation. The computational technique used by Mardizzi and Bekey is a gradient method implemented on the digital computer. (A method of this type was tried for ODLC, but proved to be unsatisfactory.)

Tou (23) deals with the twin problems of finding the optimum quantization levels for quantized (Discrete Level) control of sampled-data systems, and of implementing the quantized control through the use of switching surfaces. Two things distinguish Tou's work from the ODLC concept being investigated.

1. Tou restricts his systems to sampled-data systems with constant sample rates. An unfortunate consequence of this is that his analytical development is also restricted to constant rate sampled-data systems.
2. Also, as a result of considering only sampled-data systems, Tou requires a switching map for each sample interval. This means that for a system which will move to the desired final state in _n_ sample intervals, there must be _n_ switching maps. Figure 7 illustrates the type of switching surfaces required by Tou for a simple second order system.

A pertinent point discussed by Tou in his determination of optimal quantization levels is that optimal quantization can be accomplished only for a single trajectory (or a specific probability distribution which has been assigned to the possible initial conditions). Although optimal quantization was not investigated in this immediate research, it is certainly a topic for future interest. Further, the concept of establishing a probability distribution of initial conditions would seem to be the most appropriate method of pursuing the matter.

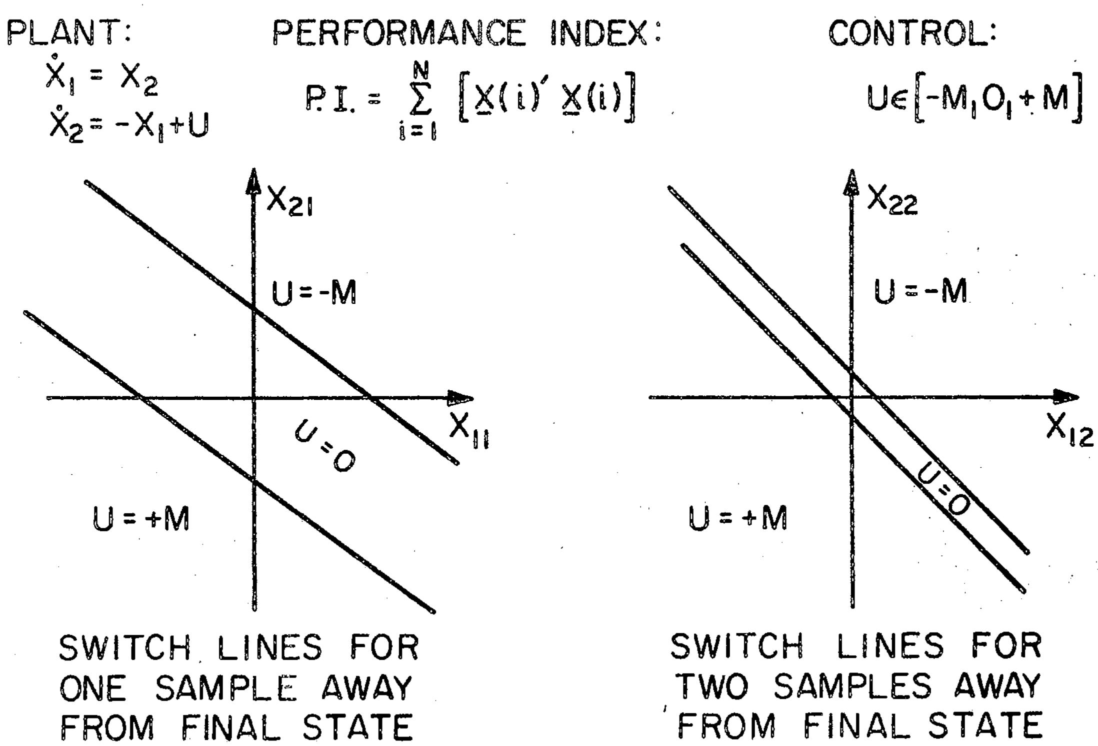



Figure 7. Typical Switching Lines Obtained by Tou for a Second Order Plant

## Approximation of Switching Surfaces

It is well known that the optimal control obtained by application of optimal control theory could be expressed simply as a function of the state variables. However, there is presently no closed form method of determining this functional relationship. For implementation of a closed loop control from optimal control theory, it is extremely important to have available suitable means of either determining or approximating this functional relationship. Although this relationship may be extremely complicated, it can be shown to be displayable as a set of switching (hyper) surfaces in state space for the case where switching occurs. Several investigators have developed methods to approximate these surfaces with implementable functions. The resulting equations express one state variable (the dependent variable) as a function of the remaining ones (independent variables).

The simplest method of implementation would be to use simple linear segment functions of the state variables. Frederick and Franklin (9) have investigated this case. The procedure outlined is a heuristic one in which an iterative search was initiated to discover a best combination of segments to use. Figure 8 shows two possible linear segment approximations. Each approximation would be tried and compared to the other on the basis of performance. All approximations would be iteratively tried and the best performing approximation chosen. The primary shortcoming of this approach is that, for higher order plants, there is no means of accommodating any cross-multiplying terms between the independent variables which means that none of the approximations might be very good.

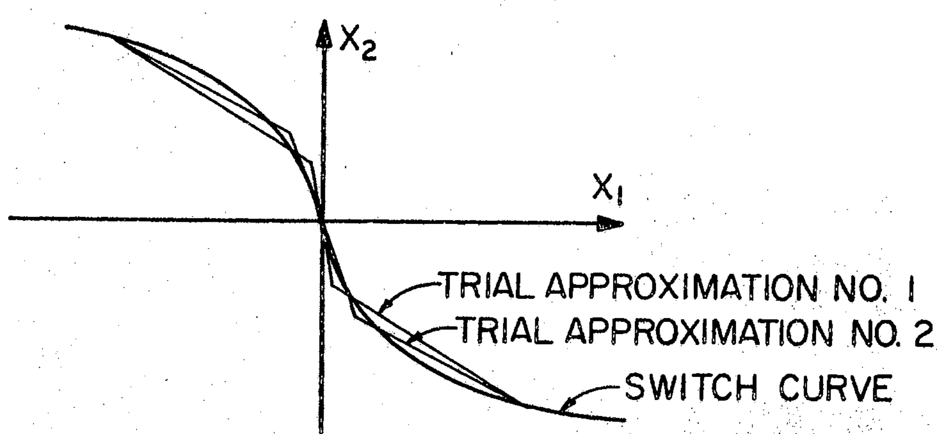



Figure 8. Typical Second Order Switching Curve With Two Possible Linear Segment Approximations

Smith (22) approaches the approximation problem from a slightly different viewpoint which is most attractive from an implementation standpoint. His procedure is to use a least squares fitting technique to geometrically obtain the best piecewise linear functional approximation to the surface.[1] He assumes the approximation functions will always consist of a number of piecewise linear functions of the state variables. The assumption of functions of state variable combinations enables the technique to approximate functions containing cross-products of (state) variables in a manner analogous to that used in "quarter-square" multipliers. The greatest benefit realized from the use of Smith's technique is the ease of implementation. All the piecewise linear functions can be implemented by means of simple diode-function generators. Controller implementation then consists of simply hooking these function generators together.

Geometrically, Smith's technique can be interpreted as approximating the optimal switching surface by planar triangular segments which are then joined together at their boundaries. The effect is much the same as approximating a sphere by a multitude of small flat triangular pieces appropriately cut and glued together.

Another attractive feature of Smith's procedure is that the technique is almost completely analytical and does not require precise visualization of the surface to be approximated. This means that the

---

[1]Obviously, a "best" geometric fit does not necessarily give the best approximation to the actual surface in terms of minimum performance index values. However, on an intuitive level, it would seem reasonable to expect better performance from approximations which are geometrically "close" than those which are "far" from the optimal surface.

technique is easily extended to high order systems where visualization is impossible. For the numerous reasons cited above, Smith's method was used as the approximation method in later chapters of this thesis.

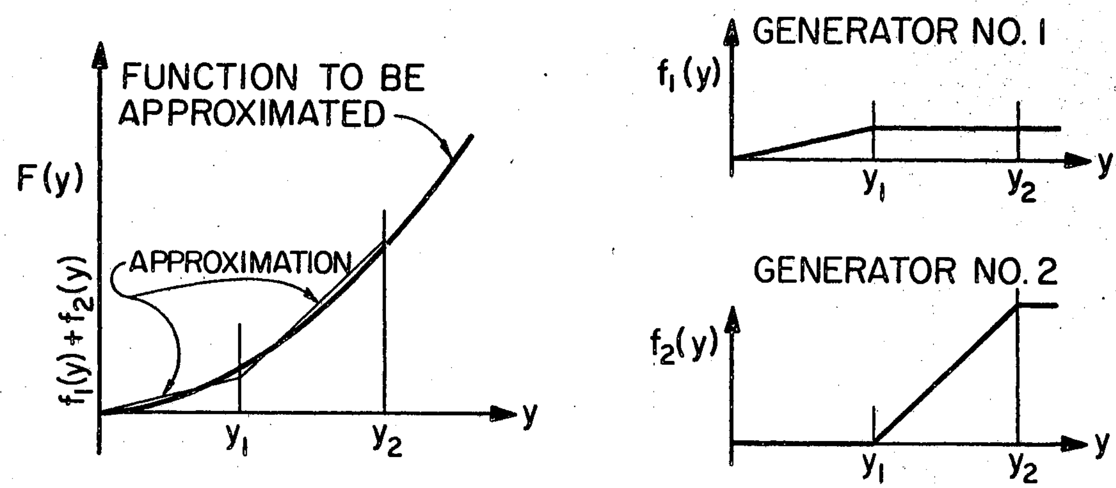



Figure 9. Example of Curve Approximation by Smith's Technique

A different approach to the approximation of the optimal switching surface has been adopted by Ibragimov (10). His procedure is to choose a series of "simple functions", (preferably orthogonal), and least squares - fit them to the optimal surface. The advantage to be gained by this technique over that of Smith is that it is possible to achieve a better controller than Smith's, without necessarily increasing the complexity of the controller. However, considerable engineering judgement would be required to choose appropriate "simple functions" which would both give "good" performance and be easy to implement. This approach, although it was not utilized in this dissertation, has great potential and must be earmarked as a topic suitable for further investigation.

## Computer Techniques

The set of split boundary-value differential equations which arise in optimal control theory are extremely difficult to solve. For a majority of cases, closed-form analytical solutions are not even possible. Since the advent of the computer, however, powerful numerical techniques have become available for solution of these problems. Usually, these procedures are limited to the solution of continuous equations with continuous inputs and are not suitable for solution of the problem posed by Discrete Level Control. A numerical technique which is eminently suitable for solution of DLC problems is Discrete Dynamic Programming. This technique, which is simply an optimal search technique, does not require continuous inputs (or plant equations) and, in fact, works better if they are not continuous.

Basically, Discrete Dynamic Programming is a numerical implementation of Bellman's "principle of optimality". This principle simply states that if a control is to be optimal, every segment of that control must also be optimal. For example, in Figure 10, if the four step control sequence is considered optimal (solid lines), then for Step 3, there can be no control (dotted lines) which is more optimal in that segment.

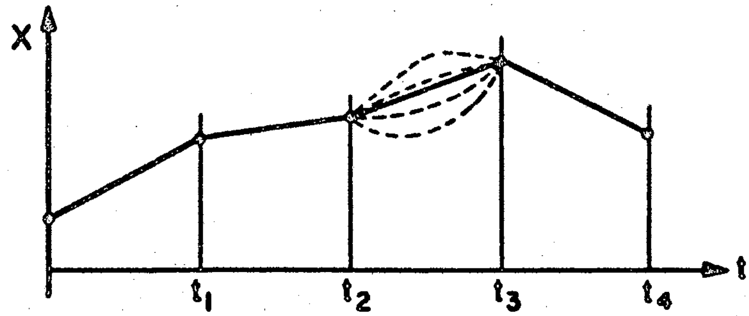



Figure 10. Illustration of the Optimality Principle

Although the basic idea seems quite simple, it can be used to build a tool for solving extremely difficult problems. This construction is accomplished by discretizing each state variable domain into a grid and searching each grid combination for the input(s) which transfer the states to the next discrete time step in a best manner. Appendix II discusses this procedure in detail. There is one computational difficulty which arises - the curse of dimensionality. To understand this difficulty, consider the following example:

The problem to be solved has a fifth ($n^{th}$) order plant. Also, because good accuracy is desired, each state variable domain is divided into a grid of 100 (N) grid points and 100 (M) time increments are used. The number of computer storage locations (P) will be

$$P = 2M \prod_{s=1}^{n} (N_s) = 2 \times 10^{12}.$$

This storage requirement far exceeds the capability of most computers, since computations must be made at each location.

Several people have done work in development of techniques which will allow reduction of these storage requirements. Two procedures which are of special interest are those of Davis (6) and Larson (13).

Davis (6) considers an iterative technique which reduces the total storage requirements from $2\,M \prod_{s=1}^{n} (N_s)$ to $2 \prod_{s=1}^{n} (N_s)$ which is a reduction by a factor of M. (M is the number of time divisions and $N_s$ is the number of grid divisions for each state variable.) The computational procedure for Davis' method is the same as for conventional dynamic programming except that instead of investigating all points in state space at M intervals in time, points in state space which are close to previously investigated points are investigated and the data for these points is iteratively improved until the optimal controls associated with these points are found.

As illustrated in Figure 11, the basic procedure of Davis' technique is to assign some guessed performance index (and input) value to each point in state space. Then, starting at the desired final point, improved values and input values are determined for the state space points, working outward from the final position step-by-step. By going through the performance index improvement procedure several times, the method will converge to a solution. Davis has worked a number of problems with the techniques and he reports reasonable accuracy. The major problem with the technique, however, remains the problem of dimensionality.

For the system quoted above, the computer storage requirement (computation must be carried out at each point at least once!) is still $2 \times 10^{10}$ locations.

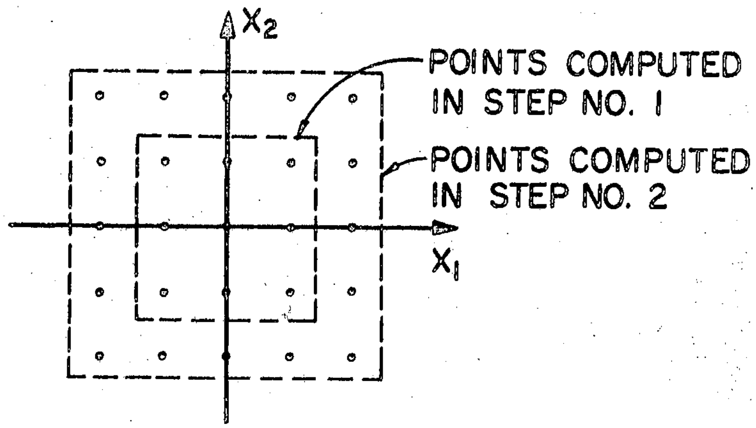



Figure 11. Iterative Dynamic Programming Technique Used by Davis (Shown in state space)

Larson (13) has developed a very attractive iterative computational procedure which can reduce the storage location requirements drastically. For the example problem, it could possibly require as few as 25,000 storage locations. The basic approach used is to partition state space into blocks (see Figure 12). The size of each block is determined by the number of grid points deemed desirable for each block. The time-interval spacing within the block is not fixed, but is set at the value such that the fastest changing state variable has changed by one state increment. This feature allows interpolation to be carried on at the (n-1) state level (rather than at the n), but adds an interpolation in time. Performance Index values at the trailing edge points of the block must either be estimated or supplied by computations from a previously worked block. The iterative process is that the solution is determined block-by-block until all blocks required for the problem solution have been worked. The technique is simply a procedure whereby the large problem is divided into some number of simple grid-block sub-problems.

Only those grid-blocks necessary for solution of the specific problem being considered are investigated. All other state space is ignored.

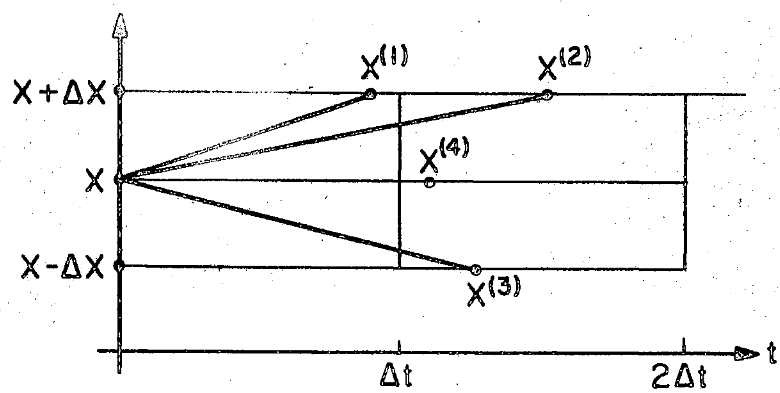



Figure 12. One Dimensional Example of Grid Blocks Used by Larson

Some of the more important features claimed for the method are:

1. Very large reductions in storage locations required for problem-solution are possible.
2. The iterative nature of the technique retains the generality of standard dynamic programming.
3. The large reduction of locations at which the algorithm must be applied makes the technique competitive time-wise with other solution techniques.

Although Larson's technique was not used in this investigation, it served as good background material for the method that was developed and used. This new technique which is outlined in Chapter III and developed in detail in Appendix II requires storage requirements similar to Larson's technique.

In summary of this chapter on related work, it is appropriate to state that no direct work on ODLC systems was discovered. However, the works cited provided the author with background information that was quite valuable for determining appropriate action which should be undertaken in the investigation of ODLC. With this background of information, the development of ODLC theory which is presented in the following chapter, can be placed in a proper perspective.

## CHAPTER III

## THEORETICAL CONSIDERATIONS

The objectives of the theoretical work were to determine necessary conditions for ODLC; to examine the characteristics of this control when displayed as switching (hyper) surfaces; and to establish some limits on the applicability of this type of controller. The only type of plants which were considered were those of the type

$$\frac{d\underline{X}}{dt} = \underline{f}(\underline{X}, \underline{U}, t) \tag{3-1}$$

where $\underline{X}$ is an $n \times 1$ vector, $\underline{f}$ is an $n \times 1$ vector function, and $\underline{U}$ is an $m \times 1$ vector.[1] It is assumed that both

$$\frac{\partial f_i}{\partial X_j} \quad \begin{matrix} i=1,n \\ j=1,n \end{matrix} \qquad \text{and} \qquad \frac{\partial^2 f_i}{\partial X_j \partial X_k} \quad \begin{matrix} i=1,n \\ j=1,n \\ k=1,n \end{matrix} \qquad \text{exist.}$$

The control vector is assumed to be restricted to a certain previously chosen set of Discrete Levels for each component of the input.

[1] Unless otherwise noted, $\underline{X}, \underline{f}, \underline{u}$, etc., are vectors of the appropriate order.

## Necessary Conditions

Pontryagin's Maximum Principle provides the theoretical basis for establishing necessary conditions for ODLC.[2] It was chosen as the most suitable method because no modifications to the basic statement of the Principle are required to make it valid for ODLC.[3]

In summary form, the necessary conditions as specified by the Maximum Principle are:

Given

A plant of the type $\dot{\underline{X}} = \underline{f}(\underline{X}, \underline{U}, t)$, (3-2)

and a performance index of the type

$$PI = \int_{t_1}^{t_f} G(\underline{X},\underline{U},t)\,dt \tag{3-3}$$

with the input components restricted to their respective desired magnitude levels. It is necessary that the optimal control (if it exists) will satisfy the condition that the Hamiltonian defined by $H = \langle \underline{P} \cdot \underline{X}' \rangle$, (3-4)

where

$$\dot{\underline{X}}' = \underline{f}(\underline{X}',\underline{U},t) = \frac{\partial H}{\partial \underline{P}} \tag{3-5}$$

and $\underline{P}$ is defined by

$$\dot{\underline{P}} = - \frac{\partial H}{\partial \underline{X}'}, \tag{3-6}$$

---

[2]A relatively short, heuristic derivation of the Maximum Principle is given in Appendix I.

[3]Pontryagin et al. (19) include in their definitions of admissible controls, the class of piecewise-constant controls. This obviously includes discrete level control.

must be a maximum with respect to the control $\underline{U}$.[4] Mathematically stated:[5]

$$H^* = \underset{\underline{U} \in U}{\mathrm{Max}}\, H. \tag{3-7}$$

Notice that although the statement of necessary conditions is quite simple, application of these conditions will require the solution of 2n simultaneous differential equations. Furthermore, the boundary conditions for the set of equations will be split. This is typical of optimal-control theory problems. Also, the optimal input can only be expressed as a function of various state plus adjoint variables or determined as a function of time (i.e., it is open-loop in character). Functionally, this can be written

$$\underline{U} = \underline{L}(\underline{X}', \underline{P}, t). \tag{3-8}$$

The desired functional relationship is the input expressed as a function of the state variables alone, or, expressed mathematically,

$$\underline{U} = \underline{L}'(\underline{X}).^{6} \tag{3-9}$$

Since Equation (3-9) does not contain the independent variable time (t) or the intermediate variables ($\underline{P}$), it is closed-loop in nature.

---

[4] $\langle \cdot \rangle$ denotes inner, or dot product. For example, where $\underline{A}$ and $\underline{B}$ are $n \times 1$ vectors, it implies $\sum^{n} A_i B_i$. $\dot{\underline{X}}$ implies the time derivative of the vector $\underline{X}$. $X'$ denotes that $\underline{X}$ has had the new component $X_{n+1} = G(\underline{X}, \underline{U}, t)$ added to it as the n+1 component.

[5] $\underline{U} \in U$ simply means that $\underline{U}$ must belong to the set of allowable inputs U.

[6] Actually, by defining a new state variable $X_{n+2} = t$, and adding this to the vector $\underline{X}'$ throughout the development of the necessary conditions, Equation (3-9) will still be valid for the case where $\underline{U} = \underline{L}'(\underline{X}, t)$.

(Control signals can be generated simply on the basis of the current values of the system variables.) The equation could be used as the relationship on which a simple optimal control could be based. It is a relatively easy task to prove that such a relationship does exist, but there presently is no analytical approach available for determining it (except for several special cases [17]).[7]

On an intuitive level, the existence of such a relationship is readily apparent when a switching control is considered. For example, consider the familiar instance of the switching surface of a time-optimal controller for a simple second order plant, as plotted in state space.

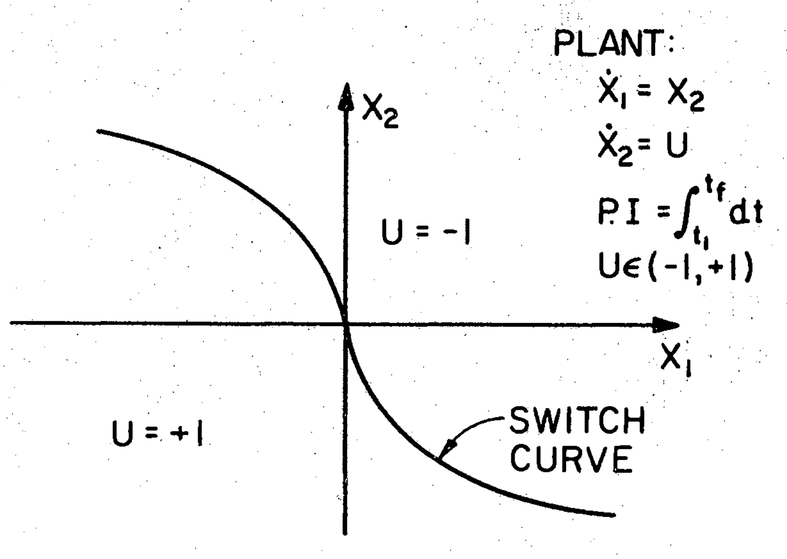



Figure 13. Typical Time-Optimal Switching Surface for a Linear Secondary-Order Plant

[7]A proof modeled after that of Rozonoer (20) will be presented later in this chapter.

This switching surface is precisely the functional relationship required between the input and state variables for optimal control and implementation of it will result in the desired closed-loop controller, as illustrated in Figure 14.

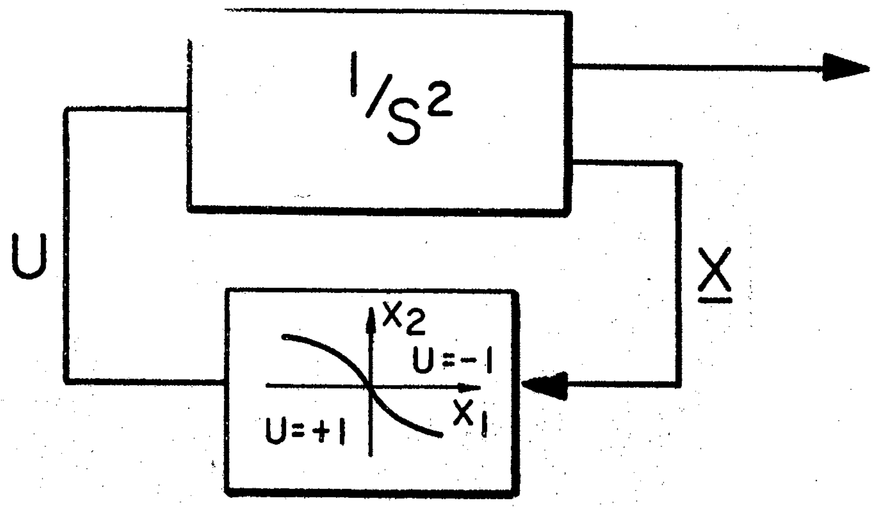



Figure 14. Typical Closed-Loop Time-Optimal Control System

The problem then becomes: "Is it always possible to define switching (hyper) surfaces and curves in state space which, when implemented, will give optimal closed-loop control?" and "If so, what are the characteristics of these surfaces?".

The first step toward answering these questions is verifying that switching points can be functions of the state variable alone. This is a fairly easy task and the development presented here follows closely that presented by Rosoener (20).

For the type of systems and performance indices under consideration the Maximum Principle can be used to determine

$$\underline{U} = \underline{\psi}(\underline{X}, P, t) \tag{3-10}$$

with fixed intial conditions $\underline{X}(T_o) = X_o$ there are some corresponding fixed intial values of $\underline{P}(T_o) = \underline{P}_o$ which satisfy the requirements for boundary conditions of the 2n differential equations and extremize the performance index. For each fixed set of initial values $\underline{X}_o$, there is a single set of values $\underline{P}_o$ which provides the optimal process.[8] Therefore, one can define a function

$$\underline{P}_o = \underline{\xi}(\underline{X}_o, T_o), \tag{3-11}$$

and at time $T_o$

$$\underline{U}(T_o, X_o) = \underline{\psi}(\underline{X}_o, \underline{\xi}(X_o, T_o), T_o). \tag{3-12}$$

However, if the time of operation is not fixed beforehand, this relationship must also hold true for arbitrary time or

$$\underline{U} = \underline{\psi}(\underline{X}, \xi(X, t), t). \tag{3-13}$$

An important exception must be pointed out here. If a portion of two or more different optimal trajectories is common to them both (or all), on that portion $\underline{P}$ is not uniquely determined by $\underline{X}$. However, since the trajectories of all are the same on that portion, $\underline{U}$ must be the same for

---

[8] This will not be necessarily true if $\underline{X}$ is located on a terminal trajectory. See the following paragraph of text for a definition of terminal trajectory and a complete explanation.

all cases and is determined uniquely by $\underline{X}$.[9]

Now, if the system equations have only constant coefficients

$$\underline{U} = \underline{\psi}(\underline{X}).^{10} \tag{3-14}$$

This leads directly to the conclusion that, for an optimal trajectory, each switching event of the input will be associated with a particular point in state space. Moreover, each collection of like switching events from a number of optimal trajectories must be associated with a corresponding set of points in state space. However, it still remains to be shown that the complete collection of points for each set of input switching events must form a surface or curve in state space rather than simply an infinite number of isolated points. Investigation of the nature of switching points will help shed some light on this problem.

Consider a point $\underline{X}$ in state space and consider only optimal trajectories in state space. Define $\underline{X}^*$ to be a point such that $\underline{U}^*$ associated with $\underline{X}^*$ is different from $\underline{U}$ for all $\underline{X}$ in some particular direction from $\underline{X}^*$. The Hamiltonian for $\underline{X}^*$ will be

$$H^* = H(\underline{X}^*,\underline{P}^*,\underline{U}^*). \tag{3-15}$$

For a point $\underline{X}'$ arbitrarily close to $\underline{X}$ in the direction mentioned, the Hamiltonian is

[9]Obviously, these common portions will remain common completely to the origin. The justification is that two or more optimal paths from a common point cannot occur except in singular cases. These common trajectories will be called terminal trajectories.

[10]Or they are reduced to an equivalent constant-coefficient form by defining $X_{n+1} \equiv t$ and expanding $\underline{X}$ accordingly.

$$H' = H(X', P', U'). \tag{3-16}$$

This could be written in terms of $X^*$, $P^*$, and $U^*$ as

$$H' = H(X^* + \delta X, P^* + \delta P, U^* + \delta U), \tag{3-17}$$

or expanding in Taylor series about $(X^*, P^*)$

$$H' = H(X^*, P^*, U^* + \delta U) + \frac{\partial H}{\partial X}\delta X + \frac{\partial H}{\partial P}\frac{\partial P}{\partial X}\delta X + \text{H.O.} \tag{3-18}$$

Now $\quad \frac{\partial H}{\partial P} = \dot{X} \quad$ and $\quad \frac{\partial H}{\partial X} = -\dot{P}.$

Substitution of these into (3-18) yields

$$H' = H(X^*, P^*, U^* + \delta U) + (-\dot{P} + \dot{X}\frac{\partial P}{\partial X})\ \delta X + \text{H.O.} \tag{3-19}$$

but since,

$$\left.\frac{\partial H}{\partial X}\right|_t = \frac{\partial H}{\partial X} + \frac{\partial H}{\partial P}\frac{\partial P}{\partial X} = 0, \tag{3-20}$$

$$\dot{X}\frac{\partial P}{\partial X} = +\dot{P}. \tag{3-21}$$

Use of (3-21) causes (3-19) to become

$$H' = H(X^*, P^*, U^* + \delta U) + (0)\delta X + \text{H.O.} \tag{3-22}$$

Now it was assumed that a $U'$ different than $U^*$ was required at $X'$ in the neighborhood of $X^*$ regardless of how small the neighborhood was chosen to be. Mathematically, this can be stated as $\delta X \to 0$ does not imply that $\delta U \to 0$ because it must remain at some different fixed level. If this is so, then allowing $\delta X \to 0$ changes (3-22) to

$$\left. H' = H(\underline{X}^*, \underline{P}^*, \underline{U}^* + \delta \underline{U}) \right|_{X' = X^*} . \tag{3-23}$$

But, it has already been stated that

$$H_{X^*} = H(\underline{X}^*, \underline{P}^*, \underline{U}^*) . \tag{3-24}$$

Obviously,

$$H(\underline{X}^*, \underline{P}^*, \underline{U}^* + \underline{\delta U}) = H(\underline{X}^*, \underline{P}^*, \underline{U}^*) . \tag{3-25}$$

This means that at any point defined as above, the Hamiltonian will be indifferent to whether $\underline{U}^*$ or $\underline{U}^* + \underline{\delta U}$ is used.[11]

This constitutes a contradiction to the basic premise that $\underline{U}^*$ was necessarily different than $\underline{U}$ for all points arbitrarily close in the direction of interest. Important conclusions can be drawn from the above development. First, if a point is to be associated with a uniquely valued input, it must be totally surrounded by points which require the same input for optimality or at least do not require one of different value. Also, any two points, however close, requiring inputs of different value must be separated by a point which is indifferent to the choice between those two values of input assigned to it. This point of indifference is called a switching point. The line of reasoning for this conclusion is to choose a switching point as $\underline{X}^*$ in the above development. Looking first in the one direction and then the other, it

---

[11]The previous development assumes that $\underline{P}$ is uniquely determined by $\underline{X}$. This is not true on terminal trajectories, and the development is not valid for points lying on them. However, all trajectories intercepting a terminal trajectory will require an input switch to remain on it. Thus, all points on a terminal trajectory must be switch points and the terminal trajectory, a switching surface.

becomes clear that $\underline{X}^*$ must display this characteristic of indifference. The final conclusion then is that each point associated with a unique input value must be totally surrounded by:

1. points associated with the same input value,
2. switching points, or
3. a combination of both.

As a result, it is proper to state that each region of state space associated with a particular input will be totally surrounded by switching surfaces (loci of switching points) which completely separate it from all other regions. (An exception would be where both the region and the surface extend to infinity. However, the region is still separated from all other regions by the switching surfaces.)

Switching surfaces can be placed into two categories depending on whether they do or do not occur along system trajectories (or loci of trajectories). The first type, which is more commonly encountered in present systems, consists of the terminal trajectory surfaces. The classic example of this type of surface occurs in the time-optimal control of a simple second-order plant, as shown in Figure 15.

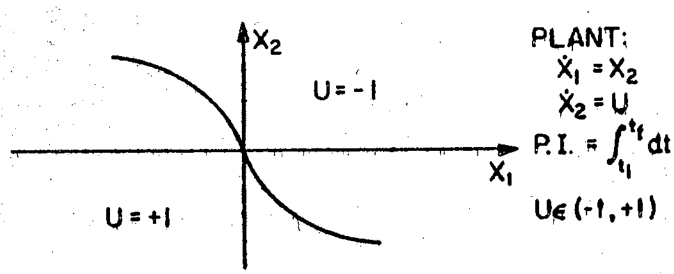



Figure 15. Time-Optimal Swithcing for a Simple Linear Second-Order Plant

In this case, the switching surfaces completely separate the two distinct regions of state space, and do so by extending to infinity.

For an example of a switching surface which is not simply a system trajectory, consider the case of time-optimal control of a linear second-order system with imaginary roots.

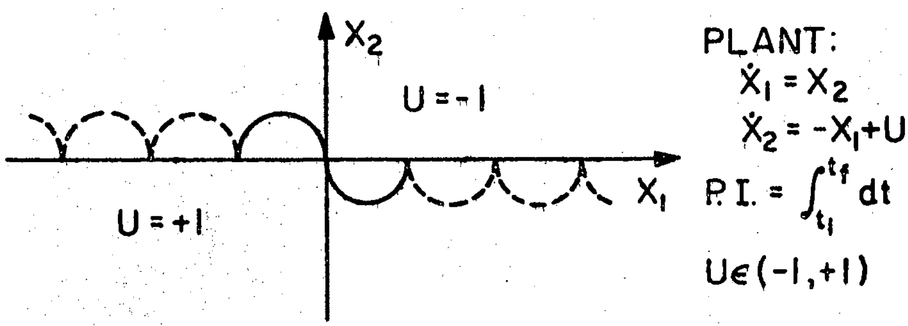



Figure 16. Time-Optimal Switching for Second-Order System With Imaginary Roots and No Damping

The switching curve consists of an infinite set of connected semi-circles. Only those two semi-circles immediately adjoining the origin are system trajectories. The others are simply loci of switching points.

Fuel-optimal control, as shown in Figure 17 illustrates that entire switching surfaces may be non-trajectory in nature.

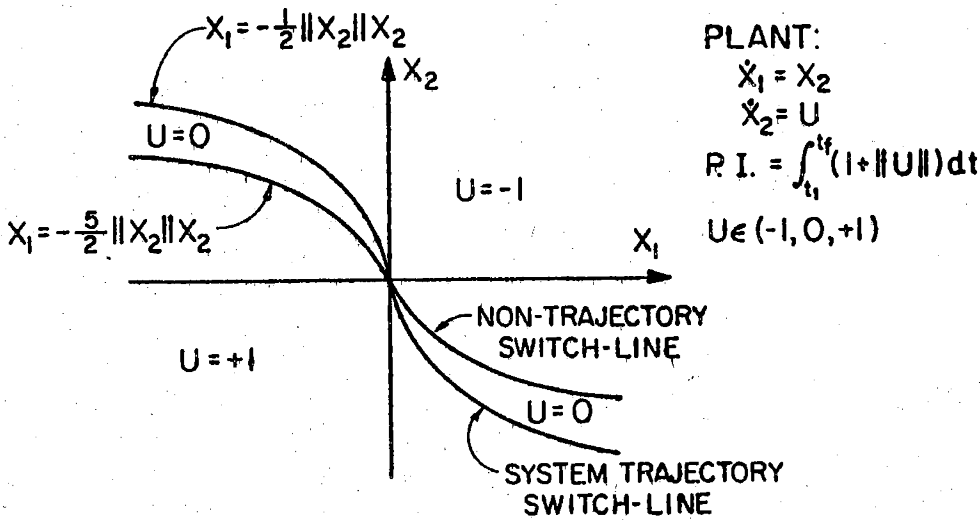



Figure 17. Fuel-Optimal Switching for a Simple Second-Order Plant

Notice that the one surface is not associated in any way with system trajectories and the other lies exactly along the terminal system trajectory. These simple illustrations will also be typical of the switching surfaces to be expected from general ODLC problems, although the switching surfaces will be more numerous and complex.

## Controllability

For certain systems, there are regions of state space from which no allowable input can drive the system to a given desired point in a finite time. These regions are referred to as regions of uncontrollability. Conversely, all other regions are ones in which the system is controllable. Cosniderable work has been carried on in attempts to analytically define general conditions under which a system will be controllable and/or stable. For linear systems with real positive roots, lack of controllability can be determined by evaluation of effects of the magnitude of these real positive roots. However, for the general case, there are no sufficiency tests available to determine controllability, although Lyapunov stability tests might provide helpful information.

A detailed study of controllability was considered to be beyond the scope of the present research. Indications of regions of uncontrollability will be discovered automatically during synthesis since the computer technique for locating optimal swithcing points will not converge in these regions.

## Non-Autonomous Systems

Implicit in all previous discussion of optimal switching surfaces is the concept of always driving the system to the origin of state space from arbitrary initial conditions. Systems which operate in this manner are commonly referred to as "regulators" and will not accept external reference inputs but must always operate in an autonomous manner. This is a serious limitation in practical applications. Fortunately, it is a limitation which is easily removed.

If an external reference input can be viewed as a variable desired steady-state value for a system state variable--then, subject to two basic requirements, reference inputs can be used in a controller with optimal swithcing surfaces. These requirements are:

1. Reference inputs can only demand changes in those state variables which are not fixed at specified values for the steady-state operating point.

Example 3-1. Consider the plant

$$\mathring{X}_1 = X_2$$

$$\mathring{X}_2 = U(t).$$

The steady-state point can be determined by setting the left side of the equation to zero,

$$0 = X_2$$

$$0 = U(t).$$

A steady-state value of $X_1$ has not been specified so a reference input $r_1 = x_{1d}$ which demands changes in the steady-state value of $X_1$ is allowable. The state $X_2$, however, must have a value of zero. Therefore, the input $r_2 = X_{2d}$ would be invalid since it violates this requirement.

2. The geometry of the switching surfaces must not be altered by the reference inputs. Allowable reference inputs have been defined as desired shifts in the location of the steady-state operating point. It is convenient to maintain this steady-state operating point at the origin of state space. This can be accomplished at any instant of time by a simple linear transformation of the coordinate system. If the shapes of the switching surfaces are not altered by this transformation, then direct implementation of the surfaces found for the autonomous case can be used for the non-autonomous.

Example 3-2. Consider the plant

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = U(t)$$

with the performance index

$$P.I. = \int_{t_1}^{t_f} dt$$

and the input

$$U \in [-1, +1].$$

This is the same time-optimal control problem discussed earlier. Figure 18 is a plot of the optimal swithcing surfaces for the autonomous case.

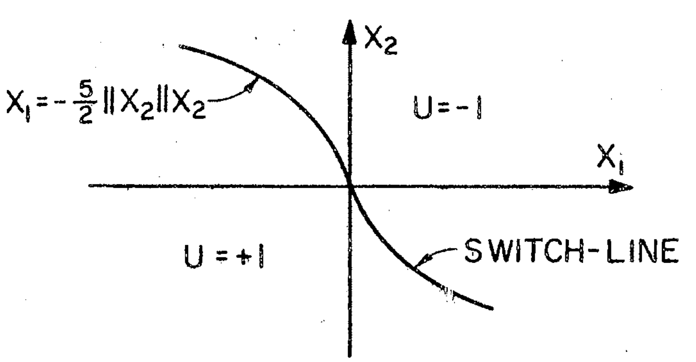



Figure 18. Optimal Switching Curve for Example 3-2

Closed loop implementation of this surface yields optimal autonomous control of the plant. Suppose, however, a reference input is desired. Obviously, the only valid reference input available is $r_1 = X_{1d}$. But, since this input will shift the switching curve along the $X_1$ axis, will the switching curves change shape because of the transformation? Checking this point is quite easy. If a linear coordinate transformation in the direction of interest can be made in the plant equation without altering the plant structure, then the surfaces are unaffected by the change. In the example being considered, the transformation $Y_1 = X_1 - X_{1d}$ introduced into the plant equations results in

$$\dot{Y}_1 = X_2$$

$$\dot{X}_2 = U(t).$$

Obviously, the plant characteristics are not a function of $X_1$ and $r_1 = X_{1d}$ is valid and can be used.

Plants with characteristics which change as a function of the value of the reference input will have switching curves which are altered by coordinate transformations. These plants are not directly compatible with non-autonomous controllers of this type.

Example 3-3. Given the plant

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = -X_1^2 + U(t)$$

and an arbitrary performance index, the first requirement for non-autonomous closed loop control will be satisfied by $r_1 = X_{1d}$. However, the coordinate transformation $Y_1 = X_1 - r_1$ will change the plant equations to

$$\dot{Y}_1 = X_2$$

$$\dot{X}_2 = -Y_1^2 + (-2Y_1 r_1) + (-r_1^2 + U(t)).$$

The plant equation now contains the new term $(-2Y_1 r_1)$ which cannot be lumped into the input term since it contains $Y_1$. Certainly, the shape of any switching curves for this plant would be a function of $X_{1d}$.

There is a convenient means of eliminating this difficulty, but it does increase the dimensionality of the switching surfaces by one. By defining the reference input as a new state variable with no dynamics, effects of changes in the reference input can be added into the plant characteristics. This alteration transforms the coordinate shift of the surfaces previously required to a change in the value of the new state variable. Consequently, the new switching surface with its extra coordinate will always contain the proper optimal switching information regardless of the input value. An alternate method of viewing this result is that a unique set of surfaces can be found for each desired value of the reference input. Since the shape of the surfaces is a function of the reference input, if it is used as a new variable, then all possible optimal switching surfaces will be defined in the expanded state space.

<u>Example 3-4.</u> Consider the plant

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = -X_1 \|X_1\| + U(t),$$

the performance index

$$P.I. = \int_{t_1}^{t_f} dt,$$

and an input constrained to

$$U \in [-1, +1].$$

Non-autonomous control of the system is not immediately possible because the plant equations are sensitive to the transformations in $X_1$. However, introduction of the new variable $X_{1d}$ changes the second-order optimal switching map to one of third order and makes non-autonomous control of the system possible. Figure 19 shows the optimal switching map for the example. (System coordinates have been arbitrarily restricted to $\pm 1$ (in value.)

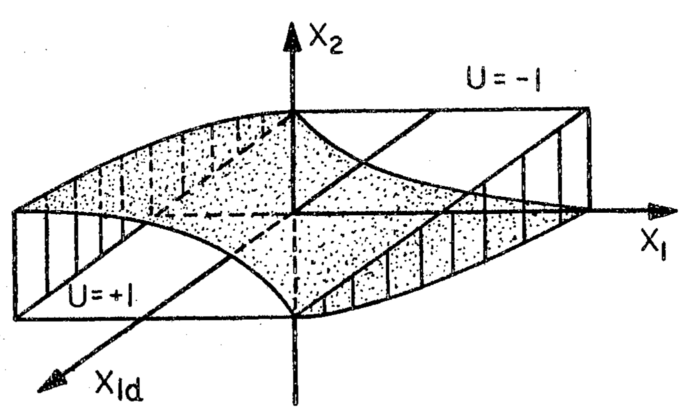



Figure 19. Time-Optimal Switching Surface for Example 3-4

implemented, the system schematic would be as in Figure 20.

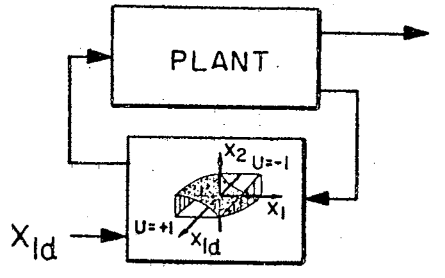



Figure 20. Non-Autonomous Control System for Example 3-4

## Time-Varying Plants

Another category of systems which are not directly suitable for closed-loop switching control is that of systems with time-varying plant parameters. Since the plant itself changes characteristics as a function of time, any optimal switching surfaces must do likewise. If, however, time is defined as a new state variable (increasing the optimal switching surface dimensions by one), the surfaces will contain all time-dependent information and will no longer be explicitly time-dependent. Thus, in a manner analogous to that for the non-linear plant, time-varying systems can be made suitable for closed-loop optimal-switching control.

Example 3-5. Consider the plant

$$\dot{X}_1 = X_2$$

$$\mathring{X}_2 = \frac{1}{m(t)} U(t)$$

with allowable inputs of

$$U \in [\pm 2, \pm 1, 0]$$

and the performance index

$$\text{P.I.} = \int_{t_1}^{t_f} (1 + \|U\|)\,dt$$

m(t) is a time-varying mass which is assumed to be measurable. The resulting switching surfaces, which are shown in Figure 21, are suitable for implementation for either autonomous or non-autonomous systems, since the plant is linear.

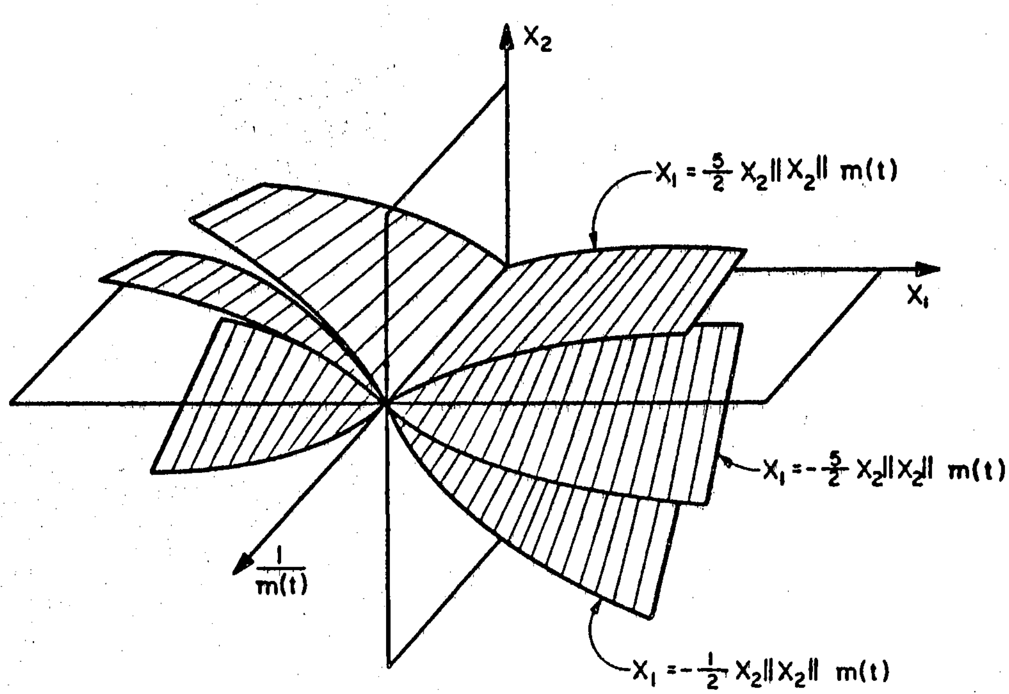



Figure 21. Optimal Switching Surfaces for Example 3-5

## CHAPTER IV

## SYNTHESIS TECHNIQUES

Before any control concept can be used in the design of actual hardware, computational techniques which can be used for synthesis must be developed. This chapter details a synthesis procedure developed for ODLC. In concept, the procedure is simple and straightforward. In practice, it can become a very difficult task. Basically, the procedure is comprised of three distinct steps.

First, the problem must be defined. This definition must include defining the plant equation (and boundary conditions), the performance index and the desired discrete input levels. Also, depending on the computational procedures used in step two below, definition of the necessary conditions for optimality (i.e., using the Maximum Principle) may be required.

Secondly, an adequate number of points on the optimal switching surfaces must be located to ensure successful description of these surfaces by a chosen approximation technique. For several reasons, this step is by far the most difficult task in the entire synthesis procedure.[1]

---

[1]As will be demonstrated in a later example, exact analytical determination of optimal switching surfaces can sometimes be obtained, but in general this is not true. The availability of such exact solutions not only demands a linear plant, but is highly dependent on the chosen performance index. In contrast, it will always be possible to locate any desired number of points on the surfaces. For this reason, the technique must be capable of synthesis through the use of these points.

Several different approaches to this task were attempted with varying degrees of success. These will be discussed later in the chapter, with emphasis upon a modified dynamic programming technique which was developed as the most generally applicable approach.

The third and final step is functional approximation of the optimal switching surfaces. During this step, the ODLC may (or may not) become sub-optimal. If no approximation is needed, the controller will remain optimal. If it is necessary or desirable to approximate the ODLC surface, sub-optimal control will obviously result. How nearly this sub-optimal control approaches the optimal will naturally depend on the quality of the approximation. This statement is made in light of the intuitive feeling that geometrically "close" approximations will give closer to optimal behavior than approximations which are less "close". Although it is possible to obtain approximations with any desired degree of quality, increasingly good approximations will usually require increasingly complicated approximating functions. It will be the task of the designer to determine a reasonable compromise between the optimality and the complexity of the controller. A number of standard approximation techniques such as least-squares-fitting of polynomials, Fourier Series, or Chebyshev polynomials are available. An alternate method proposed by Smith, (see Chapter II, and later in this chapter, is linear-segment approximation by least squares fitting). This technique is of primary concern in this thesis.

At this point, a complete three step procedure which can be used for synthesis has been outlined. Before any synthesis examples are presented, however, it is appropriate to review each step in detail and to examine both the problems encountered and the methods developed in each.

Step I - Problem Definition

It was always assumed that the plant equations, the performance index, and the desired input levels were previously specified. The essence of this step then is to determine the necessary conditions for optimality. This was done by use of the Maximum Principle. In working form, this procedure is as follows:

A. Construct the Hamiltonian function with the performance index included as a state variable. Thus, where

$$\dot{X}_1 = f_1(\underline{X},\underline{U},t)$$

$$\dot{X}_2 = f_2(\underline{X},\underline{U},t)$$

$$\vdots$$

$$\dot{X}_n = f_n(\underline{X},\underline{U},t)$$

$$\dot{X}_{n+1} = f_{n+1}(\underline{X},U,t) = \frac{d}{dt}(P.I.) \tag{4.1}$$

the Hamiltonian can be defined as

$$H \equiv P_1f_1 + P_2f_2 + \ldots P_nf_n + P_{n+1}f_{n+1}. \tag{4.2}$$

The adjoint variables $(\underline{P})$ are as yet undefined. However, by using the Canonic Equations of the Maximum Principle,

$$-\frac{\partial H}{\partial \underline{X}} = \dot{\underline{P}}, \tag{4.3}$$

a suitable definition is obtained. At this point, there exists a system of 2n+1 boundary conditions. The remaining boundary condition is obtained from the fact that $P_{n+1} = -1$ (see Appendix I). In practice, this fact not only provides the additional boundary condition, but is used to reduce the system to 2n differential equations (excluding the P.I.).

Example 4-1. Consider the plant,

$$\dot{X}_1 = X_2$$
$$\dot{X}_2 = U$$

with the performance index

$$P.I. = \int_{t_1}^{t_f} (1 + U^2)\,dt$$

and the input

$$U \in [+1, 0, -1].$$

It will be assumed that both initial and final boundary conditions are specified for both state variables. With

$$\dot{X}_{n+1} = 1 + U^2$$

added to the state variables, the Hamiltonian is

$$H = P_1X_2 + P_2U + P_3 + P_3U^2 \,.$$

The adjoint variables, defined by

$$-\frac{\partial H}{\partial \underline{X}} = \underline{\dot{P}}$$

are

$$\dot{P}_1 = 0$$
$$\dot{P}_2 = -P_1$$
$$\dot{P}_3 = 0.$$

Making use of the fact that $P_3 = -1$,

$$H = P_1X_2 + P_2U - 1 - U^2.$$

At this point, these four equations cannot be uniquely solved because they contain six variables. The necessary additional relationships will be provided by defining the control law.

B. Defintion of the Control Law.

The control law, which will be an algebraic relationship

$$\underline{U} = \underline{\psi}(\underline{X},\underline{P},t), \tag{4.4}$$

is determined by requiring that

$$H_{opt.} = \max_{\underline{U} \in U} H. \tag{4-5}$$

With continuous control signs, (4.5) can be carried out by equating $\frac{\partial H}{\partial \underline{U}} = 0$. For the ODLC case, those terms of H which contain $\underline{U}$, must be inspected to determine the relationship between $\underline{U}$ and $\underline{X}$, $\underline{P}$, and t which make those terms maximal.

Example 4.2. Consider the system described in Example 4.1. The Hamiltonian is $H = P_1X_2 + P_2U - 1 - U^2$, and the allowable inputs are $U \in [-1, 0, +1]$. The terms in the Hamiltonian which involve U are $P_2U$ and $U^2$. Therefore, the problem becomes one of determining U such that

$$P_2U + U^2$$

is maximized.

Since U is constrained to (+1, 0, -1), it is easy to verify that for

$$\| P_2 \| < 1 \quad , \quad U = 0,$$

and for

$$\| P_2 \| > 1, \; U = \text{signum } (P_2).$$

These two relationships are the required control law for ODLC. Several important conclusions can already be gleaned from the control law. Since $\dot{P}_2$ is constant $(P_1 = C)$, $P_2$ must linear in time or a sloped straight line. This means that U will never switch more than twice and, further, cannot switch from a control level and then later switch back to it. Thus, the only possible optimal control sequences would be:

$$(+1, 0, -1); \; (0, -1); \; (1);$$

$$(-1, 0, +1); \; (0, +1); \text{ or } (+1).$$

as time progresses from $t_1$ to $t_f$.

C. Evaluation of the Hamiltonian.

If the performance index has been specified over a fixed time interval, then there are now enough constraints on the necessary conditions to solve for the optimal control as a function of time. For the problems considered throughout this research, however, only those with free final times were used. There are several reasons for this. Not only is this case more difficult to handle analytically, but it includes most practical system applications (i.e., optimal regulators). With free final time, one extra constraining relationship will be needed to solve the 2n system of equations. This can be provided by evaluation of the Hamiltonian. It can be shown that, for problems with free final time, the Hamiltonian is identically equal to zero on an optimal trajectory (20). This relationship,

$$H = \langle \underline{\dot{X}}\ \underline{P} \rangle = 0, \tag{4.6}$$

completes specification of the necessary conditions.

Example 4.3. Again considering the problem in Example 4.1,

$$H = P_1X_2 + P_2U - 1 - U^2.$$

But H = 0. Therefore,

$$0 = P_1X_2 + P_2U - 1 - U^2.$$

If it is agreed that the final boundary conditions in the state variables are

$$X_{1_{t_f}} = 0$$

$$X_{2_{t_f}} = 0,$$

then

$$0 = P_{2_{t_f}} U_{t_f} - 1 - U_{t_f}^2.$$

By inspection, it is obvious that

$$U_{t_f} \neq 0$$

and $U_{t_f}$ must either be -1, or +1. From $0 = P_1X_2 + P_2U - 1 - U^2$, if

$U_{t_f} = -1$, then

$$P_{2_{t_f}} = -2,$$

and if

$$U_{t_f} = 1,$$

then

$$P_{2_{t_f}} = 2.$$

With this information, the choice of possible optimal input sequences are (+1, 0, -1), (0, -1), and (-1) for $P_{2_{t_f}} = -2$, and the negative of these for $P_{2_{t_f}} = +2$.

An additional piece of information can be gained from the evaluation of the Hamiltonian:

$$P_2 = \pm 2 \quad \text{whenever} \quad X_2 = 0.$$

If this is true, then any system which has $X_{2_o} = 0$ must have $P_{2_o} = 2$ for $X_1$ approaching 0 from the positive direction.

An additional note about the necessary conditions resulting from the Maximum Principle should be added. These conditions are necessary, but not always sufficient for optimality (8). Switching controls always involve a signum function of one or more of the state-plus-adjoint variables. These signum functions are indeterminate if their argument becomes null; that is,

$$\text{Signum (o)} = \text{Indeterminate.} \tag{4.7}$$

It cannot be assumed, even for linear plants, that the argument will not become null and remain there for finite periods of time. If it does so, the possibility arises of the existence of singular controls. This simply means that there may be not one but several optimal control sequences from a given point in state space, none of which is uniquely optimal. Notice, however, the Maximum Principle, although it will not identify such singular controls does indicate the possibility of such, whenever they might occur. This is a final argument in favor of always including complete development of the necessary conditions during synthesis.

### Step II. Generation of Optimal Switching Points

The essence of this step is to locate enough optimal switch points so that approximations to the ODLÇ switching surfaces may be obtained. This can be accomplished by making a number of solutions for the optimal control (as a function of time), each for a different set of initial conditions. Since it has been shown that switching events are functions of the state variables alone, each switching event on each trajectory locates a point in state space on its switching surface. Some arbitrary number of solutions will locate enough points on each surface to properly define its shape and location. This is graphically illustrated by Example 4.4.

Exampe 4.4. Consider the fuel-optimal problem of Example 4.1. The system of plant-plus-adjoint equations are:

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = U$$

$$\dot{P}_1 = 0$$

$$\dot{P}_2 = -P_2.$$

The boundary conditions and control law are taken as

$$X_1(0) = X_{10}$$

$$X_2(0) = 0$$

$$X_1(t_f) = 0$$

$$X_2(t_f) = 0$$

$$P_2(t_f) = 2$$

and

$$U = 0 \qquad \text{for } \|P_2\| < 1$$

and

$$U = \text{signum } (P_2) \text{ for } \|P_2\| > 1.$$

Repeated solution for a number of $X_{10}$ yields the following:

| Initial Condition $X_{10}$ | Switching Points | | | |
|---|---|---|---|---|
| | from -1/to 0 | | from 0/to +1 | |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 0.8335 | -0.577 | 0.1667 | -0.577 |
| 2.0 | 1.667 | -0.816 | 0.333 | -0.816 |
| 3.0 | 2.500 | -1.00 | 0.500 | -1.00 |
| 4.0 | 3.333 | -1.155 | 0.6667 | -1.155 |
| 5.0 | 4.1667 | -1.291 | 0.833 | -1.291 |
| 6.0 | 5.00 | -1.414 | 1.00 | -1.414 |
| 7.0 | 5.833 | -1.526 | 1.167 | -1.526 |
| 8.0 | 6.667 | -1.633 | 1.333 | -1.633 |
| 9.0 | 7.500 | -1.732 | 1.500 | -1.732 |

Plotted in $(X_1, X_2)$ state space, they form a sufficient number of sample points to accurately construct the ODLC surfaces.

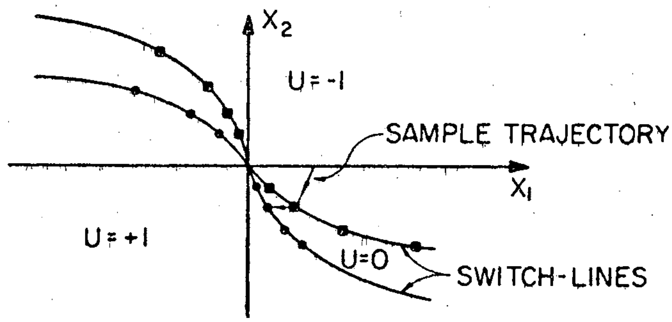



Figure 22. Construction of the Optimal Switching Surfaces for Example Problem 4.4

For this simple example, this solution approach was not really necessary because analytical solution for the surfaces can be directly made.

$$(X_1 = -\tfrac{1}{2}X_2 \, \|X_2\|, \text{ for } U \in 0/{+1};$$

and

$$X_1 = -\tfrac{5}{2}X_2 \, \|X_2\|, \text{ for } U \in -1/0).$$ [2]

This solution is accomplished by recognizing the fact that

$$P_2(0) = -2 \quad \text{and} \quad P_2(t_f) = +2.$$

[2] $U \in 0/{+1}$ signifies U changing from 0 to +1.

It then necessarily follows ($P_2(t)$ is a straight line) that

$$t_1 = t_3 = 1/2t_2$$

where

$t_1$ is the time interval during which $U = -1$,

$t_2$ is the time interval during which $U = 0$,

and $t_3$ is the time interval during which $U = 1$.

This relationship allows time to be eliminated from the equations and results in the switch curve equations. Direct solution for these switching equations was possible in this case because of its simple nature, but in general direct solutions are not possible. However, the technique of solution for switch points, which is always possible and provides valid switching information is completely independent of system complexity. Thus, it can always be used to determine the switching surfaces.

Repeated solution of the system equations to obtain switching points is, conceptually, a straightforward process. However, this class of equations is most difficult to solve. For all cases, the problem involves a set of differential equations with split boundary conditions and additional algebraic relationships to be satisfied. For the ODLC system, the additional difficulties of unknown switching times are added. These unknown intermediate switching times actually make the problem one composed of the sequential solution of a number of split boundary problems joined together through their interconnecting boundary conditions at these unknown times. This makes the problem even more complex.

During the research, four different approaches were attempted. A fifth, quasilinearization, was considered, but was determined to be unsuitable for the problem. The first two approaches tried were gradient methods.

The first of these two was to guess all unknown initial boundary conditions and, using these, to generate a trial solution from which improved boundary values could be obtained. This was not deemed to be a particularly suitable method because the equations (and results) were extremely sensitive to the initial condition values on the adjoint variables, making convergence very difficult. In fact, six place accuracy in these values was essential to get convergence.

Because of the instability of the adjoint equations in forward-time, a second approach in which these variables were integrated backward in time was tried. The scheme was to guess an initial solution, then, using this solution, to integrate the adjoint variables backward. By using these initial solutions, the optimal control law could be used to generate a new trial optimal control. This trial control was then used to generate a new trial solution, and the process was repeated until, hopefully, convergence was reached. Unfortunately, except for strictly non-oscillatory cases, convergence could never be assured. In fact, for oscillatory cases, the solution would invariably settle into an unacceptably large limit-cycle about the true solution. This method was also discarded.

The third method tried is simplicity itself: backward integration. This method proved especially suitable for second-order systems. The reason for this is that evaluation of the Hamiltonian will provide one constraining relationship at time $t_f$. In practice, along with the boundary conditions on the state variables, only one boundary condition value is unconstrained. Choosing a number of values to be used for this boundary condition will produce a corresponding number of optimal trajectories which is the desired end result. An undesirable characteristic of the method, however, is that the choice of boundary values for the free variable tends to be blind guessing; that is, it is difficult to predict how much a given boundary value change will change the optimal trajectory. Thus, a uniform search of state space can be difficult.

<u>Example 4.5.</u> Backward Integration

Consider Example 4.1. The system equations are

$$\mathring{X}_1 = X_2$$

$$\mathring{X}_2 = U$$

$$\mathring{P}_1 = 0$$

and

$$\mathring{P}_2 = -P_1.$$

Final boundary conditions are assumed to be

$$X_1(t_f) = 0$$

and

$$X_2(t_f) = 0.$$

It is also known from evaluation of the Hamiltonian that

$$P_2(t_f) = 2.$$

Now, if a new variable $\tau$ is defined by

$$\tau = (t_f - t),$$

then the system equations, in terms of $\tau$ are

$$\dot{X}_{1_T} = -X_2$$

$$\dot{X}_{2_T} = -U$$

$$\dot{P}_{1_T} = 0$$

$$\dot{P}_{2_T} = P_1.$$

The boundary conditions are

$$X_{1_T}(0) = 0$$

$$X_{2_T}(0) = 0$$

$$P_{2_T}(0) = 2.$$

The control law is

$$U = 0 \quad \text{for} \quad \|P_2\| < 1$$

$$U = \text{signum}\ (P_2) \ \text{for} \ \|P_2\| > 1.$$

A series of guessed values for $P_{1_T}(0)$ will generate a number of trajectories whose switch points lie exactly on the ODLC surfaces (not necessarily uniformly spaced). From these, a table such as in Example 4.4 can be constructed.

Backward integration becomes only marginal for third-order systems. The factor responsible is that evaluation of the Hamiltonian can provide only one additional relationship to the boundary conditions. Thus, there are now two unconstrained boundary conditions to be guessed. A guessing game then ensues with uniform spacing of optimal trajectories throughout the area of interest in state space being a virtual impossibility.

The fourth method used is one which evolved during the research. It is designed to be easily usable, even for high-order systems. It is a computer technique, based upon Dynamic Programming, but employs an iterative feature to reduce computer storage requirements. Although a complete description of the method, with detailed discussions of program features and limitations, is given in Appendix II, a brief conceptual description will be given here.

As a very brief background, Dynamic Programming is an optimal search technique. In the discrete version, which is utilized here, the problem can always be expressed in terms of finding the optimal solution to an n- stage (n increments in time) optimal decision process. Discrete Dynamic Programming converts this problem to that of finding the optimal solutions to a much larger-than-n number of single stage optimization problems. The single n-stage optimal solution will then be contained in this larger set of simpler solutions. It can be obtained by piecing together the proper sequence of optimal single stage solutions. This is called "embedding" and is an extremely powerful technique for the solution of difficult problems.

Practically speaking, Discrete Dynamic Programming is carried out in two steps. First, all state variables are discretized with the desired degree of fineness. In essence, this creates a grid network in state space, as shown in Figure 23.

Now, starting at a first trial final time ($n^{th}$ stage), all possible input combinations (also discretized) are tried at each possible combination of state variables ($(n-1^{th})$ stage) to find the best single stage input which satisfies final boundary conditions for each grid point. A performance index value is also computed for each grid point at this

$(n-1^{th})$ stage. This completes the $(n-1^{th})$ stage. The $(n-2^{th})$ stage is carried out in the same manner, except that the inputs will drive the variables to acceptable grid values for the $(n-1^{th})$ stage rather than to values which satisfy the final boundary conditions. Additionally, the performance index value which is used to determine the best $(n-2^{th})$ stage input from an $(n-2^{th})$ grid point will now consist of two parts. The first part will consist of the cost associated with the (n-2th) stage only. The second, is the cost associated with the $(n-1^{th})$ grid point to which the $(n-2^{th})$ stage input moves the system. This can best be illustrated by a one-dimensional example.

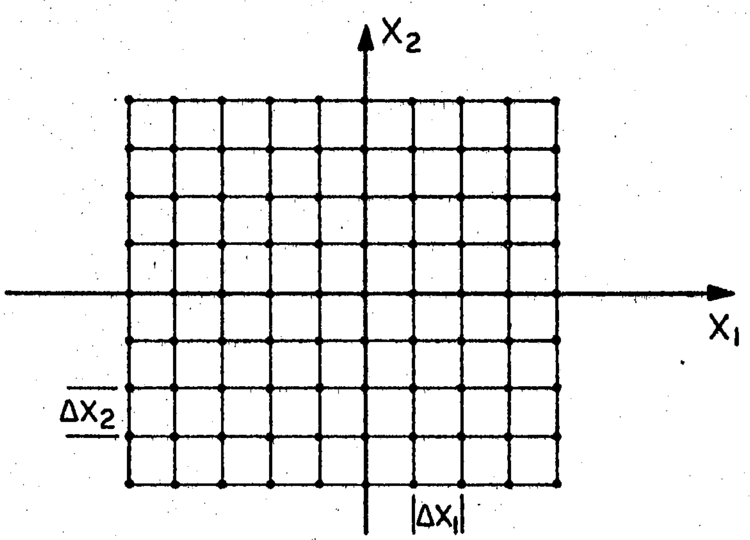
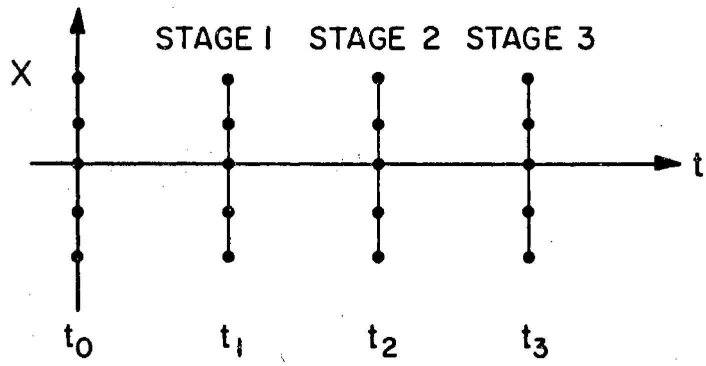



Figure 23. Division of Second-Order State Space Into a Grid

Example 4.6. Assume, for simplicity, that the single dependent variable is divided into a grid of five values, and time is divided into three increments, making the problem a three stage process. Pictorally, this is




Further, assume that the final condition on the variable (X) which must be satisfied at stage three is

$$X_{t_3} = 0.$$

Now, from each grid point at stage two, all allowable values of the input are tried and their corresponding cost values (performance index) computed. Assuming that there does exist at least one allowable input from each grid point at stage (2) which moves the system to $X = 0$ at stage (3), a best input for each point can be selected.

Moving back to the grid at stage (1), all allowable values of the input are again tried. This time, the inputs will be used to move the variable (X) from the stage (1) grid points to stage (2) grid points. Selection of the best inputs for each stage (1) grid point will this time involve more than just the cost of moving the variable to a stage (2) grid point. Already computed and associated with each stage (2) grid point is the cost included in the stage (2) to stage (3) process. This cost value associated with the stage (2) grid point <u>must be added</u> to the stage (1)-to-stage (2) cost function. It is on the basis of this total cost that best inputs are selected for stage (1). At this point, a pattern has evolved, and all intermediate single-stage operations are carried out in an identical manner until stage (0) is reached. Notice that, at stage (0), not only is the total performance cost of the complete optimal process for each initial value of X known, but that, if complete bookkeeping was maintained, the optimal input sequence is also known. The problem is solved.

The method developed during the research is designed primarily to drastically reduce the number of grid points over which the algorithm must search for the optimal trajectory. It is based upon the concept that, with a starting trial solution given, the search need only be carried out only in a certain constrained region of stage space around this trial solution. If a new and better solution is contained in this "tube" of state space surrounding the trial solution, this new solution will be used as the basis for constructing a new "tube" to be searched. This process is repeated until convergence is reached.

Since the iterative process can only move in the direction of increasingly optimal solutions, in concept, the process will always move to a local optimum at least. In practice, there are several difficulties which arise to complicate matters.

First, it is not always possible to guess a trial solution which will physically satisfy the boundary constraints on the problem. This means the "tube" may not contain any solutions which satisfy the boundary conditions and a search in the tube is meaningless. This difficulty was handled by making the search a two-phase process. The first phase is a search phase in which a very stiff artificial penalty is imposed for not satisfying the boundary conditions. Iterations are then made until the boundary conditions are satisfied. This normally requires several iterations. The second phase, then, is the actual iterative searching of "tubes" for the optimal control.

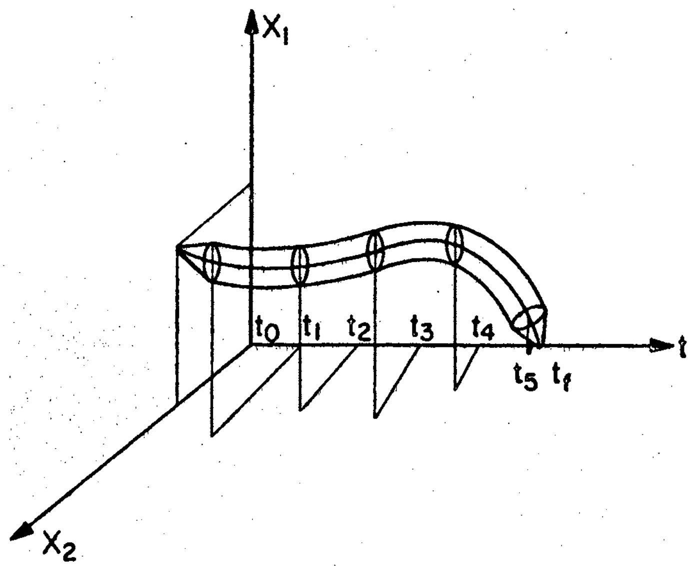



Figure 24. Illustration of the "Tube" to be Searched by the Iterative Method

Second, after a satisfactory "tube" is found, interpolations in time, as well as in the other variables, are necessary. With the system discretized, unless the time intervals are taken quite small compared to the size of the "tube", the algorithm may be unable to recognize the existence of more than one satisfactory solution in a given tube. Thus, either the time increments must be small or time interpolation must be carried out. The most straightforward solution would be to decrease the size of the time increments; but this leads into difficulty. If the time increments are very small compared to the diameter of the "tube", then even large input changes over this small increment of time will only slightly affect the state variables.

Thus, during the backward-working portion of the process, interpolations between widely spaced grid points to closely clustered groups of trajectory values will be necessary. (The $n-k$ stage integrations forward to the $n-k+1$ stage grid network will form a small cluster in state space) (see Figure 25). Since accurate interpolation of values for points in this small cluster from the widely spaces arid points is virtually impossible, a scheme of time interpolation for input switches was incorporated into the algorithm. This interpolation was accomplished by recomputing every time an input switch was indicated to determine the optimal switching time to within a very small time increment.

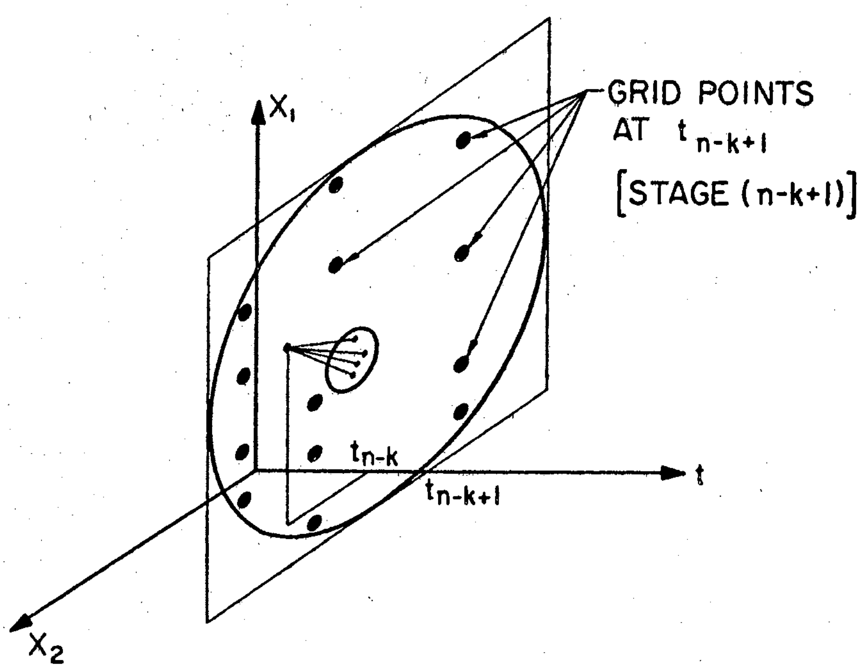



Figure 25. Comparison of "Large Diameter Tube" to Control Effects for Small Time Increments

Another difficulty which arose is also related to the size of the grid "tube" compared to the time increment length. In the algorithm, all input values which drive the system outside the "tube" are assigned some arbitrary high penalty. This is a necessary step because, outside the tube, space is "unexplored" and cost values cannot be accurately predicted. However, if the tube "diameter" is small compared to the length of the time increment, then all inputs except the one used to construct the last trial solution may drive the system out of the tube. A "small" tube cannot be tolerated. It is, consequently, necessary to assure the tube sizing will always be correct. A grid-scaling routine which sizes the tube "diameter" in accordance with the chosen time interval length and system input sensitivity was accordingly incorporated.

In light of the requirement for a starting solution for the iterative procedure, the algorithm was further modified to provide (if desired) its own starting solution. It finds this starting solution by use of regular dynamic programming over a very coarse grid with large time increments. For a detailed account of other features and programming details, refer to Appendix II.

## Step III. Approximation Techniques

The approximation technique which received primary emphasis during research was a least-squares linear-segment technique developed by Smith (9) for quasi-optimal minimum-time controllers. Advantages of the method are: it is analytic and does not require visualization of the surfaces; it does not require functional forms to be assumed; and it results in the specification of piecewise-linear functions.

In the technique, surfaces are always assumed to be of the form

$$X_i(\underline{X}') = \sum_{\substack{j=1 \\ j \neq i}}^{n} \hat{f}_j(X_j) + \sum_{k=n}^{m} \hat{g}_k(S_k) \tag{4.8}$$

where $(X')$ is considered to be a vector of all the state variables except $X_i$ which was chosen to be dependent. (Also n is the number of state variables contained by the plant, and m is n plus the number of cross-variables selected.) The $f_j(X_j)$ are each linear-segment functions of a single state variable $(X_j)$ alone. The second term in Equation (4.8), $(\sum_{k=r}^{m} \hat{g}_k\ (S_k))$ is intended to handle state variable cross-product effects. As such, each $\hat{g}_k$ is a linear-segment function of a single variable $(S_k)$ alone, in a manner completely analogous to the $\hat{f}_j$. However, in this case, the independent variable $S_k$ is an artificial variable composed of a linear combination of the independent state variables (sums, differences, etc.).

The approximation which results from the use of Equation (4.8) is a "quarter-square" type of approximation. The term "quarter-square" has a geometric interpretation which can be nicely illustrated for the three-dimensional case. Basically, the first term in Equation (4.8) attempts to approximate the surface with planar squares (or parallelopipeds) (see Figure 26). Obviously, for surfaces which are "warped", such an approximation is totally inadequate. This is the reason for the addition of second term in Equation (4.8). These $\hat{g}(k)$ terms effectively divide each "square" into four planar triangular pieces (thus, the term "quarter-square"). As Figure 27 illustrates, the resulting approximation is capable of handling "warped surfaces" very nicely.

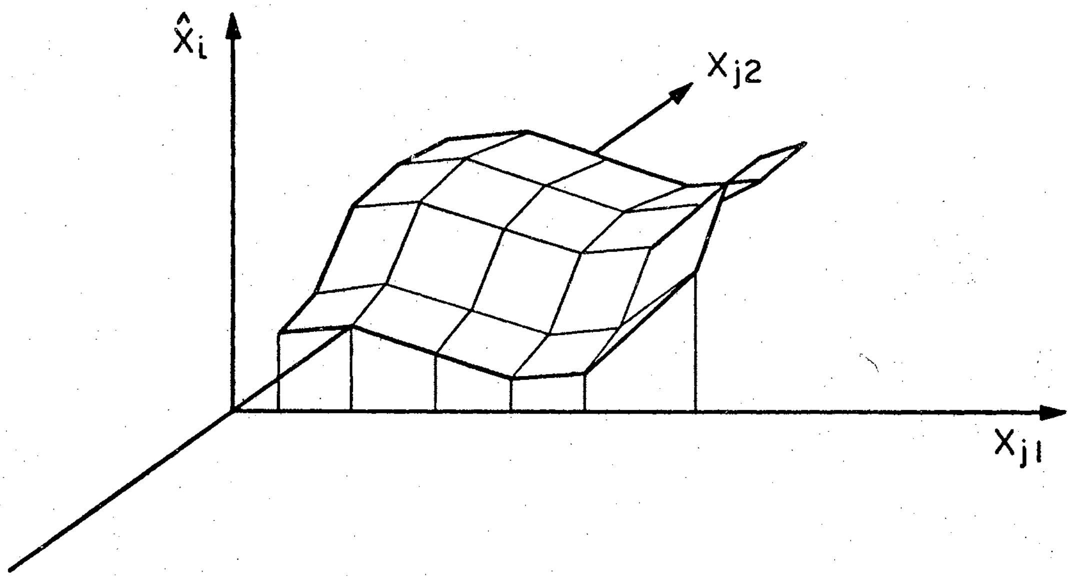



Figure 26. Linear-Segment Approximation to a Surface Using a Function of the Type $\hat{X}_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \hat{f}_j(X_j)$

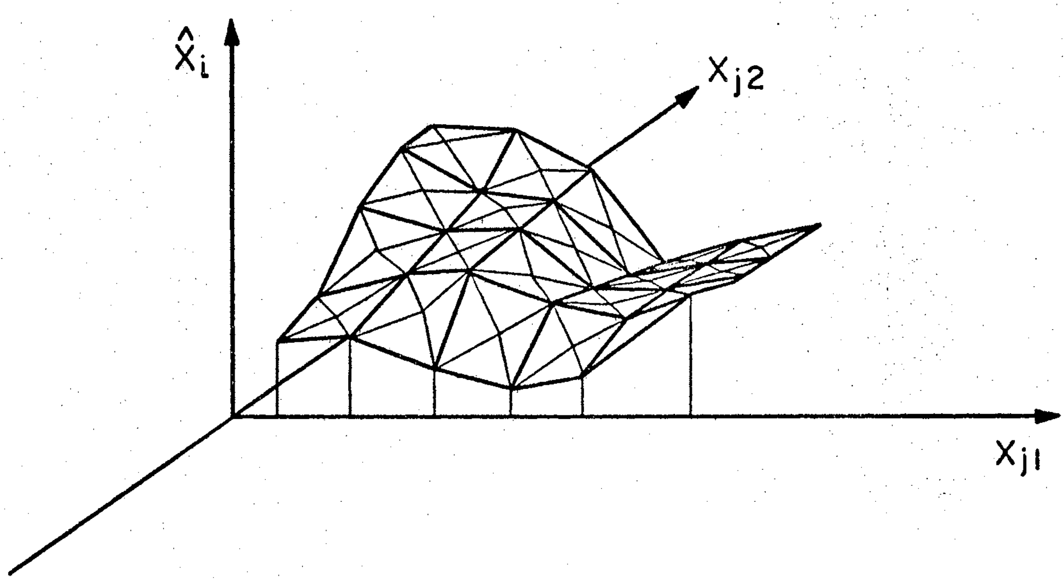



Figure 27. Linear-Segment Surface Approximation Using the Functional Form $\hat{X}_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \hat{f}_j(X_J) + \sum_{k=r}^{m} \hat{g}_k(S_k)$

To facilitate implementation, certain assumptions about the forms of $\hat{f}_j$ and $\hat{g}_k$ are made: the $\hat{f}_j$ and $\hat{g}_k$ are each composed of the weighted sum of some chosen number of linear-segment non-linear functions.

$$\hat{f}_j(X_j) = W_{j1}Y_{j1}(X_j) + W_{j2}Y_{j2}(X_j) + \dots \tag{4.9}$$

and

$$\hat{g}_k(S_k) = W_{k1}Y_{k1}(S_k) + W_{k2}Y_{k2}(S_k) + \dots . \tag{4.10}$$

Each linear-segment non-linear function will be of the form shown in Figure 28.

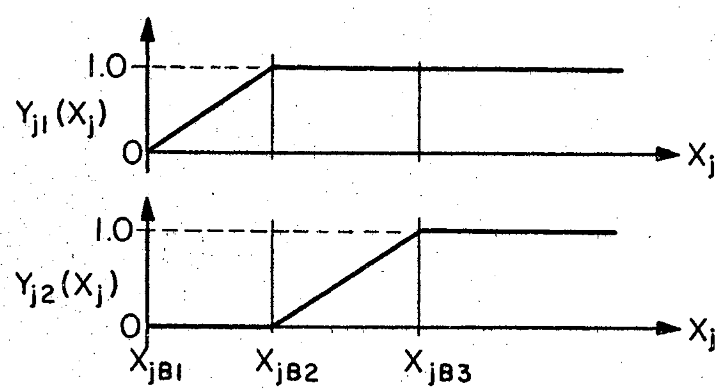



Figure 28. Linear-Segment Nonlinearities

Equation (4.8) can now be written in vector form as

$$\hat{X}_i(X') = \underline{Y}^T\underline{W}, \tag{4.11}$$

where

$$\underline{Y}^T = [Y_{11}(X_1), Y_{12}(X_1),\ \ldots,\ Y_{21}(X_2), Y_{22}(X_2),\ \ldots]$$

($X_i$ not included)

and

$$\underline{W} = \begin{bmatrix} W_{11} \\ W_{12} \\ \cdot \\ \cdot \\ \cdot \\ W_{21} \\ W_{22} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}.$$

Remember that each $W_{jk}$ is an unknown coefficient to be found. It is also useful here to review the meaning of the vector $\underline{Y}$. Each component of $\underline{Y}$ (for example $Y_{ji}(X_j)$, shown in Figure 28, has a region of its independent variable ($X_j$) over which it has zero value. It also has a region over which it has a value of one. Between these regions is a proportional range. If the extremities of this proportional region are called break points, then

$$Y_{jk}(X_j) = \frac{X_j - X_{jB_k}}{X_{jB_{k+1}} - X_{jB_k}} \qquad (4.12)$$

where $B_k$ signifies the lower value and $B_{k+1}$, the upper. Now the sum of the errors squared can be written in terms of these relationships as

$$E^2 = \sum_{q=1}^{P} [Y_q^T \underline{W} - X_i^*(\underline{X}_q')]^2 \qquad (4.13)$$

where q is the index of the data points used and P is the total number of data points. $X_i^*(\underline{X}_q')$ signifies the data value of the dependent state variable corresponding to data set q.

In order to minimize the sum of the errors squared for least squares fitting, the partial derivative of Equation (4.13) with respect to $\underline{W}$ is taken and then set equal to zero. That is,

$$0 = \frac{\partial E^2}{\partial \underline{W}} = 2 \sum_{q=1}^{P} \underline{Y}_q \underline{Y}_q^T \underline{W} - 2 \sum_{q=1}^{P} \underline{Y}_q X_i^*(\underline{X}_q'). \tag{4.14}$$

Rearranging Equation (4.14) gives the result

$$\underline{W} = [\sum_{q=1}^{P} \underline{Y}_q \underline{Y}_q^T]^{-1} \sum_{q=1}^{P} \underline{Y}_q X_i^*(\underline{X}_q'). \tag{4.15}$$

This equation is the necessary relationship that determines the unknown coefficients $\underline{W}$, provided certain fundamental decisions on the part of the designer have been made. These decisions are:

1. The state variable which will be considered dependent must be chosen.
2. The number and composition of the S variables (linear combinations of the independent state variables) must be chosen.
3. The number and spacing of partial functions (linear-segment nonlinearities, or $Y_{jk}(X_j)$, must be chosen. The extremeties of these are the function break points.

Example 4.7 illustrates how these three decisions and Equation (4.15) can be used to approximate an optimal switching surface.

Example 4.7. Linear Segment Approximations. Consider the data points for the 0 to +1 surface generated and listed in Example 4.4. They are

| Initial Condition X.0 | | Switch Points | |
|---|---|---|---|
| | | $X_1$ | $X_2$ |
| 0.0 | | 0.0 | 0.0 |
| 1.0 | | 0.1667 | -0.577 |
| 2.0 | | 0.333 | -0.816 |
| 3.0 | | 0.500 | -1.00 |
| 4.0 | | 0.667 | -1.155 |
| 5.0 | | 0.833 | -1.291 |
| 6.0 | | 1.00 | -1.414 |
| 7.0 | | 1.167 | -1.526 |
| 8.0 | | 1.333 | -1.633 |
| 9.0 | | 1.500 | -1.732 |

Assume that $X_1$ will be selected as the dependent variable, and that break points will be located at $X_2 = 0$; $X_2 = -1.00$; and $X_2 = -1.732$. There will be no S variables since the relationship has only one independent variable. Now,

$$\underline{Y}_1 = \begin{bmatrix} -.577 \\ 0 \end{bmatrix}, \quad \underline{Y}_2 = \begin{bmatrix} -.816 \\ 0 \end{bmatrix}$$

$$\underline{Y}_3 = \begin{bmatrix} -1.00 \\ 0 \end{bmatrix}, \quad \underline{Y}_4 = \begin{bmatrix} -1.00 \\ -.155/.732 \end{bmatrix},$$

$$\underline{Y}_5 = \begin{bmatrix} -1.00 \\ -.291/.732 \end{bmatrix}, \quad \underline{Y}_6 = \begin{bmatrix} -1.00 \\ -.414/.732 \end{bmatrix},$$

$$\underline{Y}_7 = \begin{bmatrix} -1.00 \\ -.526/.732 \end{bmatrix}, \quad \underline{Y}_8 = \begin{bmatrix} -1.00 \\ -.632/.732 \end{bmatrix},$$

$$\underline{Y}_9 \quad \begin{bmatrix} -1.00 \\ -.732/.732 \end{bmatrix}, \quad \underline{Y}_{10} = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}.$$

Substituting these into the first expression on the right-hand side of

Equation (4.15) gives:

$$\sum_{q=1}^{P} \underline{Y}_q \underline{Y}_q^T = \begin{bmatrix} 8.00 \\ 3.759 \end{bmatrix} \begin{bmatrix} 3.759 \\ 2.788 \end{bmatrix}.$$

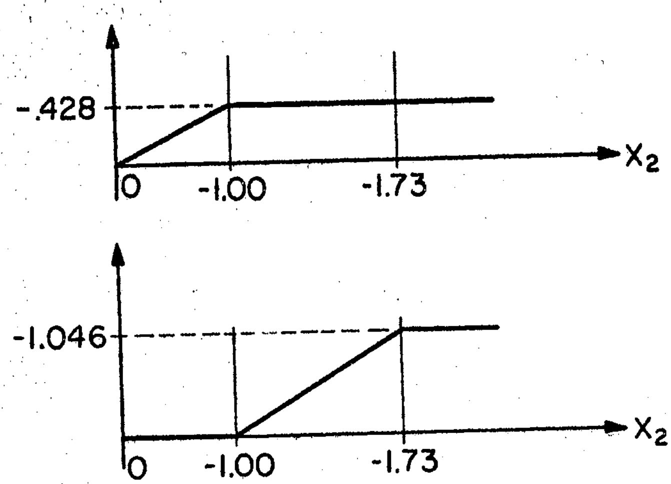
Now, substitution of the $\underline{Y}_q$ vectors and the $X_i^*(\underline{X}_q')$ quantities into the second expression yields

$$\sum_{q=1}^{P} \underline{Y}_q X_i^*(\underline{X}_q') = \begin{bmatrix} -7.368 \\ -4.530 \end{bmatrix}.$$

Carrying out the indicated inversion and multiplication operations gives the result of

$$\underline{W} = \begin{bmatrix} -.428 \\ -1.096 \end{bmatrix},$$

The two diode-type function generators required to implement the controller have now been completely specified. Their characteristics are shown below.




Implementation of these results is very simple and straightforward and may be carried out by the most suitable means, whether electrical, electronic, mechanical, or fluidic. As example, Figure 29 shows diode-generator type electronic and fluidic linear-segment functions. Implementation is achieved by the summation of the outputs of a number of these. It also includes comparison of the actual value of the dependent variable with this result, and assignment of the active levels accordingly.

The implementation of Example 4.7 would take the form shown in Figure 30, if such generators were used. Notice the extreme simplicity of the system.

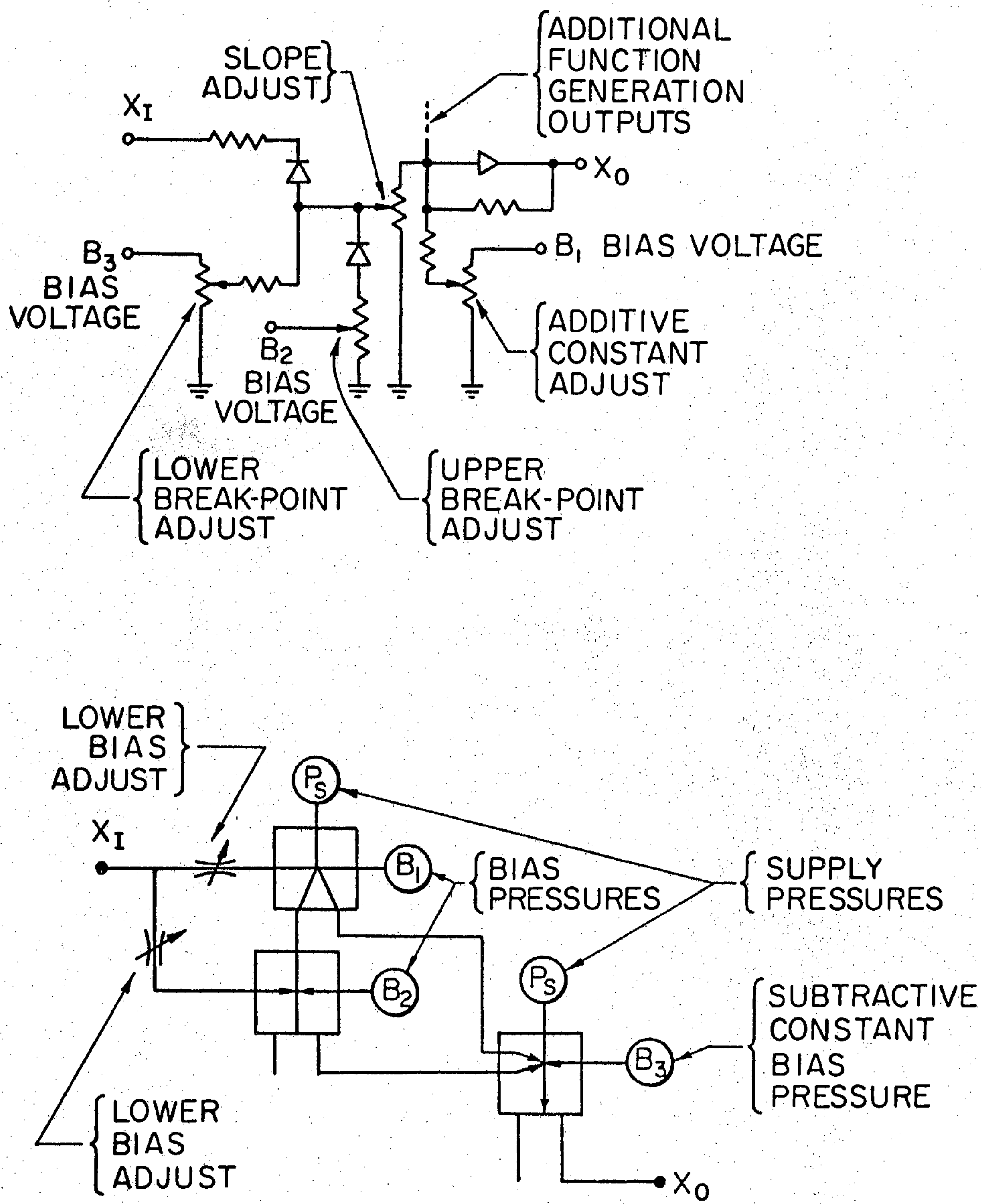



Figure 29. Electronic and Fluidic Diode-Type Function Generators

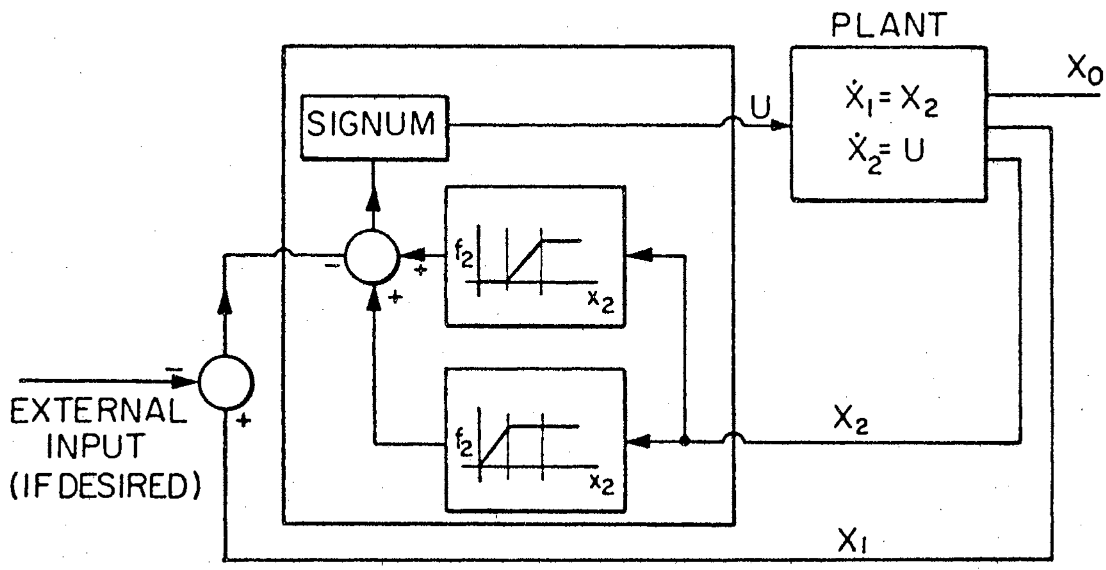



Figure 30. Implementation of the Linear Approximation Found in Example 4.7

# CHAPTER V

## APPLICATIONS

A theoretical base and a synthesis procedure for ODLC have already been presented in the thesis. The purpose of this chapter is to show how ODLC can be applied to various types of systems. This will be done by working several problems. Where possible, the plant equations will be modeled after some physical system. No attempt is made to exhaustively cover all types of plants and performance indices for which ODLC is valid. Rather, the few examples presented were chosen because they best illustrate significant characteristics of ODLC.

<u>Problem 5-1.</u> Roll control of a space vehicle. (P.I. - minimum time plus effort squared). Space vehicles are a natural application for ODLC because control effort must be furnished by small maneuvering thrusters. Normally, these are on-off devices incapable of being throttled. Thus, the controller must inherently be a pulsed controller. Let it be assumed that the equations of roll motion can be represented by the simplified equations,

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \frac{1}{I_R} U \,, \end{aligned} \tag{5.1}$$

where

$$X_1 = \text{roll position}$$

$$X_2 = \text{roll velocity}$$

$$I_R = \text{roll inertia}$$

and

$$U = \text{thruster output force.}$$

Suppose it is desired to construct the controller with more than one level of effort in each roll direction available. This could be provided by mounting two thrusters in each direction. Assume that this is done and control efforts of one and two units will be available in both directions. Therefore, assume that the control is to be a five-level ODLC. That is,

$$U \in \left[-2, -1, 0, +1, +2\right] \quad . \tag{5.2}$$

Assume also that the performance index is

$$\text{P.I.} = \int_{t_1}^{t_f} (1 + U^2)dt \quad . \tag{5.3}$$

The additional assumption will be made that the roll inertia ($I_R$) is changing as a function of time (fuel is being consumed by the controller), but slowly enough that the dynamics of this process can be ignored. The arbitrary values

$$U = \pm 2, \pm 1, 0$$

were chosen for computational convenience since their values will affect only the scaling of the problem. The Hamiltonian for the above system is

$$H = P_1X_1 + P_2 \frac{1}{I_R} U + P_3 + P_3U^2 \quad . \tag{5.4}$$

Since $P_3 = -1$, it can immediately be reduced to

$$H = P_1 X_2 + P_2 \frac{1}{I_R} U - 1 - U^2 \quad . \tag{5.4a}$$

The adjoint variables are

$$\begin{aligned} -\frac{\partial H}{\partial X_1} &= \dot{P}_1 = 0 \\ -\frac{\partial H}{\partial X_2} &= \dot{P}_2 = -P_1 \quad . \end{aligned} \tag{5.5}$$

For optimality, the term

$$P_2 \frac{1}{I_R} U - U^2$$

must be maximized. It can be determined by inspection that the relationship

$$\begin{aligned} U &= 0 && \text{for} && \left\| P_2 \frac{1}{I_R} \right\| < 1 \\ U &= \text{signum}(P_2) && \text{for} && 1 < \left\| P_2 \frac{1}{I_R} \right\| < 3 \\ U &= 2\ \text{signum}(P_2) && \text{for} && 3 < \left\| P_2 \frac{1}{I_R} \right\| \end{aligned} \tag{5.6}$$

will be the control law for ODLC. Evaluation of the Hamiltonian at time $t_f$ provides the relationship

$$0 = P_{2_{t_f}} \frac{1}{I_R} U_{t_f} - 1 - U_{t_f}^2 \quad . \tag{5.7}$$

Substitution of possible input values gives the result

$$\left( P_2 \frac{1}{I_R} \right)_{t_f} = 2 \qquad \text{for} \qquad X_{10} > 0 \quad . \tag{5.8}$$

Since $I_R$ is assumed to be a slowly varying function of time, it will suffice to make a number of solutions for the ODLC with $I_R$ "frozen" at various values. Let these values be taken as:

$$\frac{1}{I_R} = 1 \quad , \quad \frac{1}{I_R} = \frac{4}{3} \quad , \quad \frac{1}{I_R} = 2 \quad , \quad \frac{1}{I_R} = 4 \quad .$$

Now, switching points on the ODLC surfaces can be located by the use of backward integration. These points are shown in Table I.[1] The results are plotted in Figure 31.

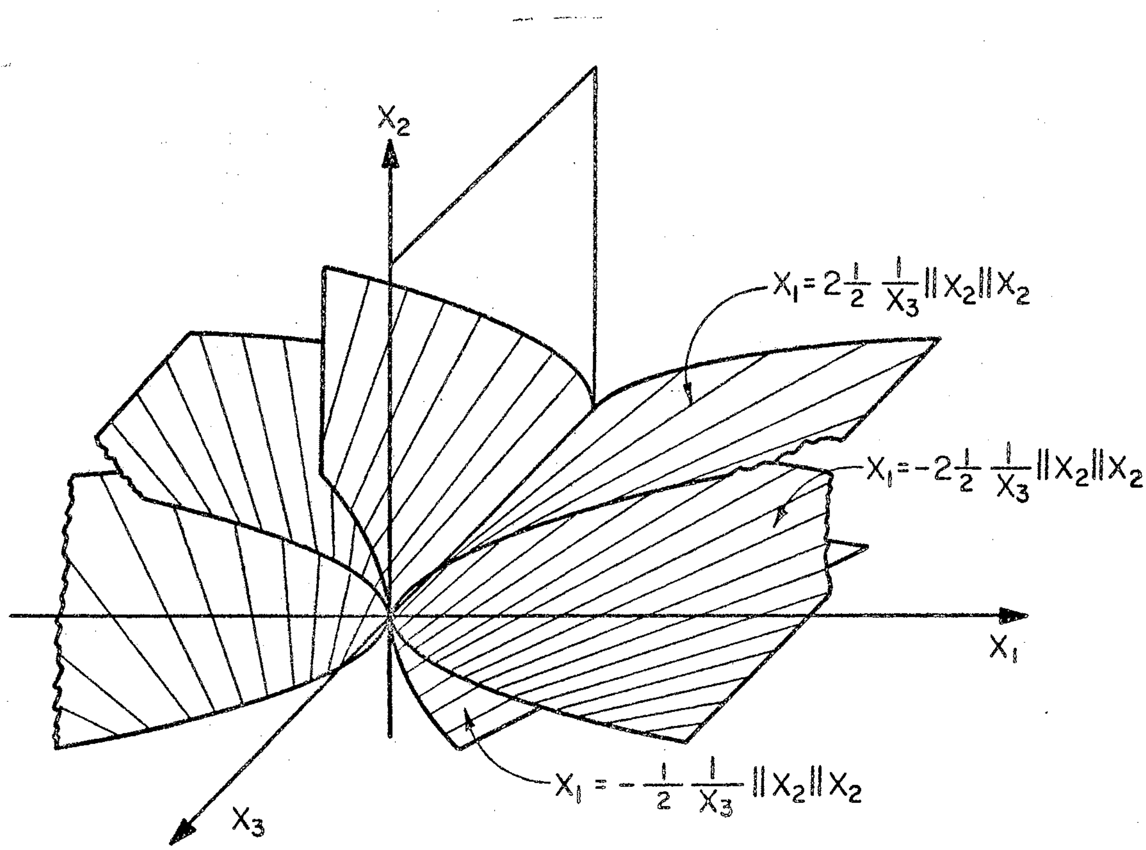



Figure 31. Optimal Discrete Level Switching Surfaces for Problem 5-1

---

[1]Because of the symmetry of the plant equations, it is sufficient to deal only with the right half of the state space.

TABLE I

OPTIMAL SWITCH POINTS FOR PROBLEM 5-1

| U = +1 | | | | | | Terminal Surface | | |
|---|---|---|---|---|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $X_1$ | $X_2$ | $X_3$ | $X_1$ | $X_2$ | $X_3$ |
| 0.023 | -0.250 | 1.330 | 0.012 | -0.250 | 2.500 | 1.531 | -3.500 | 4.000 |
| 0.094 | -0.500 | 1.330 | 0.050 | -0.500 | 2.500 | 0.009 | -0.250 | 3.500 |
| 0.375 | -1.000 | 1.330 | 0.200 | -1.000 | 2.500 | 0.036 | -0.500 | 3.500 |
| 1.500 | -2.000 | 1.330 | 0.450 | -1.500 | 2.500 | 0.143 | -1.000 | 3.500 |
| 6.000 | -4.000 | 1.330 | 0.800 | -2.000 | 2.500 | 0.321 | -1.500 | 3.500 |
| 0.016 | -0.250 | 2.000 | 1.250 | -2.500 | 2.500 | 0.571 | -2.000 | 3.500 |
| 0.063 | -0.500 | 2.000 | 1.800 | -3.000 | 2.500 | 0.893 | -2.500 | 3.500 |
| 0.250 | -1.000 | 2.000 | 2.450 | -3.500 | 2.500 | 1.286 | -3.000 | 3.500 |
| 1.000 | -2.000 | 2.000 | 3.200 | -4.000 | 2.500 | 1.750 | -3.500 | 3.500 |
| 4.000 | -4.000 | 2.000 | 0.010 | -0.250 | 3.000 | 2.286 | -4.000 | 3.500 |
| 0.008 | -0.250 | 4.000 | 0.042 | -0.500 | 3.000 | 0.0 | 0.0 | 1.000 |
| 0.031 | -0.500 | 4.000 | 0.167 | -1.000 | 3.000 | 0.031 | 0.250 | 1.000 |
| 0.125 | -1.000 | 4.000 | 0.375 | -1.500 | 3.000 | 0.125 | -0.500 | 1.000 |
| 0.500 | -2.000 | 4.000 | 0.667 | -2.000 | 3.000 | 0.500 | -1.000 | 1.000 |
| 2.000 | -4.000 | 4.000 | 1.042 | -2.500 | 3.000 | 2.000 | -2.000 | 1.000 |
| 0.0 | 0.0 | 1.500 | 1.500 | -3.000 | 3.000 | 8.000 | -4.000 | 1.000 |
| 0.0 | 0.0 | 2.000 | 2.042 | 03.500 | 3.000 | 1.125 | -1.500 | 1.000 |
| 0.0 | 0.0 | 2.500 | 2.668 | -4.000 | 3.000 | 3.125 | -2.500 | 1.000 |
| 0.0 | 0.0 | 3.000 | 0.281 | -1.500 | 4.000 | 4.500 | -3.000 | 1.000 |
| 0.0 | 0.0 | 3.500 | 0.781 | -2.500 | 4.000 | 6.125 | -3.500 | 1.000 |
| 0.0 | 0.0 | 4.000 | 1.125 | -3.000 | 4.000 | | | |

TABLE I (Continued)

| U from -1 to 0 | | | Switching Surface | | | | | |
|---|---|---|---|---|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $X_1$ | $X_2$ | $X_3$ | $X_1$ | $X_2$ | $X_3$ |
| 0.115 | -0.250 | 1.330 | 0.062 | -0.250 | 2.500 | 7.656 | -3.500 | 4.000 |
| 0.470 | -0.500 | 1.330 | 0.062 | -0.500 | 2.500 | 0.714 | -0.250 | 3.500 |
| 1.875 | -1.000 | 1.330 | 1.000 | -1.000 | 2.500 | 0.178 | -0.500 | 3.500 |
| 1.500 | -3.000 | 1.330 | 2.250 | -1.500 | 2.500 | 0.714 | -1.000 | 3.500 |
| 30.000 | -4.000 | 1.330 | 40.000 | -2.000 | 2.500 | 1.607 | -1.500 | 3.500 |
| 0.078 | -0.250 | 2.000 | 6.250 | -2.500 | 2.500 | 2.857 | -2.000 | 3.500 |
| 0.312 | -0.500 | 2.000 | 9.000 | -3.000 | 2.500 | 4.465 | -2.500 | 3.500 |
| 1.250 | -1.000 | 2.000 | 12.250 | -3.500 | 2.500 | 6.430 | 3.000 | 3.500 |
| 5.000 | -2.000 | 2.000 | 16.000 | -4.000 | 2.500 | 8.750 | -3.500 | 3.500 |
| 20.00 | -4.000 | 2.000 | 0.052 | -0.250 | 3.000 | 11.430 | -4.000 | 3.500 |
| 0.039 | -0.250 | 4.000 | 0.209 | -0.500 | 3.000 | 0.0 | 0.0 | 1.000 |
| 0.156 | -0.500 | 4.000 | 0.833 | -1.000 | 3.000 | 0.156 | -0.250 | 1.000 |
| 0.625 | -1.000 | 4.000 | 1.875 | -1.500 | 3.000 | 0.625 | -0.500 | 1.000 |
| 2.500 | -2.000 | 4.000 | 3.333 | -2.000 | 3.000 | 2.500 | -1.000 | 1.000 |
| 10.000 | -4.000 | 4.000 | 5.208 | -2.500 | 3.000 | 10.000 | -2.000 | 1.000 |
| 0.0 | 0.0 | 1.500 | 7.500 | -3.000 | 3.000 | 10.000 | -4.000 | 1.000 |
| 0.0 | 0.0 | 2.000 | 10.208 | -3.500 | 3.000 | 5.625 | -1.500 | 1.000 |
| 0.0 | 0.0 | 2.500 | 13.335 | -4.000 | 3.000 | 15.625 | -2.500 | 1.000 |
| 0.0 | 0.0 | 2.500 | 1.406 | -1.500 | 4.000 | 22.500 | -3.000 | 1.000 |
| 0.0 | 0.0 | 3.500 | 3.906 | -2.500 | 4.000 | 30.625 | -3.500 | 1.000 |
| 0.0 | 0.0 | 4.000 | 5.625 | -3.000 | 4.000 | | | |

TABLE I (Continued)

| U from -2 to -1 | | | Switching Surface | | | | | |
|---|---|---|---|---|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $X_1$ | $X_2$ | $X_3$ | $X_1$ | $X_2$ | $X_3$ |
| 0.115 | 0.250 | 1.330 | 0.062 | 0.250 | 2.500 | 7.656 | 3.500 | 4.000 |
| 0.470 | 0.500 | 1.330 | 0.250 | 0.500 | 2.500 | 0.714 | 0.250 | 3.500 |
| 1.875 | 1.000 | 1.330 | 1.000 | 1.000 | 2.500 | 0.178 | 0.500 | 3.500 |
| 7.500 | 2.000 | 1.330 | 2.250 | 1.500 | 2.500 | 0.714 | 1.000 | 3.500 |
| 30.000 | 4.000 | 1.330 | 4.000 | 2.000 | 2.500 | 1.607 | 1.500 | 3.500 |
| 0.078 | 0.250 | 2.000 | 6.250 | 2.500 | 2.500 | 2.857 | 2.000 | 3.500 |
| 0.312 | 0.500 | 2.000 | 9.000 | 3.000 | 2.500 | 4.465 | 2.500 | 3.500 |
| 1.250 | 1.000 | 2.000 | 12.250 | 3.500 | 2.500 | 6.430 | 3.000 | 3.500 |
| 5.000 | 2.000 | 2.000 | 16.000 | 4.000 | 2.500 | 8.750 | 3.500 | 3.500 |
| 20.000 | 4.000 | 2.000 | 0.052 | 0.250 | 3.000 | 11.430 | 4.000 | 3.000 |
| 0.039 | 0.250 | 4.000 | 0.209 | 0.500 | 3.000 | 0.0 | 0.0 | 1.000 |
| 0.156 | 0.500 | 4.000 | 0.833 | 1.000 | 3.000 | 0.156 | 0.250 | 1.000 |
| 0.125 | 1.000 | 4.000 | 1.875 | 1.500 | 3.000 | 0.625 | 0.500 | 1.000 |
| 2.500 | 2.000 | 4.000 | 3.333 | 2.000 | 3.000 | 2.500 | 1.000 | 1.000 |
| 10.000 | 4.000 | 4.000 | 5.208 | 2.500 | 3.000 | 10.000 | 2.000 | 1.000 |
| 0.0 | 0.0 | 1.500 | 7.500 | 3.000 | 3.000 | 40.000 | 4.000 | 1.000 |
| 0.0 | 0.0 | 2.000 | 10.208 | 3.500 | 3.000 | 5.625 | 1.500 | 1.000 |
| 0.0 | 0.0 | 2.500 | 13.335 | 4.000 | 3.000 | 15.625 | 2.500 | 1.000 |
| 0.0 | 0.0 | 3.000 | 1.406 | 1.500 | 4.000 | 22.500 | 3.000 | 1.000 |
| 0.0 | 0.0 | 3.500 | 3.106 | 2.500 | 4.000 | 30.625 | 3.500 | 1.000 |
| 0.0 | 0.0 | 4.000 | 5.625 | 3.000 | 4.000 | | | |

Since the problem is of rather simple form, the equations for the surfaces can be solved directly. They are included in Figure 31. Closed loop synthesis of the ODLC using Smith's (22) linear segment technique will now be carried out.

Application of the technique results in the function-generator specification shown in Figure 32. $X_1$ was assumed to be the dependent variable and the break points along $X_2$, $X_3-1$, $S_1 = (-2X_2+X_3-1)$, and $S_2 = (-2X_2-X_3+4)$ were chosen to be -0.50, -1.0, -2.0, -4.0; 1.0, 2.0, 3.0; 2.0, 4.0, 6.0, 8.0, 10.0, 12.0; 2.0, 4.0, 6.0, 8.0, 10.0, 12.0.

A schematic of the implementation required by Smith's (22) technique is shown in Figure 33. Figures 34 and 35 are comparisons of the trajectories and control sequences, respectively, between exact implementation of the ODLC surfaces and the approximation.[3] The dotted lines are a mixed implementation where approximations were used for all surfaces but the terminal-trajectory one.

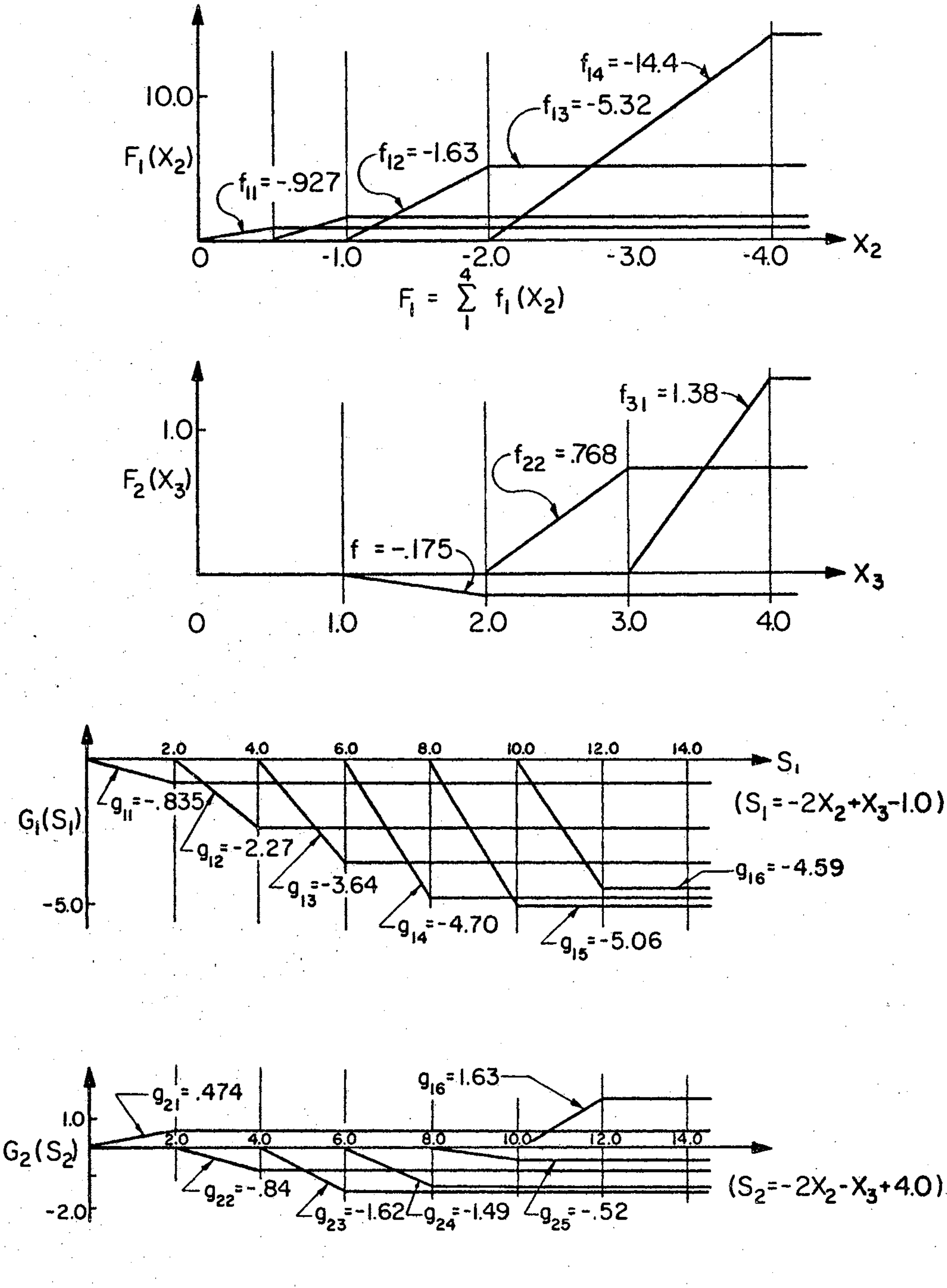
Non-autonomous control of these systems could be realized by the simple addition of a summing junction to compare the values of the state variable $X_1$ with the allowable input. The total non-autonomous system is shown in Figure 36.

For the example shown the approximation achieves a remarkably close-to-optimal control. Most of the system degradation which does appear shows up as false switching of the control as the system

---

[2] Because of the problem symmetry only the function generation for the right-half space is discussed. Further, only the (U=+1) terminal surface is shown, since the procedure yielded for the other surfaces identical results, except for a multiplicative factor of five.

[3] Simulation of implementations was carried out on the digital computer. For the simulation in question, initial conditions of $X_1 = 3.0$, $X_2 = 0.0$, $X_3 = 1.0$ were assumed.




$$\hat{X}_1 = F = F_1 + F_2 + G_1 + G_2$$

NOTE: Multiply all coefficients by five for non-terminal surfaces.

Figure 32. Function Generators for Implementation of $U = +1$ Terminal ODLC Surface for Problem 6.1

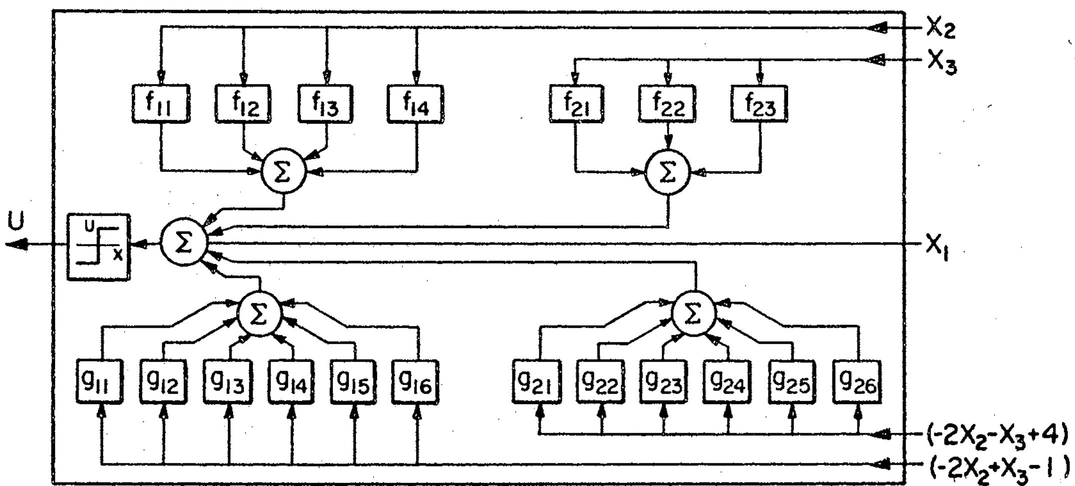



Figure 33. Schematic Diagram for Implementation of the Switching Surface Approximation Shown in Figure 32

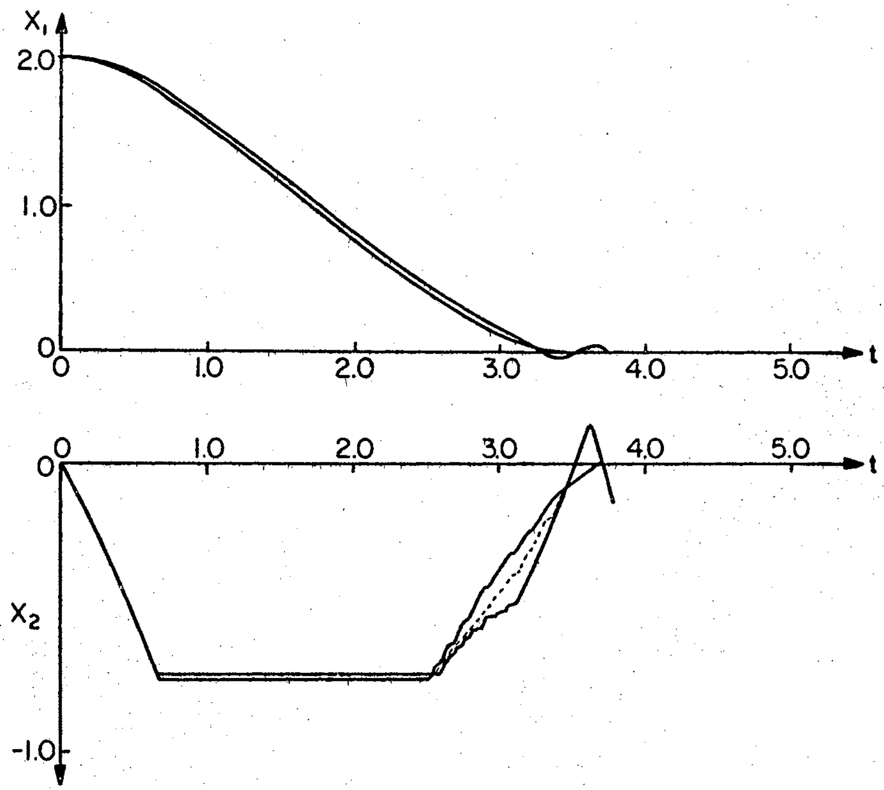



Figure 34. Comparison of Trajectories Resulting From Exact and Linear Segment Implementation of ODLC for Problem 5.1

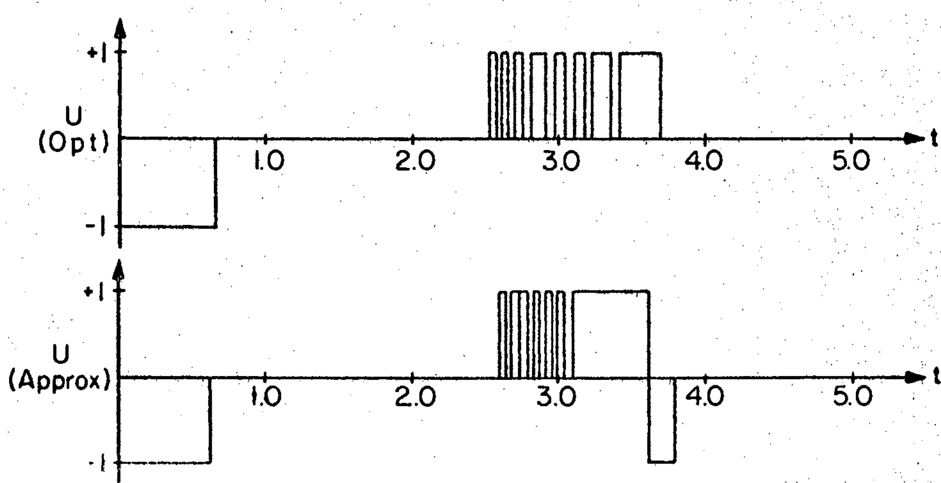



Figure 35. Comparison of the Control Resulting From Exact and Linear Segment Implementation of ODLC for Problem 5.1

trajectory moves along the terminal surface. This is to be expected, since the controller attempts to force the trajectory along the approximation rather than the true surface. Use of the mixed approximation with an exact expression used for the terminal surface and approximations used for the other produces system behavior which is nearly indistinguishable in performance from the completely optimal. Obviously, terminal surface approximations more critically affect system performance than do the approximations of other surfaces.

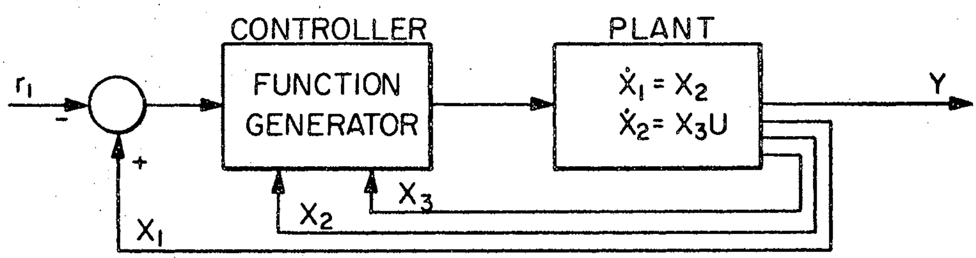



Figure 36. Non-Autonomous Suboptimal DLC for Problem 5-1

It is interesting to note that in the example even the optimal control required multiple firing on the terminal surface. The reason for this is that the roll inertia is actually a function of firing time and as the roll inertia changes so does the terminal trajectory (see Figure 31). Thus the control is repeatedly turned off as the controller repeatedly re-evaluates the control needed to drive the system to the origin. Intermittent firings such as this are probably not desirable, but, in actual system where the roll inertia does not change so rapidly and where hysteresis in the control actuators inherently exists (or could be introduced), such intermittency could be minimized by the introduction of a terminal-switch dead-band.

In essence, this would change the terminal surface into a terminal region. Approximation accuracy requirements for the surface can then be reduced to satisfying the requirement that the approximation will be entirely contained within this region. Implied is a trade-off between this terminal region "thickness," which will adversely affect steady-state accuracy, and approximation accuracy, which will adversely affect controller implementability. Although it was considered to be beyond the scope of this paper, further investigation of this technique to synthesize nearly optimal DLC systems with non-critical terminal surfaces is recommended.

Problem 5-2. Roll control of a space vehicle (P.I. = minimum time plus effort and error squared). This problem will illustrate the effects of different performance indices on optimal switching surfaces. This will be done by using a performance index of the form,

$$\text{P.I.} = \int_{t_o}^{t_f} (1 + X_1^2 + X_2^2 + U^2)\,dt \tag{5.9}$$

in the synthesis of a controller for the system of Problem 5-1.

In review, the system of 5-1 is described by the equations

$$\dot{X}_1 = X_2$$
$$\dot{X}_2 = \frac{1}{I_R} U \quad .^{4} \tag{5.1}$$

[4] $I_R$ will be assumed to equal a constant value of unity in Problem 5-2. This is done simply for the sake of clarity. The validity of the illustration will not be affected by this simplification.

Allowable input levels are agreed to be

$$U \in \left[-2,\ -1,\ 0,\ +1,\ +2\right] \quad . \tag{5.2}$$

The Hamiltonian for this system with the new P.I. is

$$H = P_1X_2 + P_2U - 1 - X_1^2 - X_2^2 - U^2 \quad . \tag{5.10}$$

Adjoint variables are

$$\begin{aligned} \dot{P}_1 &= 2X_1 \\ \dot{P}_2 &= -P_1 + 2X_2 \end{aligned} \quad . \tag{5.11}$$

Inspection of the Hamiltonian indicates that

$$(P_2U - U^2) \tag{5.12}$$

must be maximized to achieve optimal control. The resulting optimal control law is

$$\begin{array}{lll} U = 0 & \text{for} & \|P_2\| < 1 \\ U = \text{signum}\ (P_2) & \text{for} & 1 < \|P_2\| < 3 \\ U = 2\ \text{signum}\ (P_2) & \text{for} & 3 < \|P_2\| \end{array} \quad . \tag{5.13}$$

Notice that not only are the system equations the same for both Problems 5-1 and 5-2, but that the control laws appear to be the same. However, the adjoint variables are considerably different. As will be shown, they seriously affect the surface shapes.

Since the system is second-order, backward integration from the final states was used as the preferred method of solution. The final state conditions are

$$X_1(t_f) = 0$$
$$X_2(t_f) = 0 \quad . \tag{5.14}$$

An additional final condition can be obtained by evaluation of the Hamiltonian at the final states. It is that

$$P_2(t_f) = 2 \tag{5.15}$$

(for trajectories in the right half of state space). By the use of these three final boundary conditions and the selection of various values for the unspecified adjoint variable ($P_1$), the following table (Table II) was compiled.

TABLE II

OPTIMAL SWITCH POINTS FOR PROBLEM 5-2

| U = +1 Terminal Surface | | U = -1 to 0 Terminal Surface | | U = -2 to -1 Terminal Surface | |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 5.550 | -3.000 | 9.111 | -3.870 |
| 0.167 | -0.577 | 4.475 | -2.500 | 6.204 | -2.770 |
| 0.333 | -0.816 | 3.600 | -2.000 | 5.109 | -2.230 |
| 0.500 | 1.000 | 2.775 | -1.500 | 4.182 | -1.680 |
| 0.667 | -1.155 | 1.840 | -1.000 | 3.238 | -1.140 |
| 0.833 | -1.211 | 1.286 | -0.750 | 2.173 | -0.560 |
| 1.000 | -1.414 | 0.665 | -0.500 | 1.531 | -0.240 |
| 1.167 | -1.526 | 0.166 | -0.250 | 0.786 | 0.090 |
| 1.333 | -1.633 | 8.600 | -4.000 | 0.170 | 0.210 |
| 1.500 | -1.732 | 12.860 | -5.000 | | |

Again, because of system symmetry, only the right half of state space need be explored. Figure 37 is a plot of the optimal switching surfaces for this system and performance index. Only the terminal trajectory switching surface is unchanged from Problem 5-1.

Because of the close similarity between Problems 5-1 and 5-2, closed-loop synthesis of 5-2 will not be carried out. Synthesis steps would be unchanged.

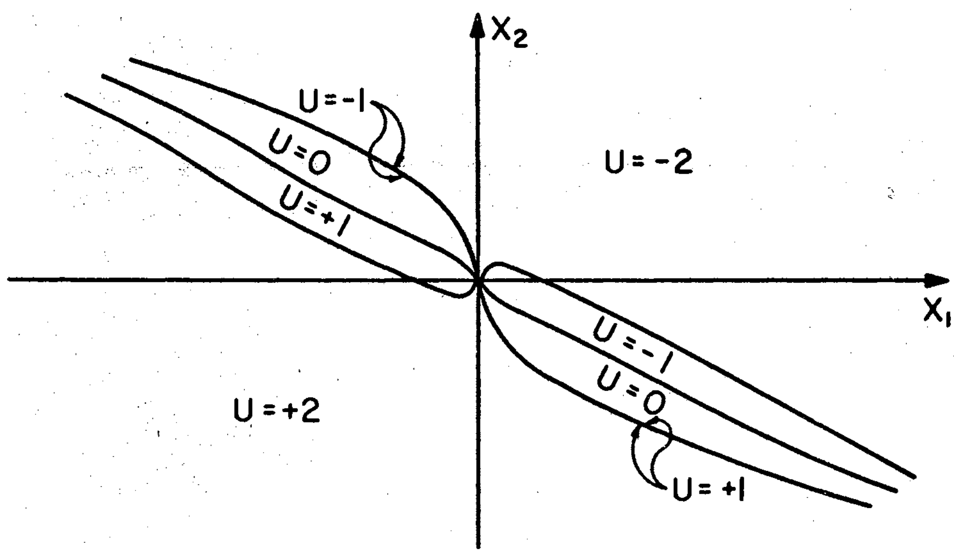



Figure 37. ODLC Surfaces for Problem 5-2

Problem 5-3. Non-linear dynamical system (non-autonomous control). Consider a non-linear dynamical system described by the equations

$$\dot{X}_1 = X_2$$
$$\dot{X}_2 = -X_1||X_1|| + U \quad . \tag{5.16}$$

This is basically a non-linear spring-mass system with a "square-law" hardening spring. Assume that the chosen performance index is

$$\text{P.I.} = \int_{t_o}^{t_f} (1 + U^2)dt$$

and the allowable control levels are

$$U \epsilon \left[-2, -1, 0, +1, +2\right] \quad . \tag{5.17}$$

The system equations must be modified slightly since non-autonomous control is desired. The check for allowable external controls,

$$\begin{aligned} 0 &= X_2 \\ 0 &= -X_1 \| X_1 \| + U \end{aligned} \tag{5.18}$$

indicates that only $r_1 = X_{1d}$ can be allowed.

However, introduction of this external input by the linear co-ordinate transformation

$$Y_1 = X_1 - X_{1d} \tag{5.19}$$

changes the system equation to

$$\begin{aligned} \dot{Y}_1 &= X_2 \\ \dot{X}_2 &= - \; Y_1 + X_{1d} \; \| \; Y_1 + X_1 d \; \| + U \quad . \end{aligned} \tag{5.20}$$

Therefore, $r = X_{1d}$ will be carried along as a new pseudo-state variable, and the switching surfaces will be increased from two to three dimensional. The Hamiltonian for the expanded system is

$$H = P_1X_2 - P_2(Y_1 + r) \| (Y_1 + r) \| + P_2U - 1 - U^2 \quad . \tag{5.20}$$

Adjoint variables for the system are

$$\dot{P}_1 = 2P_2||(Y_1 + r)||$$

$$\dot{P}_2 = -P_1 \quad . \tag{5.21}$$

Final boundary conditions for the problem are

$$Y_1(t_f) = 0$$

$$X_2(t_f) = 0 \quad . \tag{5.22}$$

The additional final constraint, obtained from evaluation of the Hamiltonian is

$$0 = -P_2||r||r + P_2U - 1 - U^2 \quad . \tag{5.23}$$

Necessary conditions for optimality specify the optimal control as

$$\begin{array}{lll} U = 0 & \text{for} & ||P_2|| < 1 \\ U = \text{signum } (P_2) & \text{for} & 1 < ||P_2|| < 3 \\ U = 2 \text{ signum } (P_2) & \text{for} & 3 < ||P_2|| \quad . \end{array} \tag{5.24}$$

At this point, enough information has been determined that specification of values for r and $P_1(t_f)$ will allow backward integration to generate optimal trajectories.[5]

Figure 38 shows the ODLC surfaces for Problem 5-3. These surfaces are surprisingly complex and possess some rather unusual shapes and other characteristics. Because of this complexity approximation of the surfaces by any means would be a challenging task. In fact, although

---

[5]Because physical limitations dictate that U cannot drive the system to greater values of $||X_1||$ than $\sqrt{2}$, r was restricted to $(-\sqrt{2}, +\sqrt{2})$.

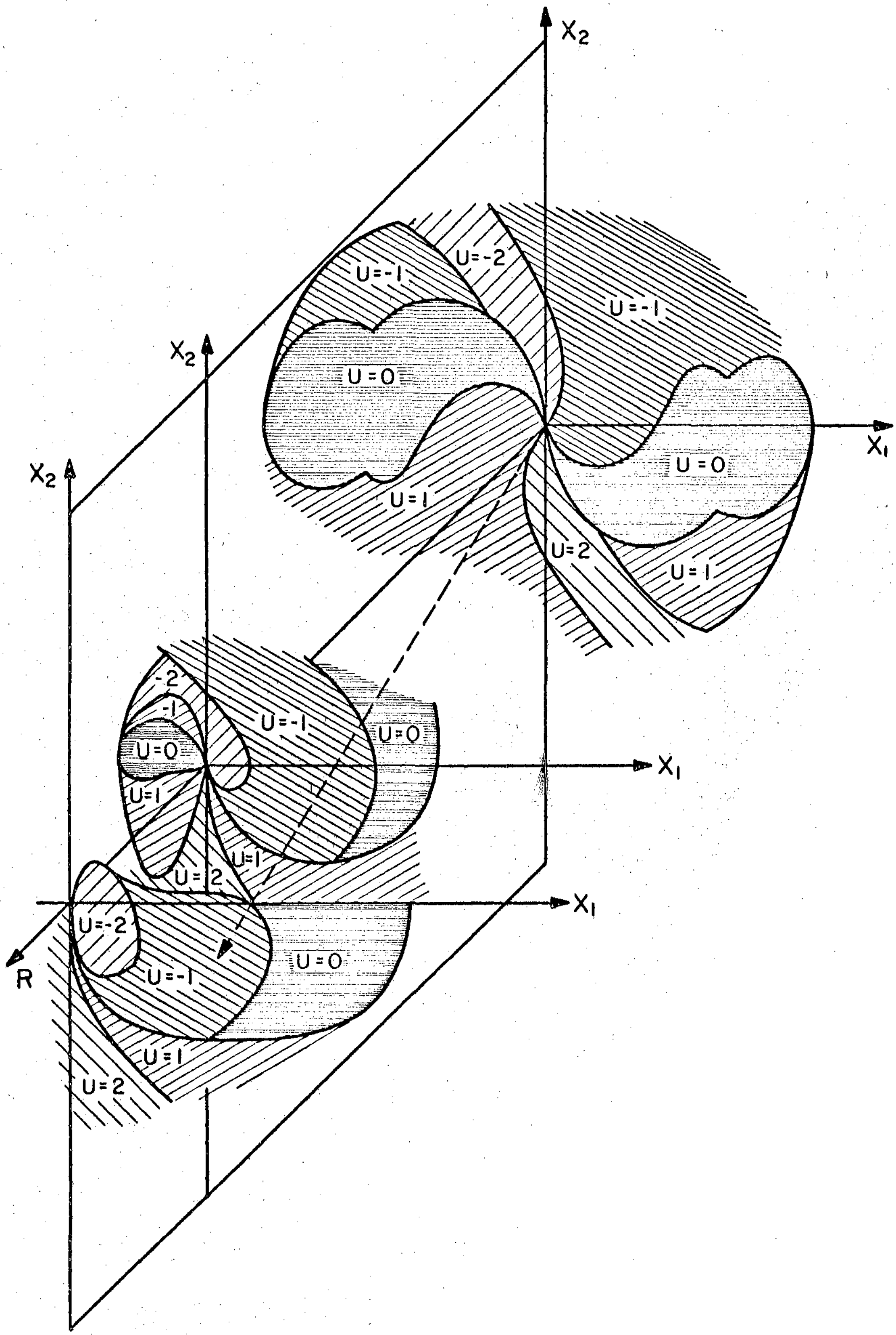



Figure 38. ODLC Surfaces for Problem 5.3 Shown are Cross-Sections at R = 0, R = -1.0, R = -1.4

it should be feasible to approximate the surfaces throughout a small region (e.g., close to the R-axis), large area approximation would appear to be so formidable as to be of dubious worth.

Exact determination of surface shapes in certain regions was found to be extremely difficult. The reason for this is that conjugate points abound in the state space of this system and the necessary (but not sufficient) conditions specified by the Maximum Principle were inadequate to determine optimality beyond such points. For this reason, certain of the switching regions shown in Figure 37 are not closed but simply trail-off into these uncertain regions. It was considered beyond the scope of the present work to attempt to explore them further than was done in Figure 37.

Before proceeding to the next problem, the results of Problem 5-3 can be used to illustrate an aspect of functional approximation where great caution must be used. All functional approximation techniques necessarily assume that the variable chosen to be dependent is uniquely determined by the remaining (independent) variables. As Figure 37 illustrates, several of the surfaces simply do not provide such a unique relationship (except in sub-regions), but in many cases are at least double-valued. Thus, the designer must exercise great care in choosing functional approximations to assure that they provide unique relationships in the areas of interest. This could become especially difficult for systems of higher order, if the hyper-surfaces are very complex.

<u>Problem</u> 5-4. Third-order plant with two input variables. This third-order system was chosen because it possesses a vector, rather than a

scalar input. As such, it serves to illustrate the applicability of ODLC synthesis to systems with vector inputs.

The plant can be described by the equations

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = -X_3 + U_2$$

$$\dot{X}_3 = -U_1 \qquad . \qquad (5.25)$$

A performance index of

$$P.I. = \int_{t_o}^{t_f} (1 + U_1^2 + U_2^2)dt$$

is chosen, and the inputs are constrained to

$$U_1 \epsilon \left[-2, -1, 0, +1, +2\right]$$

$$U_2 \epsilon \left[-2, -1, 0, +1, +2\right] \qquad . \qquad (5.26)$$

Differentiation of the Hamiltonian for the system,

$$H = P_1X_2 - P_2X_3 + P_2U_2 - P_3U_1 - 1 - U_1^2 - U_2^2 \quad , \qquad (5.27)$$

yields the adjoint variables

$$\dot{P}_1 = 0$$

$$\dot{P}_2 = -P$$

$$\dot{P}_3 = P_2 \qquad . \qquad (5.28)$$

The optimal control law is obtained by requiring that

$$(P_2U_2 - P_3U_1 - U_1^2 - U_2^2)$$

be maximized. This law is

$$U_1 = 0 \qquad \text{for} \qquad ||P_3|| < 1$$
$$U_1 = -\text{signum } (P_2) \qquad \text{for} \qquad 1 < ||P_3|| < 3$$
$$U_1 = -2 \text{ signum } (P_2) \qquad \text{for} \qquad 3 < ||P_3||$$

$$U_2 = 0 \qquad \text{for} \qquad ||P_2|| < 1$$
$$U_2 = \text{signum } (P_2) \qquad \text{for} \qquad 1 \leq ||P_2|| < 3$$
$$U_2 = 2 \text{ signum } (P_2) \qquad \text{for} \qquad 3 < ||P_2|| \quad . \tag{5.29}$$

An additional constraint obtained from evaluation of the Hamiltonian at the final states is

$$P_2U_2 - P_3U_1 - U_1^2 - U_2^2 - 1 = 0 \quad . \tag{5.30}$$

Because of the added dimensionality of this problem, both backward integration and the modified dynamic programming described in Chapter IV and Appendix II, were used to obtain points on the optimal switching surfaces.

The problem is now ready for controller synthesis by approximation to the ODLC surfaces. Before proceeding with Smith's (22) technique, it is interesting to point out that direct optimal digital control of the plant could be achieved through a very small "on-line" computer, with a sufficient number of switching points stored in its memory to allow accurate interpolation.

Figures 39 through 44 show the linear-segment non-linear functions for closed-loop controller implementations resulting from the linear-segment approximation. In the interest of simplicity, the region of approximation was limited to $-3 \leq X_2 \leq +3$; $-1 \leq X_3 \leq +1$.[6] The functional

---

[6] $X_3$ was limited to $X_3 > 0$ on the right-half plane, and $X_3 < 0$ on the left, because, for the practical situations where initial conditions are taken on the $(X_1, X_2)$-plane, the system will always remain in these regions.

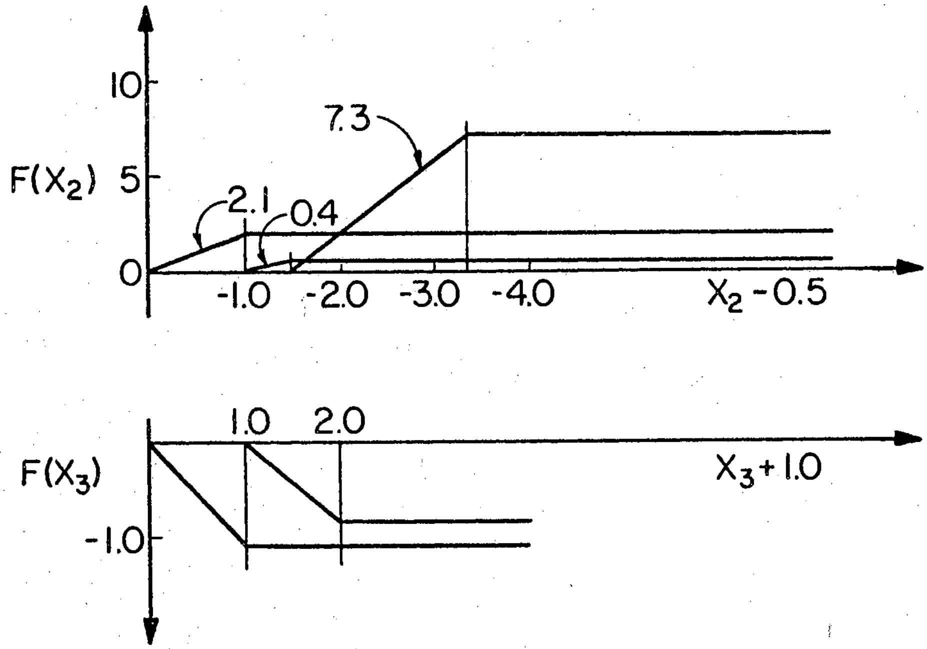



Figure 39. Optimal Switching Surface Approximation of $U_2 = -1/0$, for Problem 5.4

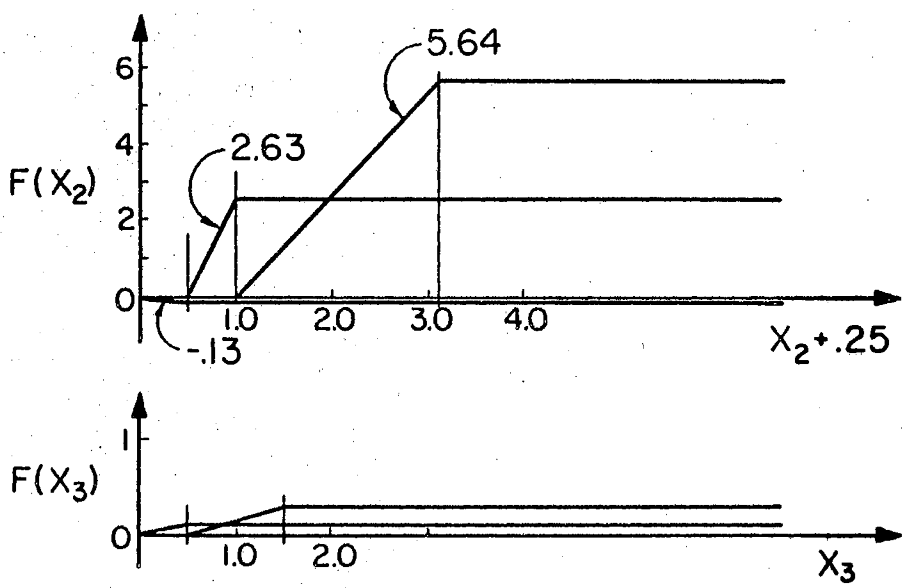



Figure 40. Optimal Switching Surface Approximation of $U_2 = -2/-1$

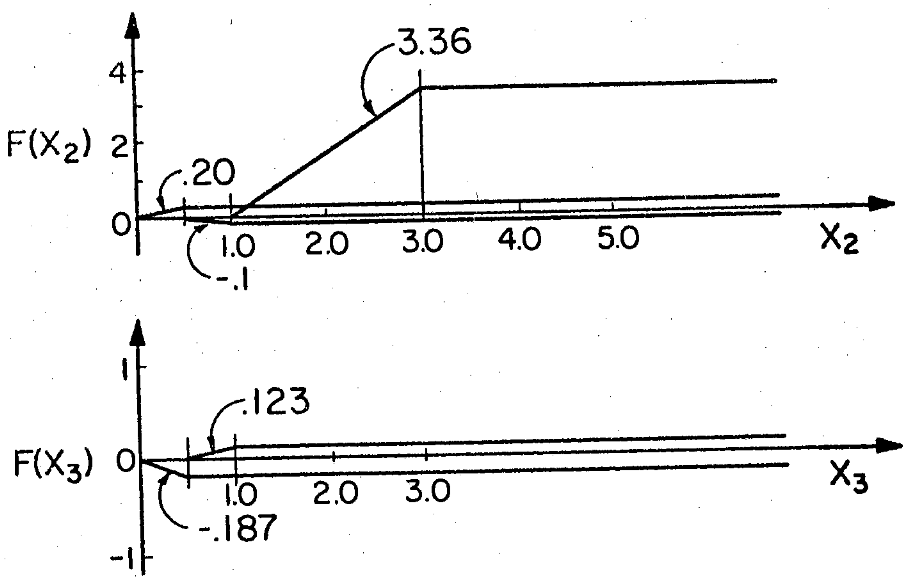



Figure 41. Optimal Switching Surface Approximation of $U_2 = 0/+1$

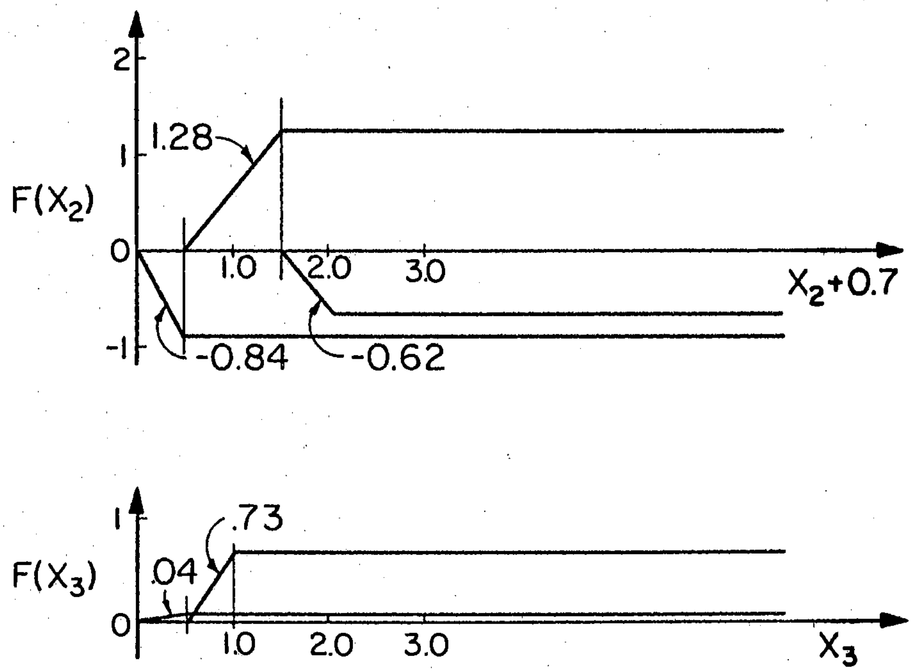



Figure 42. Optimal Switching Surface Approximation of $U_1 = +2/+1$

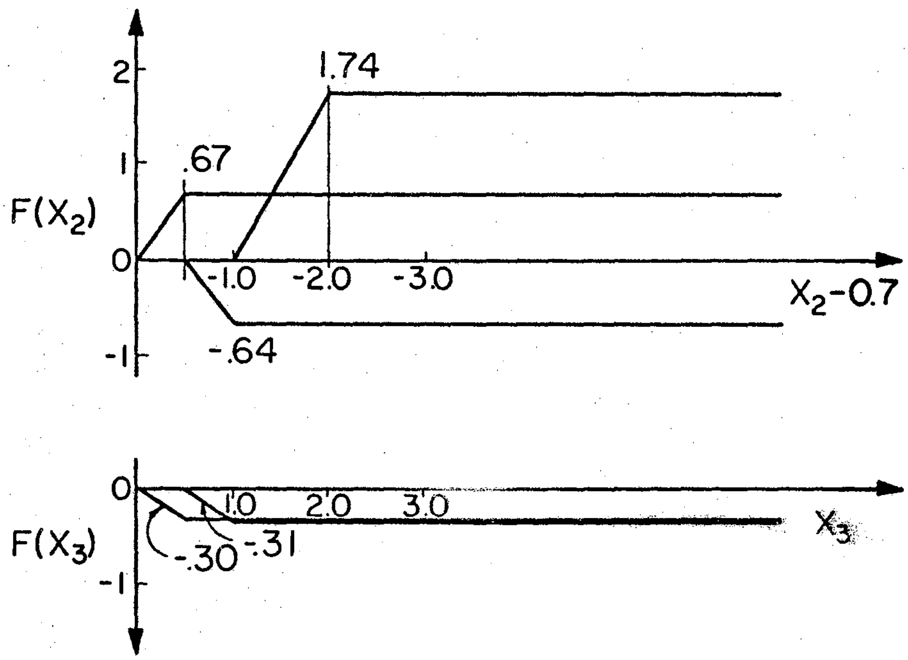



Figure 43. Optimal Switching Surface Approximation of $U_1 = +1/0$, for Problem 5.4

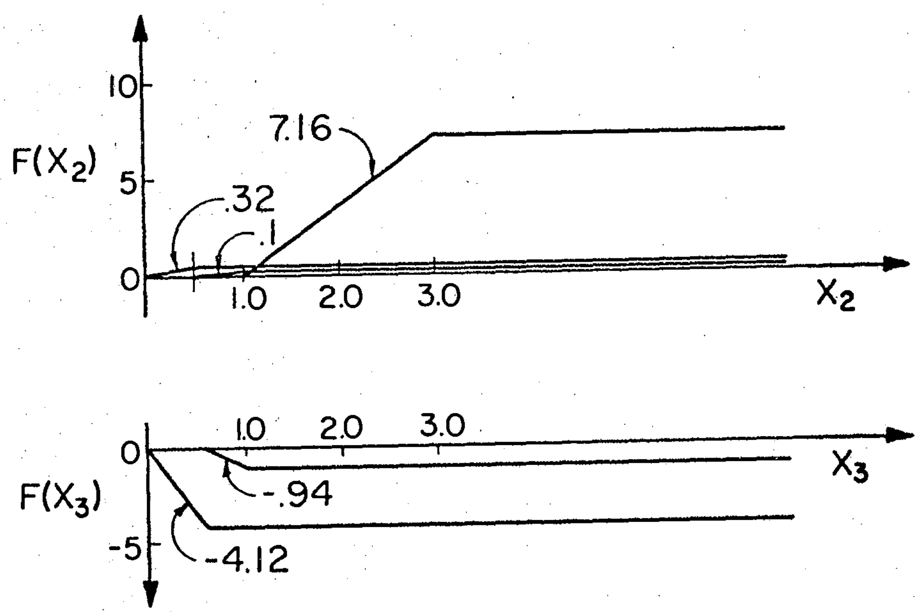



Figure 44. Optimal Switching Surface Approximation of $U_1 = -1/0$, for Problem 5.4

approximation for the surface $U_1 = -2/-1$ is not shown because it was never reached in the simulations made. Further, it will not be reached, in practical applications, from initial conditions reasonably close to the $(X_1, X_2)$-plane.

As an illustration of the controller, Figure 45 shows the system trajectory and the inputs for a typical set of initial conditions. Notice that the system becomes oscillatory about the origin. The reason for the "unsteadiness" is the approximation to the surfaces is not sufficiently accurate in the immediate vicinity of this point. If this is not acceptable, a finer approximation to the surface will be necessary. Such investigations were considered to be beyond the scope of this paper.

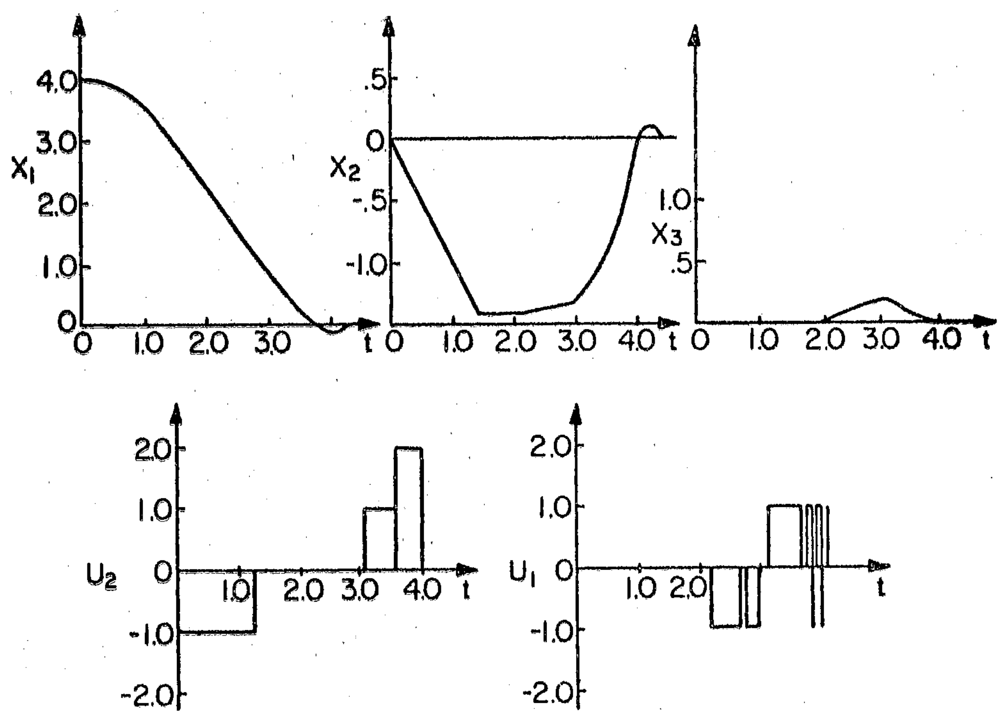



Figure 45. System Trajectory and Inputs for a Typical Set of Initial Conditions, Problem 5-4 Approximated ODLC

Although it is totally unnecessary for synthesis of the controller, since the system is only of third order, the QDLC surfaces can be drawn. They can be seen in Figures 46 and 47.

This problem illustrates the fact that the conditions for optimality are necessary, but not sufficient for a global optimum of control. For example, examine the point

$$X_1 = .6667$$

$$X_2 = -.500$$

$$X_3 = 1.00 \quad .$$

From this point, the control sequences

$$U_1 = -1$$

$$U_2 = +1$$

and

$$U_1 = 0, \ -1$$

$$U_2 = +1, \ 0$$

are both equally optimal in driving the systems to the origin. The Maximum Principle hinted at this possibility since, at this point, both

$$P_2 = +1$$

$$P_3 = +1 \quad .$$

The control law was indeterminate at this point, and obviously, a singular control could and did exist.

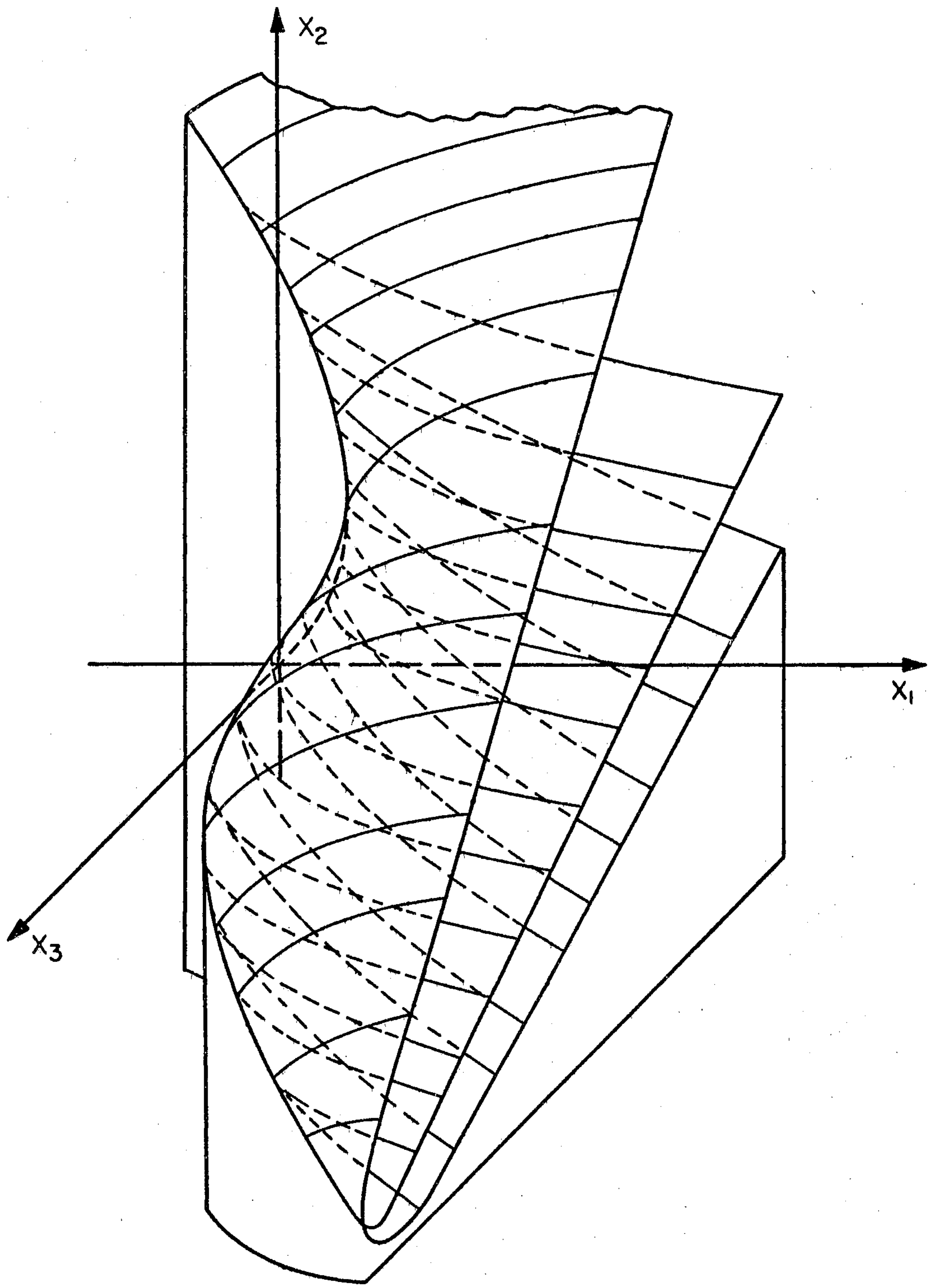



Figure 46. Optimal Switching Surfaces for $U_2$ for Problem 5.4

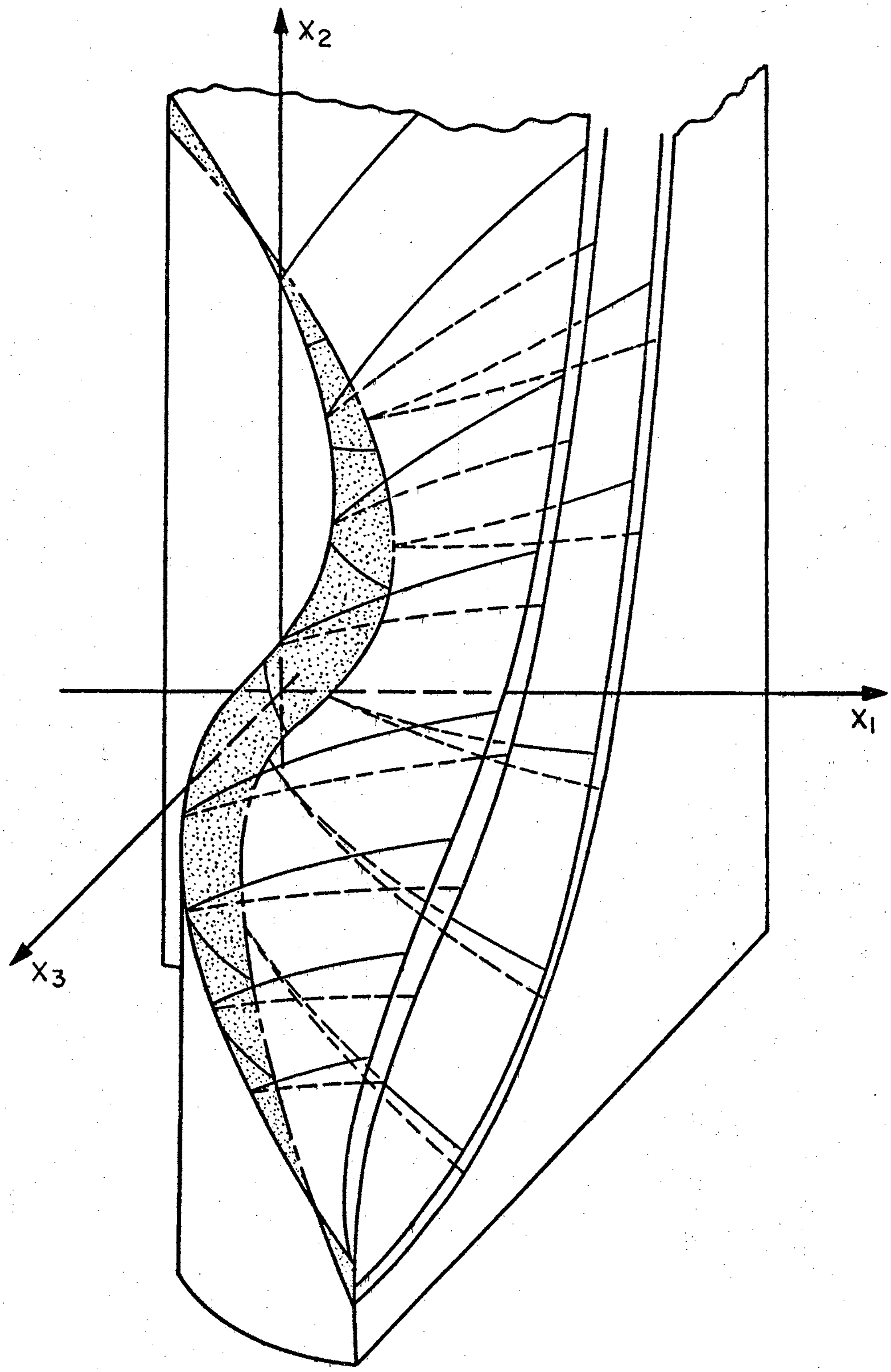



Figure 47. Optimal Switching Surfaces for $U_1$ for Problem 5.4

CHAPTER VI

# PRINCIPLE RESULTS

In the preceding chapters, a new control concept, Optimal Discrete Level Control, has been presented. Not only was a theoretical basis for ODLC developed, but a synthesis procedure, using ODLC, for design of closed-loop controllers was outlined. In Chapter V, this synthesis procedure was applied to several example problems to illustrate the technique. During development of the synthesis procedure and during application of the procedure to problems, a number of insights and results were obtained which merit further discussion. This discussion is divided into four areas: aspects of problem formulation, design guidelines, useful characteristics, and the computational algorithm used.

## Aspects of Problem Formulation

In the first category, problem formulation, there are three primary factors which must be considered: the number of discrete levels used, the form of the performance index used, and the approximation used to functionally represent the ODLC surface for synthesis. Each factor is important and merits discussion.

The number of levels used is a very problem-oriented factor and is quite variable in its effect. Generally, it can be stated that the addition of extra levels will never deteriorate controller performance,

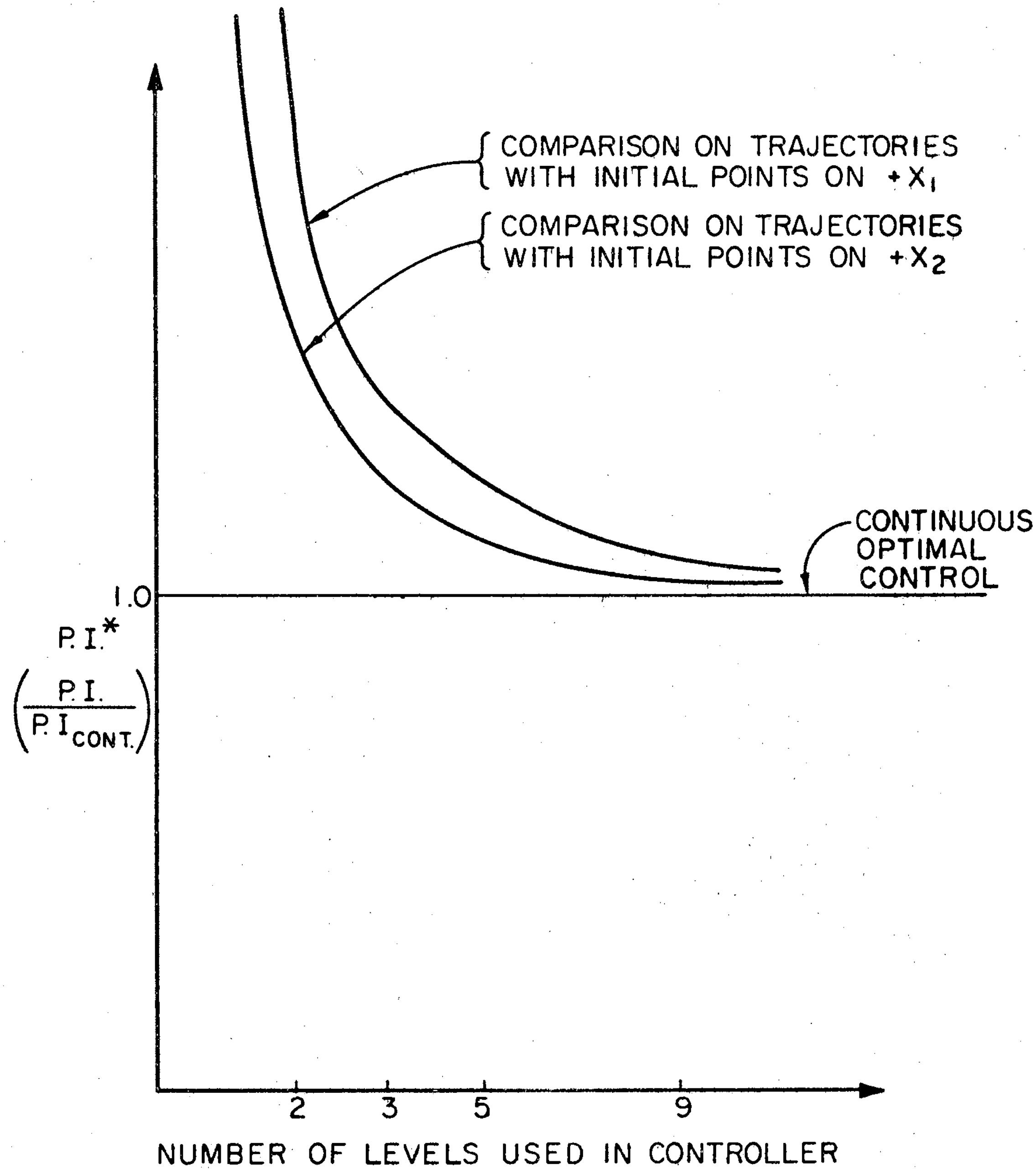



Figure 48. Comparison of the Performance From 2, 3, 5, 9, and Infinite Level ODLC for Problem 5.1

but may improve it. For most systems, the use of only a few levels will provide performance nearly comparable to the use of many levels. The addition of extra levels would affect the system performance insignificantly. This is illustrated in Figure 29 where Problem 5-1 is shown reworked with two-level, five-level, nine-level, and continuous control signals. Since any such comparison will be highly dependent on the initial location of the system, initial conditions were taken along both the $X_1$ and $X_2$ axis to obtain the results shown.

Notice that the significant gains in performance resulted from increasing the number of allowable levels from tow to three or from three to five levels. The gain to be realized over five-level control by nine or more levels would probably not be worth the extra controller complexity required for implementation. It would be impossible to form any general rules for choosing the best number of discrete levels on the basis of the limited amount of investigation carried out in this thesis. However, certain trends were noted which can provide some guidance. First, only a relatively few levels (e.g., five) will be sufficient for most applications. The added complexity of implementing additional levels is unjustifiable from the standpoint of system performance improvements. Second, an extremely important factor in the choice of the number of levels to be used is of the actual magnitudes. This selection of the discrete level magnitudes can profoundly affect the behavior of an ODLC controller. This influence can be illustrated by an extreme example. Consider the case of a five-level ODLC system where the level magnitudes are all chosen to be extremely large values. For this case the controller must use large control effort for all system action, regardless of the magnitude of the action. In so doing, it will consume excessive control effort. Conversely, for the same system, if the five levels are all chosen to be very small in magnitude, the system will be excessively sluggish in response. Obviously there must be some intermediate choice of values for the levels which will provide better control.

In the work carried out to date, sizing of the ODLC magnitudes has been done on the basis of judgment for choosing reasonable levels. Also, the levels have been chosen to be evenly-spaced (i.e., 1,2,3, ..., etc.), with no attempt made to discover what an optimal spacing would be. Obviously, an optimal spacing of levels might produce more improvement in system performance than that produced by increases in the number of levels for an arbitrarily-spaced ODLC controller. Presently, however, comparisons simply cannot be made because optimal-spacing criteria have not been developed and guidelines are not available.

Choice of the performance index is a subtle factor, but it can have a profound influence on both system performance and controller complexity. In fact, choosing a performance index is probably the most important step in the synthesis of any control system. For ODLC systems, the choice of the Performance Index will affect not only system behavior, but also the shape of ODLC switching surfaces (see Figure 37). This can significantly alter the complexity of controller implementation since it is accomplished by approximation of the ODLC surfaces. (In fact, some performance indices may make implementation impossible by causing the switching surfaces to be too complex.)

Ideally, performance indices should be chosen on the basis of both resulting system behavior and resulting controller complexity. However, the concept of what actually constitutes the ideal performance index for a given application is not well understood, even for continuous systems.

Curve-fitting affects system performance by making the controller sub-optimal if approximations to the optimal switching surfaces are used. As discussed earlier, these optimal switching surfaces are of two varieties: terminal trajectory and non-trajectory surfaces. Moderately accurate approximation techniques such as Smith's (22) linear segment method are quite suitable for the second type. The first type, terminal trajectory surfaces, however, are extremely critical.

Inaccuracies of approximation <u>for terminal surfaces</u> usually lends to multiple false switchings as the controller attempts to force the system along the approximation rather than its natural trajectory. This is a highly undesirable characteristic for a controller to possess. Thus, terminal trajectory surfaces require the use of special treatment if controller action is to be acceptable.

Two reasonable methods of handling the approximation of these special surfaces were discussed in Chapter V. First, extremely good surface approximations can be used. An alternate approach is to lessen the criticality of the terminal trajectory surface by definition of an acceptable target region. Then, instead of a single terminal trajectory surface, there will exist a region of acceptable terminal trajectory surfaces. If the approximation is made sufficiently accurate to lie within this region, then purposely-introduced switching hysteresis will produce acceptable controllers. Figure 49 illustrates this technique of reducing terminal trajectory criticality for a single second order plant with time-optimal control. Notice that there is a rather direct trade-off between target size (implying terminal trajectory region width) and approximation accuracy. The decision of what constitutes an acceptable trade-off between the two (target size, and approximation accuracy) must remain an item for engineering judgment.

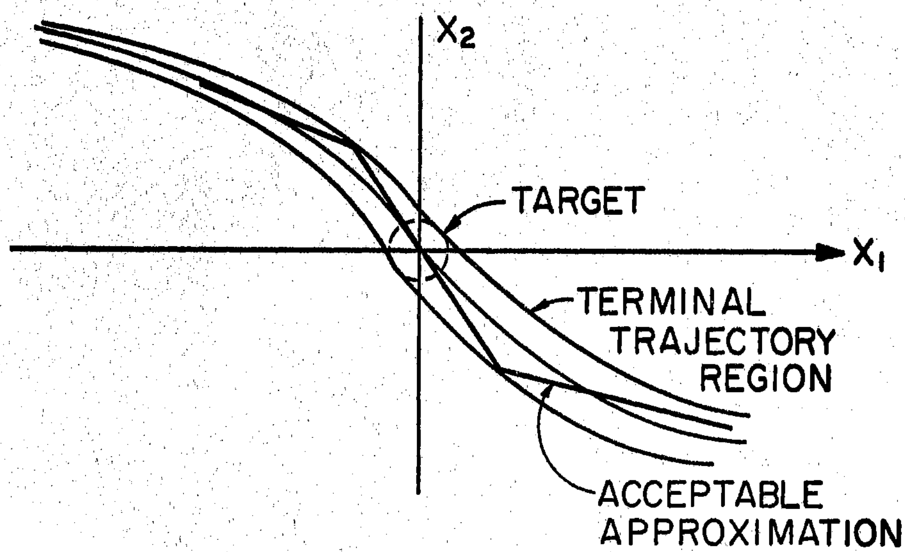



Figure 49. Terminal Trajectory Regions for System Given Terminal with Target

## Design Guidelines

Although the ODLC synthesis procedure has been discussed in some detail, there are several considerations which become important during the actual synthesis. These considerations will be highlighted here.

The synthesis procedure is conceptually a simple process of finding a quantity of optimal switching points; using a curve-fitting method on these points to functionally approximate the optimal surfaces; and translating these functions into hardware. Unfortunately, the procedure is not so simple and straightforward in practice. Large portions of engineering judgement (which must always be the result of trial-and-error or experience) are required to make the results meaningful.

The first area in which engineering judgement crops up is in the determination of points on the optimal surfaces. Since this step is computationally a difficult (and expensive) one, it would be highly desirable to locate only enough points to sufficiently define the surfaces for implementation. Unfortunately, there can be no formal rules about the sufficiency of any number of points in defining a surface. A reasonable rule-of-thumb for third- and lower-order surfaces would be that if enough points have been determined to permit a sketch of the surface to be made, the number is sufficient. For higher-order surfaces, the problem becomes much more difficult. For these higher-order cases an idea of sufficiency of the number of points can be obtained only from the degree of success in obtaining a good functional curve-fit which provides close-to-optimal behavior. In passing, it should be noted that since a number of truly optimal trajectories have been determined such a comparison of the approximate controller behavior to optimal behavior is an easy and natural step of the synthesis.

Another area of concern requiring judgement is determining the most desirable number and/or magnitude of discrete control levels to be used. Although the topic has been discussed previously, during actual synthesis, it must be resolved by trial-and-error, or judgement. The most natural point in synthesis to explore the effects of additional levels is during the determination-of-optimal-points phase. Each location of a point actually involves making an optimal solution with the attendant performance index value. Spot checks on the effect of changing numbers and/or magnitudes of control levels will provide the designer with some idea of the desirability of using various combinations of numbers and/or magnitudes of discrete levels in the controller.

A final area (also previously discussed) involving engineering judgement, is the desired (or necessary) quality obtained from the curve-fit procedure. It has already been pointed out that terminal-trajectory surfaces will be more critical than non-terminal surfaces. During actual synthesis, this information simply means that, if the designer is to avoid unacceptable limit-cycle behavior about the desired operating-point, he must take special care to improve or modify the curve-fit in these critical regions. Several possible approaches to accomplish this improvement/modification have been proposed, but it remains the responsibility of the designer to assure that a proper technique is carried out.

To summarize, in spite of the fact that the ODLC synthesis procedure outlined is primarily analytic in nature, the decisions of the designer during the process are crucial. His engineering judgement will determine whether the system synthesized will or will not perform satisfactorily.

## Useful Characteristics

ODLC systems have a number of useful characteristics. The most important of these characteristics is the feasibility of producing optimal (discrete level) closed-loop controllers. This is a quite important characteristic of the switching surface nature of ODLC controllers. No other general technique presently exists for producing optimal closed-loop controllers. The switching surface nature of ODLC controllers is also responsible for some other characteristics which can be incorporated in the ODLC controller. These characteristics are basically extensions of the switching surface concept to handle systems with: non-linear plants, time-varying plants, and/or non-antonomous inputs. These extensions are all based on the concept of adding new dimensions to the state space in which the ODLC surface lies. By adding the appropriate new dimensions a single, non-moving, non-changing, set of surfaces will provide the closed-loop ODLC law for the system. Naturally, the order of the surface will have become greater, making implementation of the ODLC surface more complex. The above three characteristics were discussed in some detail in Chapter IV and will not be discussed further here.

Another characteristic which can be introduced by this idea of extension of the state space became obvious during work on Problem 5-4. This characteristic is that approximately continuous optimal closed-loop controllers can be designed by use of ODLC synthesis methods. In this case, however, the new pseudo-state variables which are added onto the plant equations are the higher-order time derivatives of the inputs. By using this approach, it becomes possible to produce an ODLC type of controller which is capable of providing not just step inputs to the plant, but also ramp (using only first derivative term), parabolic (first and second derivatives), and higher-order input types. The justification for this extension can best be illustrated by reviewing Problem 5-4. Recall that the plant equation was

$$\dot{X}_1 = X_2$$
$$\dot{X}_2 = -X_3 + U_2$$
$$\dot{X}_3 = -U_1 \quad , \tag{6.1}$$

and the performance index

$$P.I. = \int_{t_o}^{t_f} (1 + U_1^2 + U_2^2)dt \tag{6.2}$$

was used. Inputs were constrained to

$$U_1 \epsilon \; [-2, -1, 0, +1, +2]$$
$$U_2 \epsilon \; [-2, -1, 0, +1, +2] \quad . \tag{6.3}$$

Although this system is nominally a third-order system with two inputs, the input $U_1$ and the state variable $X_3$ bear further investigation. Consider the fact that

$$\dot{X}_3 = -U_1 \quad . \tag{6.4}$$

Since $U_1$ is constrained to be one of five constant amplitude levels, during the periods of time between switching events Equation (6.4) can be integrated with respect to time. The result is that $X_3$ will be a ramp function with one of five possible slopes. In light of this fact, the third-order system (Equation (5.25)) can be written as a second-order equivalent system

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = \int U_1 dt + U_2 \quad , \tag{6.5}$$

or

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = U_e \quad . \tag{6.6}$$

The input ($U_e$) is capable not only of discrete level changes but of discrete slope changes. Because it is composed not only of step functions but also of ramp functions, $U_e$ will be capable of both switching-type and approximately continuous behavior.

This result is important, an optimal controller capable of both switch-type and nearly continuous behavior has been specified and implemented through the use of ODLC techniques. An immediate extension which comes to mind is the addition of new higher-order inputs as pseudo-state variables. With these higher-order terms, plant inputs would be composed of step, ramp, parabolic, ..., etc., terms and would be some sort of optimal control (closed-loop) signal capable of both smoothly continuous and discontinuous (step-jumps) behavior. Notice the key words "optimal," "continuous," and "closed-loop," for it is precisely this type of controller which can be specified via the extended ΘDLC design procedures.

## Computer Algorithm

During the research work, a computer algorithm was developed to solve the multi-point boundary-value problems which arise as a consequence of applying optimal control theory to ODLC. Although this algorithm is not an integral part of ODLC theory, it provides an essential tool in the synthesis of systems by the use of ODLC methods. However, the algorithm has applicability to a broad range of problems outside the realm of ODLC. Consequently, the algorithm is listed as a contribution (secondary) on its own merit. Since it is not within the intent or scope of a thesis to become a program-listing document or an instruction manual, programming details will not be presented. However, in addition to the discussion of the basic concepts of the algorithm found in Chapter IV, Appendix II contains a schematic flow chart defining the logic used in this algorithm.[1] The present discussion will be limited to a discussion of problem types to which the algorithm is well suited and some basic limitations of the approach.

Because the approach is basically a restricted-search Dynamic Programming procedure, the algorithm possesses nearly all the characteristics of straight Dynamic Programming with the following differences.

First, because it is a limited-search method it will require less core storage than conventional Dynamic Programming. As an example, consider the illustration given in Chapter II, namely a fifth-order

---

[1]Address all inquiries for further documentation, including program listings to either Professor H. R. Sebesta, Mechanical Engineering Department, Oklahoma State University, Stillwater, Oklahoma, or J. D. Engelland, Aerosystems Project Engineer, Dept. 62-7330, General Dynamics, Fort Worth, Texas.

system with each state variable domain divided into 100 grid points (N) and time divided into 100 (increments (M). The storage location requirement of the conventional technique is

$$P = 2 \times 10^{12} \quad .$$

In the iterative approach, an equivalent accuracy can be achieved by

$$P = 4 \times 10^{4} \quad .$$

Obviously, restricting the space searched reduces the number of storage allocations drastically.

The second characteristic of the new approach is also a direct result of limiting the searched space. Because only a tube is searched on each iteration, the algorithm at any stage will find only the best solution (if one exists) in this tube. Further, as it iteratively constructs new tubes about improved solutions, the algorithm may be converging to a locally optimal solution rather than the global optimal one. In this respect, since it will not guarantee finding the global optimal, the new technique is not as powerful as straight dynamic programming. Thus, reduction of the storage requirements has been achieved at the cost of guaranteed global optimality. Gained, however, is the ability to use a Dynamic Programming approach to a wide range of multiple boundary-value problems.

Apart from these differences the approach retains the important characteristics of dynamic programming, namely:

(1) Inputs do not need to be continuous.

(2) System equations are not required to be linear or even continuous.

(3) Hard constraints can be imposed on both the system variables and inputs.

These three characteristics make the approach attractive for use in ODLC synthesis. Further, they make the approach equally attractive for highly constrained problems which are not easily solved by any other method. Examples of these highly constrained problems might be hydraulic systems with system equations which actually change form as a function of valve displacements; or mechanical systems with hard-stops; or systems with pulsed-type control signals.

CHAPTER VII

# SUMMARY

The topics investigated are the new concept of Optimal Discrete Level Control (ODLC) and synthesis of ODLC controllers for dynamical systems. ODLC controllers are optimal controllers with the added restraint that the controller outputs (plant inputs) are constrained to assume only certain magnitude levels. Thus, the ODLC's are always switching controllers but, unlike many switching controllers, they are optimal within the constraints placed on them (discrete plant input levels).

Because the ODLC controller is a switching controller, the corresponding ODLC control law can be shown to be simply a switching surface in state space. Functional implemention of this surface results in complete specification of a closed-loop ODLC controller. This is a very important result; closed-loop optimal controllers of any type are very difficult to define. A number of applications for which ODLC should be extremely useful can be envisioned, however, an important area of application should be for systems where the control effort is inherently discrete in nature (e.g., the maneuvering thrusters on a space vehicle). Further, because ODLC possess the characteristics of rather good noise immunity, and simplicity of closed-loop implementation, ODLC controllers will also be important where these characteristics are important.

Investigation was carried out in two primary areas. First, a theoretical basis for the ODLC concept was described, derived, and defined. This included verification of the applicability of the Maximum Principle for specification of necessary conditions for optimality, examination of the ODLC switching surfaces, and definition of the limitations placed on the type of plants and control schemes suitable for implementation. The ODLC switching surfaces are discussed in the context of the autonomous systems. Techniques for extending the ODLC concept to cover non-autonomous systems are developed.

The second area of investigation was a controller synthesis technique which evolved into a three step procedure:

(1) Describing the plant and deriving the conditions necessary for optimality.

(2) Locating a sufficient number of points on the ODLC surfaces to adequately define them.

(3) Approximating the ODLC surfaces with an appropriate function by using these points.

This design procedure not only allows the designer the flexibility to arbitrarily choose a performance index, but also to choose the discrete input magnitude levels he deems desirable or practical. An additional inherent characteristic of the procedure, if it is being used to design a closed-loop controller by approximation to the optimal switching surface, is that the true optimal behavior for a number of initial conditions has been determined. This availability of the optimum can be used to evaluate the performance of the approximating controller. By having such a comparison available, the designer can more intelligently determine a reasonable compromise between the conflicting attributes of controller optimality and simplicity.

Three example problems have been worked to illustrate typical characteristics of ODLC. Not only were the ODLC surfaces defined for these three systems but simulation of implemented closed-loop ODLC controllers was carried out and the results analyzed. It is felt that these results illustrate both the power and utility of system synthesis using the concept of ODLC.

## AREAS RECOMMENDED FOR FURTHER STUDY

Because the scope of research for this thesis was necessarily limited, there are a number of areas in which further research should be carried on.

(1) New, improved, numerical solution methods for solving the extremely difficult class of problems which arise during the synthesis procedure are needed. The iterative dynamic programming approach which was developed during the research presently has several shortcomings which should be eliminated, although the validity of the basic concept has been established. The primary difficulty encountered with the present formulation of the algorithm is convergence. Tied up in this problem are such factors as grid size, time step size, the number of grid points, and terminal constraints. It is felt that too much juggling of these factors is presently needed to obtain convergence. An improved convergence approach which properly handles these factors is needed for this dynamic programming method to be made truly useful.

(2) Optimal quantization of discrete levels should be investigated. In the present work, arbitrary discrete levels were chosen. Certainly, there exist some set of level values which provide better control behavior than any other. These level values are initial-condition-sensitive, so optimum quantization likely will involve probability distributions for these initial conditions.

(3) Investigation should be made of other approximating techniques which can be used to describe the ODLC surfaces. There are a number of standard techniques available, but perhaps some new method can be developed which is superior for ODLC.

(4) Studies of the effects of different performance indices on system behavior and ODLC surface complexity should be made.

(5) Another area of promise is investigation of the closed-loop ODLC switching surfaces for implementing continuous plant control. It would appear possible that reasonable approximations to the optimal continuous control might be obtained by interpolation of the input values as a function of position relative to switching surfaces. A most attractive alternate approach to implementing approximately continuous control is that discussed in Problem 5-4. This concept which produces an effective input capable of step, ramp, second-order, etc., changes by the introduction of higher-order input terms, should produce extremely good controllers regardless of whether they are continuous or switching (or both) in nature. If feasible, these techniques will provide

a general closed-loop controller synthesis method for continuous control. Such general capability is presently lacking.

## Contributions

It is believed that several significant contributions have been made. First, the new concept of optimal discrete level control has been proposed and theoretically verified. ODLC controllers show great promise for systems where control effort must be discrete in level or for systems where high-quality, inexpensive closed-loop control is required.

Second, since ODLC systems are switching systems, the characteristics of optimal switching surfaces were examined and defined. This effort not only clearly detailed ODLC surface characteristics but also presented switching surfaces in general from a new theoretical viewpoint.

A third contribution is the development of the concept for a computer algorithm, based on dynamic programming, for solution of highly constrained multiple point boundary value problems. This algorithm was developed as a means for determining points on the ODLC switching surfaces, but is applicable to the broad range of very difficult-to-solve problems consisting of multiply-split boundary value problems. Although the actual algorithm needs further improvement, the basic concept has been shown to be a valid approach to solution of these problems.

Finally, the total concept of the ODLC closed-loop synthesis procedure is viewed as a significant contribution which will aid the designer in his task of building better controls for his machines.

## BIBLIOGRAPHY


1. Apostol, Tom M. Mathematical Analysis: A Modern Approach to Advanced Calculus. Massachusetts: Addison-Wesley Publishing Company, Inc., 1957.

2. Athans, Michael, and Peter L. Falb. Optimal Control: An Introduction to the Theory and Its Applications. New York: McGraw-Hill Book Company, 1966.

3. Bellman, R. E., and R. E. Kalaba, editors. Modern Analytical and Computational Methods in Science and Mathematics. New York: American Elsevier Publishing Company, Inc., 1965.

4. Bellman, Richard. Introduction to the Mathematical Theory of Control Processes, Volume I. New York and London: Academic Press, 1967.

5. Chang, Chi S. "Discrete-Sample Curve Fitting Using Chebyshev Polynomials and the Approximate Determination of Optimal Trajectories Via Dynamic Programming." IEE Transactions on Automatic Control, January 1966, pp. 116-118.

6. Davis, Duane M. "The Iterative Dynamic Programming Solution of a Class of Optimal Regulator Problems." (Ph.D. thesis, Purdue University, 1967.)

7. Dreyfus, Stuart E. Dynamic Programming and the Calculus of Variations. New York and London: Academic Press, 1965.

8. Flugge-Lotz, Irmgard, and Mih Yin. "The Optimum Response of Second-Order, Velocity-Controlled Systems With Contractor Control." ASME Paper No. 60-JAC-3.

9. Frederick, Dean K., and Gene F. Franklin. "Design of Piecewise Linear Switching Functions for Relay Control Systems." IEEE Transactions on Automatic Control, Volume AC-12, No. 4, August 1967, pp. 380-387.

10. Ibragimov, I. S. "Synthesis of a Quasioptimal System With Two Controllers by Means of an Approximation of the Optimal Switching Surfaces." Translated from Automatika i Telemekhanika, No. 2, February 1967, pp. 59-68.

11. Kirk, D. E. "Optimization of Systems With Pulse-Width Modulated Control." IEEE Transactions on Automatic Control, June 1967, pp. 307-309.

12. Krut'Ko, P. E. "Analytical Design of Optimum Linear Digital Regulators." Translated from Automatika i Telemekhanika, Volume 27, No. 2, February 1966, pp. 109-112.

13. Larson, Robert E. "A Dynamic Programming Successive Approximations Technique." Joint Automatic Controls Conference, 1968, pp. 1071-1080.

14. Loos, C. H. "Time-Proportional Control." Control Engineering, May 1965, pp. 65-71.

15. Nardizzi, L. R., and George A. Bekey. "Optimization of Discrete Time Systems With Combined Modulation." IEEE Transactions on Automatic Control, June 1968, pp. 286-289.

16. Nemhauser, George L. Introduction to Dynamic Programming. New York: John Wiley and Sons, Inc., 1966.

17. Osborn, R. F. "Time Domain Compensation of Nonlinear Systems." (Ph.D. dissertation, Mechanical and Aerospace Engineering, Oklahoma State University, August 1969.)

18. Pavlidis, T. "Optimal Control of Pulse Frequency Modulated Systems." IEEE Transactions on Automatic Control, Volume AC-11, No. 4, October 1966, pp. 666-684.

19. Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, E. F. Mischenko. The Mathematical Theory of Optimal Processes. New York: The Macmillan Company, 1964.

20. Rozonoer, L. I. "Maximum Principle in the Theory of Optimum Systems." Automation and Remote Control, Moscow, 1958.

21. Skoglund, Victor J. Similitude. (Unpublished text, Mechanical Engineering Department, University of New Mexico.)

22. Smith, Fred W. "Design of Quasi-Optimal Minimum-Time Controllers." IEEE Transactions on Automatic Control, Volume AC-11, No. 1, January 1966.

23. Tou, Julius T. Optimal Design of Digital Control Systems. New York and London: Academic Press, 1963.

24. Van Pelt, R. W. "Digital Control Techniques in Closing Process Loops." ISA Transactions, October 1965, pp. 325-330.

## APPENDIX I

## DERIVATION OF PONTRYAGIN'S MAXIMUM PRINCIPLE

The Maximum Principle is a statement of necessary conditions for the optimality, according to a given performance index, of the behavior of a dynamical system. The approach used for the derivation will be to consider first, a simple class of problems and then to consider extensions and modifications necessary for more general cases. The simple case to be considered first will be that of a dynamical system with fixed final time and no constraints of final time on the variables.

Assume the system can be defined by the set of equations

$$\dot{X}_i = f_i(\underline{X},\underline{U},t) \quad ; \quad (i = 1, \ldots, n) .$$ [1]

Further, assume that the following exist:

$$\frac{\partial f_i}{\partial X_j} \qquad \begin{array}{l} i = 1, \ldots, n \\ j = 1, \ldots, n \end{array}$$

$$\frac{\partial^2 f_i}{\partial X_1 \partial X_k} \qquad \begin{array}{l} i = 1, \ldots, n \\ j = 1, \ldots, n \\ k = 1, \ldots, n \end{array} .$$

Allowable inputs U(t) will be defined to be at least piecewise-continuous elements of a closed, bounded set U.

---

[1] State variable notation and concepts will be used throughout this derivation.

Let the performance index to be minimized be

$$P.I. = \int_{o}^{T} J(\underline{X},\underline{U},t)dt \quad .$$

If the definition is made that

$$X_{n+1} = \int_{o}^{T} J(\underline{X},\underline{U},t)dt$$

then

$$\dot{X}_{n+1} = J(\underline{X},\underline{U},t) \quad .$$

Finally, let there be defined, a new function of the final time

$$S = \langle \underline{C} \cdot X'(T_f) \rangle$$

where $\underline{C}$ is some constant vector and S is the dot product of $\underline{C}$ with $\underline{X}'(T_f)$ an expanded state vector containing $P.I. = X_{n+1}$ as a component. S is called Pontryagin's payoff function. It can be shown that by proper definition of $\underline{C}$, it is always possible to minimize the performance index by minimization of S. Thus, the optimization can be made in terms of the minimizations of this payoff function S. As an example, for the simple case under consideration, if

$$\underline{C} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad \text{and} \quad \underline{X}(T_f) = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n+1} \end{bmatrix}$$

then

$$S = \int_{o}^{T_f} J(\underline{X},\underline{U},t)dt \quad .$$

S is the performance index.

Now, assuming that the admissible optimal control has by some mean means been obtained, perturbations around this control will be studied. These perturbations will take the form shown in Figure 50. The definition of these perturbations is that during a short interval of time $\tau-\epsilon \leq t \leq \tau$, the control is different from $U(t)^*$. Restrictions on the perturbation are:

1. $U(t)$ is different from $U(t)^*$ in the interval $(\tau-\epsilon, \tau)$.
2. $U(t)$ is admissible in the interval of interest.
3. $U(t)$ is constant during this interval.

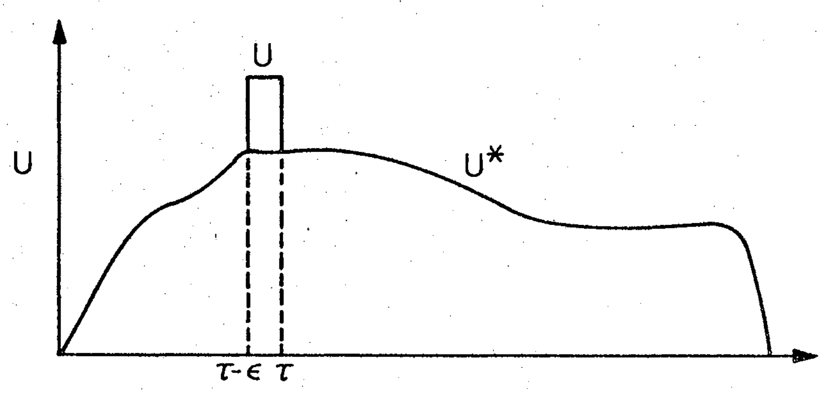



Figure 50. Perturbations of Optimal Control

Now, since $\underline{U}(t)$ is different from $\underline{U}(t)^*$ in the interval $(\tau-\epsilon \leq t \leq \tau)$, $\underline{X}(\tau)$ will be different from $\underline{X}(\tau)^*$. This difference can be defined as

$$\underline{X}(\tau) - \underline{X}(\tau)^* = \int_{\tau-\epsilon}^{\tau} \left[ \underline{f}(\underline{X}, \underline{U}+\delta\underline{U}, \tau) - \underline{f}(\underline{X}^*, U^*, \tau) \right] dt \ .$$

To a first approximation, this is

$$\underline{\delta X}(\tau) = [\underline{f}(\underline{X},\underline{U} + \underline{\delta U}, \tau) - \underline{f}(X^*,\underline{U}^*,\tau)]\,\epsilon.$$

This relationship provides the necessary initial conditions at the end of the perturbation at time $\tau$ to carry out integration to final time. However, if the perturbed state variables can be assumed to remain "close" to the optimal ones, certain simplifications can be made. If this is the case, then the perturbed state vector, which is

$$\frac{d(X + \delta X)_i}{dt} = \underline{f}_i(\underline{X} + \underline{\delta X}, \underline{U}^*, t) \qquad \begin{matrix} (i = 1, \ldots n+1) \\ (\tau \leq t \leq T_f) \end{matrix}$$

can be expanded in a Taylor Series. This result is

$$\frac{d(X + \delta X)_i}{dt} = \Big[ f_i(\underline{X},\underline{U}^*,t) + \sum_{j=1}^{n+1} \frac{\partial f_i(\underline{X},\underline{U}^*,t)}{\partial X_i} \delta X_j$$

$$+ \tfrac{1}{2} \sum_{j=1}^{n+1} \sum_{s=1}^{n+1} \frac{\partial f_i(\underline{X} + \underline{\xi}\ \delta X,\underline{U}^* t)}{\partial X_j\ \partial X_s} \delta X_d\ \delta X_s \Big]$$

$$i = 1, \ldots, n+1$$

where $0 \leq \xi \leq 1$.

However,

$$\frac{d(X + \delta X)_i}{dt} = \frac{dX_i}{dt} + \frac{d(\delta X)_i}{dt} \qquad i = 1, \ldots, n+1$$

and

$$\frac{d(X + \delta X)_i}{dt} = f_i(\underline{X},\underline{U}^*,t) + \frac{d(\delta X)_i}{dt} \qquad i = 1, \ldots, n+1.$$

This results in

$$\delta \dot{X}_i(t) = \sum_{j=1}^{n+1} \frac{\partial f_i(\underline{X},\underline{U},t)}{\partial X_j} \delta X_j + \text{H.O.T.}$$

Further, since it was assumed that the perturbation effect on the vector remains "small", the higher order term can be neglected. Thus

$$\delta\dot{X}_i \triangleq \sum_{j=1}^{n+1} \frac{\partial f_i(\underline{X},\underline{U}^*,t)}{\partial X_j} \delta X_j. \qquad \begin{array}{l}(i = 1, \ldots, n+1)\\ (\tau \leq t \leq \tau_f)\end{array}$$

This is a linear set of differential perturbation equations with initial conditions at $\tau$ of

$$\delta \underline{X}(\tau) = [\underline{f}(\underline{X},\underline{U}^* + \underline{\delta U},t) - f(\underline{X},\underline{U}^*,t)]\delta t.$$

Recall that the payoff function is

$$S = \sum_{j=1}^{n+1} C_j X_j(T_s).$$

The variation of this equation is

$$\delta S = \sum_{j=1}^{n+1} C_j \, \delta X_j(T_f).$$

Now, since $\underline{U}^*(t)$ was the optimal control, for all $\underline{U}(t) + \underline{\delta U}$ different from this

$$\delta S \geq 0,$$

or

$$\sum_{j=1}^{n+1} C_j \, \delta X_j(T_f) \geq 0.$$

This result is a necessary condition for optimality and could be used to determine the effect on optimality of any perturbation at time t in the interval

$$\tau - \epsilon \leq t \leq \tau,$$

by integrating from time $\tau$ to $T_f$. This means a complete solution for $\delta \underline{X}(t)$, $(\tau \leq t \leq T_f)$ would be necessary. On the other hand, if a variable vector $\underline{P}(t)$ could be found such that $P(T_f) = -\underline{C}$ and $\langle \underline{P}(t) \cdot \delta \underline{X}(t) \rangle$ = constant, then this integration would not be required.

Now, let $P(t)$ be examined to see what it must be. The total derivative of the above inner product is

$$\frac{d}{dt} \langle \underline{P}(t) \circ \underline{\delta X}(t) \rangle = 0 .$$

This can be written

$$\sum_{i=1}^{n+1} \overset{\circ}{P}_i(t) \circ \delta X_i(t) + \sum_{j=1}^{n+1} P_j(t) \delta \overset{\circ}{X}_j(t) = 0.$$

But

$$\delta \dot{X}_j(t) = \sum_{k=1}^{n+1} \frac{\partial f_j(\underline{X}, \underline{U}, t)}{\partial X_k} \delta X_k.$$

Thus,

$$0 = \sum_{i=1}^{n+1} \overset{\circ}{P}_i(t) \; \delta X_i(t) + \sum_{j=1}^{n+1} \sum_{k=1}^{n+1} P_j(t) \frac{\partial f_j(\underline{X}, \underline{U}, t)}{\partial X_k} \delta X_k.$$

The inner-change of summation signs in the second part, with replacement of k with i gives

$$0 = \sum_{i=1}^{n+1} \left[ \overset{\circ}{P}_i(t) + \sum_{j=1}^{n+1} P_j(t) \circ \frac{\partial f_j(\underline{X}, \underline{U}, t)}{\partial X_i} \right] \delta X_i(t).$$

Since $\delta X_i(t) = 0$ is a trivial solution to the above equation

$$0 \triangleq \dot{P}_i(t) + \sum_{j=1}^{n+1} P_j(t) \frac{\partial f_j(\underline{X},\underline{U},t)}{\partial X_i} .$$

Thus,

$$\dot{P}_i(t) = - \sum_{j=1}^{n+1} P_j(t) \frac{\partial f_j(\underline{X},\underline{U},t)}{\partial X_i} .$$

This is recognized as the well-known adjoint system to the original system. There also are available the conditions $\underline{P}(T_f) = -\underline{C}$. These provide the necessary boundary conditions for solution of the adjoints. However, there are $2n + 2 + M$ variables and only $2n + 2$ boundary conditions with $2n + 2$ differential equations. M additional algebraic equations are needed. These can be found in the following manner. Define the function H as:

$$H = \langle \underline{f}(\underline{X},\underline{U},t) \cdot \underline{P}(t) \rangle .$$

It was previously stated that

$$- \delta S = \langle \delta X(\tau) \circ P(\tau) \rangle \leq 0 .$$

But

$$\delta \underline{X}(\tau) = [\underline{f}(\underline{X}(\tau), \underline{U}(\tau) + \underline{\delta U}, \tau) - f(X^*(\tau), U^*(\tau), \tau)]\epsilon .$$

Therefore, $$- \frac{\delta S}{\delta t} = \langle \underline{f}(X(\tau), \underline{U}(\tau) + \underline{\delta U}, \tau) \circ \underline{P}(\tau) \rangle$$

$$- \langle \underline{f}(X(\tau), U^*(\tau), \tau) \circ \underline{P}(\tau) \rangle \leq 0 .$$

Since this inner product is constant throughout $\tau \leq t \leq T_f$

$$-\frac{\delta S}{\delta t} = \langle f(\underline{X}(t),\underline{U}(t) + \delta\underline{U},t) \cdot \underline{P}(t)\rangle$$

or

$$-\langle f(\underline{X}(t),\underline{U}^*(t)) \cdot \underline{P}(t)\rangle \leq 0.$$

Therefore,

$$H_u - H_u^* \leq 0.$$

$$H_u \leq H_u^*.$$

Thus, for minimum S

$$H^* = \max_{U \in U} H.$$

This is the Maximum Principle of Pontryagin and provides the m additional equations required. The above derived equations can be recast into a form which are easier to use; namely,

Define:

$$H = \langle f(\underline{X},\underline{U},t) \cdot \underline{P}(t)\rangle$$

then it can easily be shown that

$$-\frac{\partial H}{\partial X_i} = \dot{P}_i \quad , \quad +\frac{\partial H}{\partial P_i} = \dot{X}_i.$$

These are the canonical equations of Pontryagin. Also, there are

$$H^* = \max_{U \in U} H$$

and the boundary conditions,

$$X_i(0) = X_{io} \qquad i = 1, \ldots, n + 1$$

$$P_i(T) = -C_i \qquad i = 1, \ldots, n + 1$$

As a group, this set of relationships define the optimal control problem with sufficient restraints to solve the resulting set of differential equations.

Extension to More General Cases

The above results can easily be extended in applicability to more general cases. The first such is the case of free final time with no constraints on the state variables at final time. This can be handled very nicely by defining the unknown final time as a new state variable $X_{n+2} = T_f$ = constant. Now, the system of equations will have two new differential equations, but it will also have two new boundary conditions ($P_{n+2}$ (0) = 0, and $P_{n+2}$ ($\tau_f$) = 0), which occur because $X_{n+2}$ is unconstrained at both boundaries. Essentially what has been done by this re-definition is the non-dimensionalization of time so that the time period of interest is always from $0 \leq \tau \leq 1$.

A significant result can be deduced from this new problem statement. Examination indicates that the new $P_{n+2}$ adjoint variable is $\dot{P}_{n+2} = -\frac{H}{X_{n+2}}$. Since both the Hamiltonian (H) and the final time ($X_{n+2}$) are constant with respect to time, $\dot{P}_{n+2}$ can be directly integrated. The result is

$$P_{n+2}(\tau_f) = -\frac{H}{X_{n+2}}(\tau_f).$$

However, $$P_{n+2}(\tau_f) \triangleq 0.$$

Therefore, either $$H \triangleq 0$$

or $$X_{n+2} \triangleq 0.$$

The conclusion must be that $H \triangleq 0$. Further, since $H = X_{n+2}\, H^*$, where $H^*$

is the Hamiltonian for the dimensional case,

$$H^* \stackrel{\Delta}{=} 0.$$

It will always be true that the Hamiltonian will be identically zero for optimal processes with unfixed final times.

The second case of extension is for the case with algebraic constraints on the state variables at final time. This case can be divided into two classes, according to whether the final constraints are fixed-point terminal constraints or terminal surface (or manifold) constraints.

The second class shall be reviewed first. Consider the Pontryagin Payoff function

$$S = \langle \underline{C} \cdot \underline{X}(T_f) \rangle.$$

As was previously discussed, it is necessary that S be minimized for the performance index to be minimized. However, S is an algebraic function of the state variable final conditions. Because of this, the problem of algebraic terminal constraints can be handled by the familiar method of LaGrange multipliers.[1] Using LaGrange multipliers (S) can be redefined to be

$$S' = S(X_{t_f}) + \lambda^T \underline{g}(X),$$

where $\underline{g}(X)$ is the set of constraints expressed in the form $g(\underline{X}(t_f) = 0$, and $\underline{\lambda}^T$ is a vector of ($\underline{m}$) unknown constants. Notice that M new unknowns

[1]The concept of LaGrange multipliers for algebraic minimization is well known and will not be derived herein.

$(\underline{\lambda})$ have been added to the problem. Now M new relationships are needed. These can be obtained by requiring that

$$\frac{\partial S}{\partial X(t_f)_i} = 0 \qquad i = 1, \ldots, n+1.$$

Performing this operation gives

$$0 = \left.\frac{\partial S}{\partial X_i}\right|_{X^*} + \lambda_j \left.\frac{\partial g_j}{\partial X_i}\right|_{X^*\lambda^*} \qquad \begin{array}{l} i = 1, \ldots, n+1 \\ j = 1, \ldots, n+1. \end{array}$$

This provides the required M new relationships.

Now the first class of terminal constraint problems (fixed final-point) shall be considered. For this class of problems, the use of LaGrange multipliers cannot be used because $\frac{\partial g_j}{\partial X_i}$ is indeterminate. However, there actually is no need to attempt to discover new relationships, because the final constraints are boundary conditions for the state variables and these will provide a sufficient number of constraints to completely specify the problem. Mixtures of the two classes of terminal constraint problems can occur but there will always be a sufficient number of constraints to completely specify the problem.

Pontryagin's Maximum Principle has been shown to be applicable to a rather broad range of problems. There are further extensions which can be made, but they are considered to be beyond the scope of this paper. More important, it is obvious from the above derivation that it is valid to use the Maximum Principle for determining necessary conditions for ODLC. Additionally, the primary limitations on system type, etc., have been shown to be:

1. the plant must be of the type $\dot{\underline{X}} = \underline{f}(\underline{X},\underline{U},t)$,

2. $\underline{f}$ must be continuous with respect to $\underline{X}$ through the first derivative, and

3. $\underline{U}$ must at least be a piecewise continuous function (piecewise constant for ODLC) which belongs to some closed, bounded set $\Omega$ of allowable inputs.

These limitations will be considered to be basic to all derivations and discussions contained in the text of this paper.

## APPENDIX II

## MODIFIED ITERATIVE DYNAMIC PROGRAMMING

Work on the dissertation topic, Optimal Discrete-Level Control, revealed a lack of solution algorithms for solution of highly-constrained, multiple-boundary value problems. Since the preceding problem type is characteristic of ODLC formulations, development of a modified Dynamic Programming algorithm to solve problems of this type was deemed necessary. The following is a brief description of that algorithm.

The Modified Dynamic Programming algorithm under discussion differs from classical Dynamic Programming in one primary respect. It, unlike the classical approach, searches only in the immediate vicinity of a previously specified trial solution, instead of all state space. The algorithm then uses the best solution found in restricted-space search to form a new trial solution to iterate upon. Repeated iteration will find a "best" solution. Unlike the "best" solution found by classical dynamic programming, which is always a global optimal (assuming sufficient grid fineness), the iterative algorithm may find only a local optimal solution. This loss of global optimality assurance is the price exacted for rather phenominal reductions in computer memory requirements (from as high as $10^{13}$ to $10^{4 \text{ or } 5}$). Apart from this restriction of searched space to the neighborhood of a trial trajectory, the new algorithm uses the familiar relationships used in classical dynamic programming. (It is assumed as a prerequisite that the reader has working knowledge of dynamic programming techniques.)

The equations used in the algorithm are the discretized $n^{th}$ order plant equations

$$\bar{X}(q+1) = \bar{F}[\bar{X}(q),\ \bar{U}(q),\ t(q)],$$

the discretized performance index

$$\text{P.I.}\ (q) = J[\bar{X}(q),\ \bar{U}(q),\ t(q)]$$

or

$$\text{P.I.} = \sum_{q=1}^{q\ \text{total}} J[\bar{X}(q),\ \bar{U}(q),\ t(q)],$$

and the penalty factor

$$\text{Fac}(q) = \sum_{i=1}^{N} K_1[\,\|X_6 - \text{Lim}_i\|\,],$$

where

$$\text{Lim}_i = \begin{cases} X_{\min_i} & \text{if } X_i < X_{\min_i} \\ X_{\max_i} & \text{if } X_i > X_{\max_i} \\ X_i & \text{if } X_{\min_i} \le X_i \le X_{\max_i}. \end{cases}$$

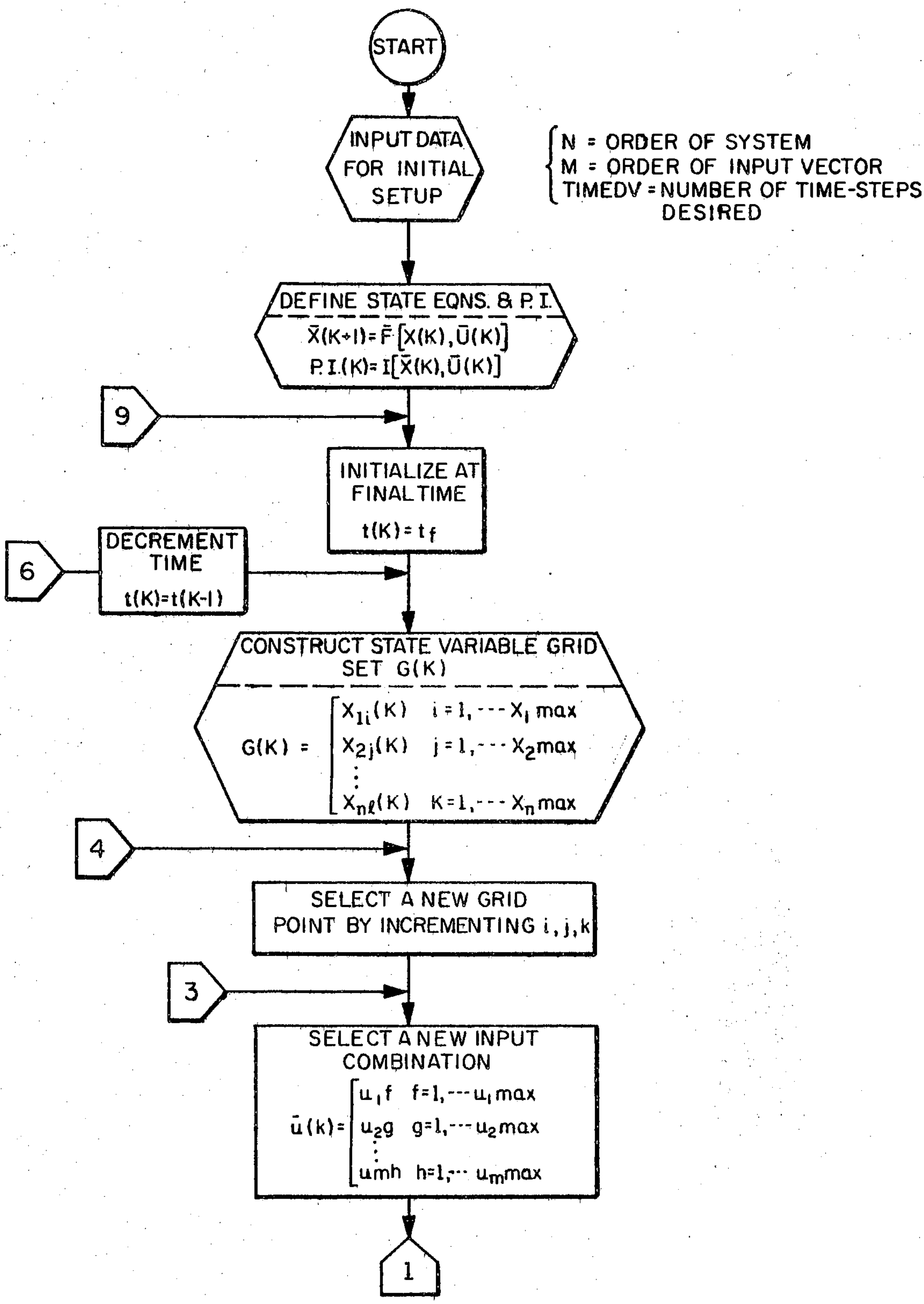
A penalty factor term is required not only to force the trajectory to satisfy system constraints, but, for the iterative algorithm, to force it to remain in the neighborhood of the previous trial trajectory. Such a constraint is necessary state space away from this neighborhood has not been properly searched.

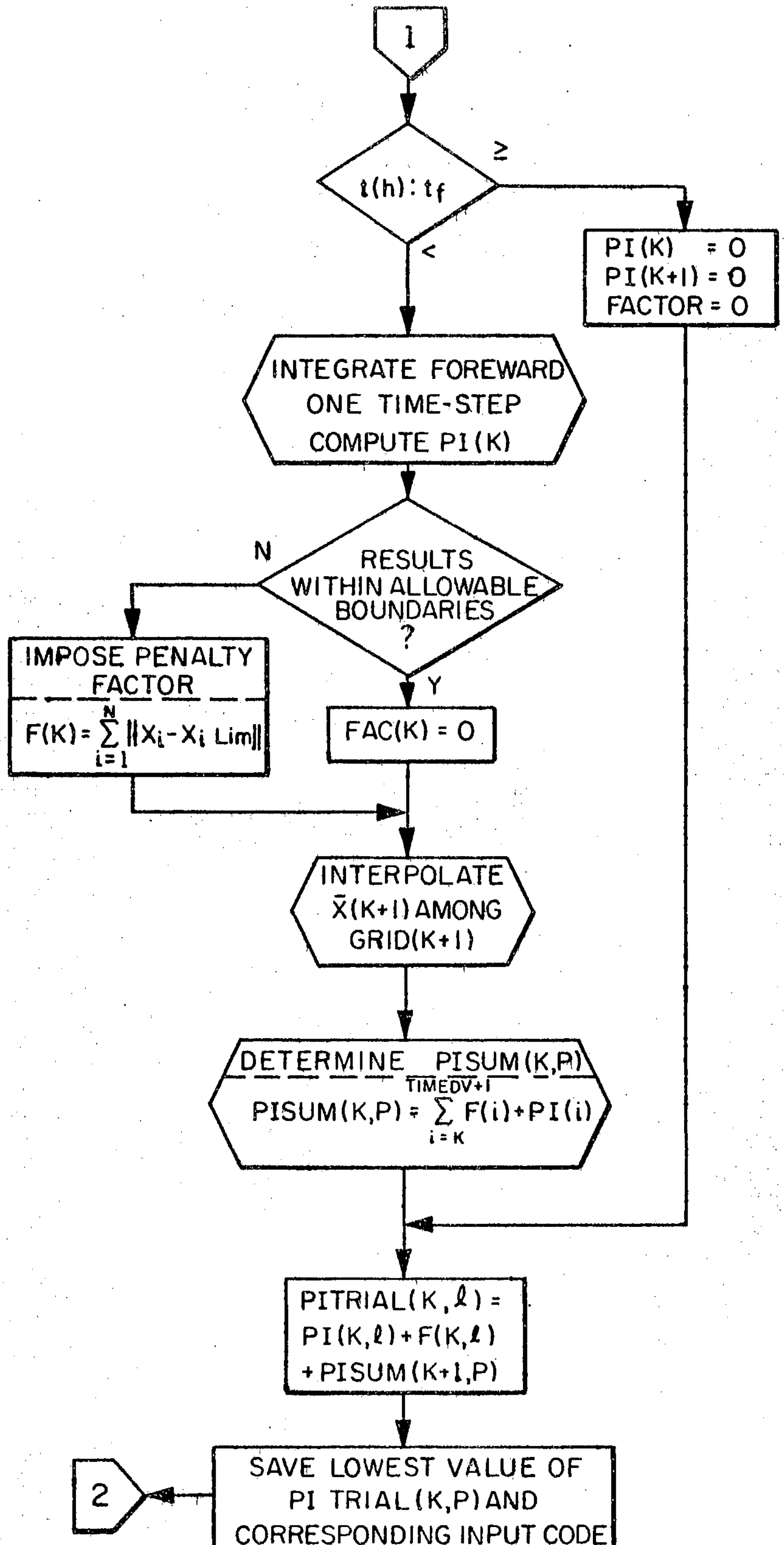
The following flow charts are for the algorithm. As these flow charts show, the algorithm initializes at final time and searches step-by-step backward in time until initial time. An important thing to notice is that a new state space grid to be searched is developed at each time step. Each grid is centered around the assumed trial solution. It is by this means that the algorithm limits its search area. It would be, and is, possible to cause the algorithm to revert back to classical Dynamic Programming by sufficiently increasing the size of each grid and the number of points contained in it.

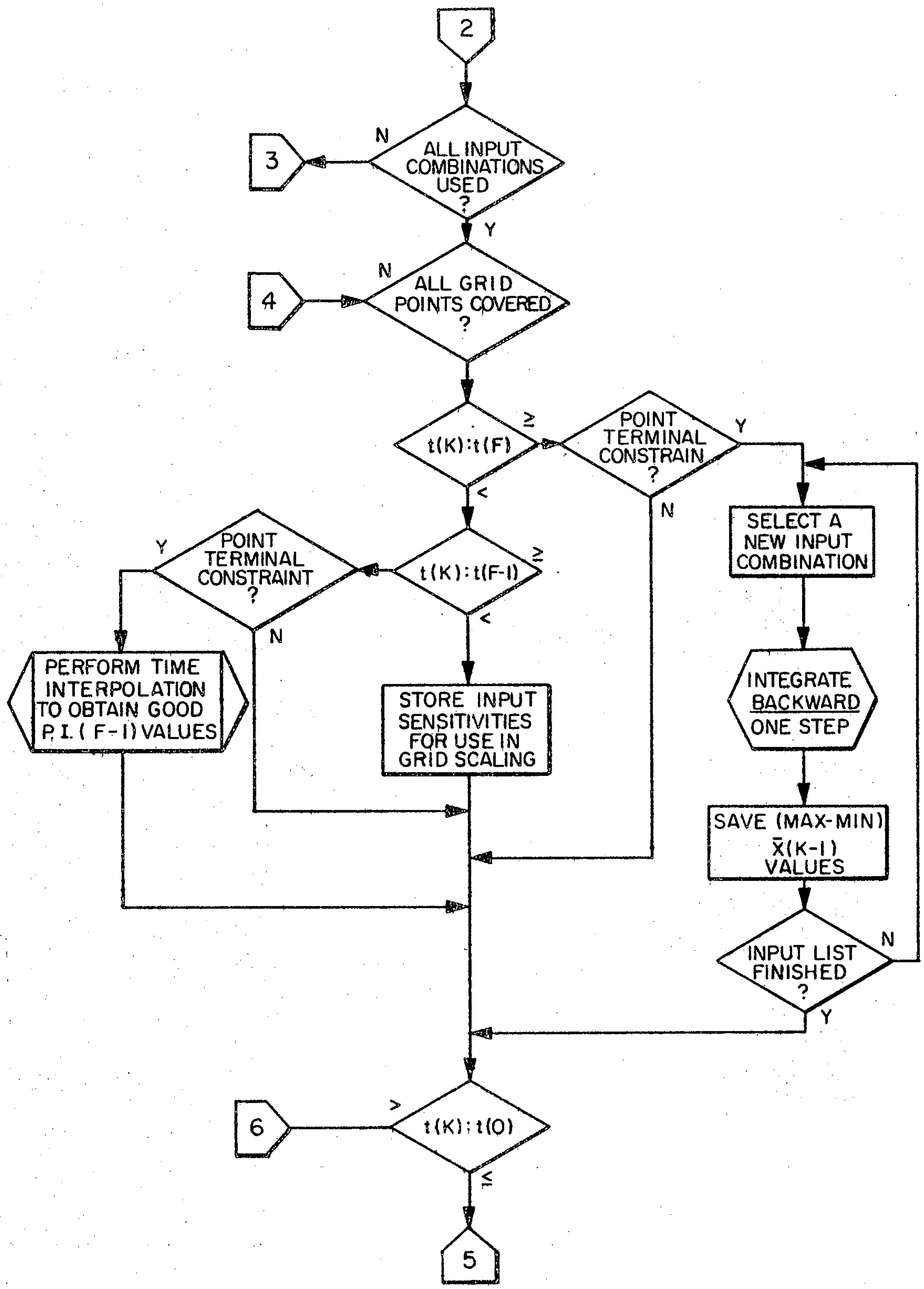
At the conclusion of the backward-going search, the algorithm uses the most optimal sequence of inputs to integrate forward in time to obtain a new trial optimal solution. This new solution is then used as the basis for the next iterative pass.

Two difficulties, which are natural consequences of the discretized nature of the algorithm arose and deserve mention. These difficulties are: inability to satisfy point final boundary conditions and inability to define control switching times more precisely than to the nearest time step. Great care must be used in the first of these two difficulties (fixed-point final boundary-value conditions) to avoid introducing artificial penalty factors which can actually create artificial local minima to which the solution will converge. The approach finally chosen in this algorithm to avoid the difficulty was to use backward integration for one time-step. The results of this backward integration are used to establish a pseudo-terminal manifold.

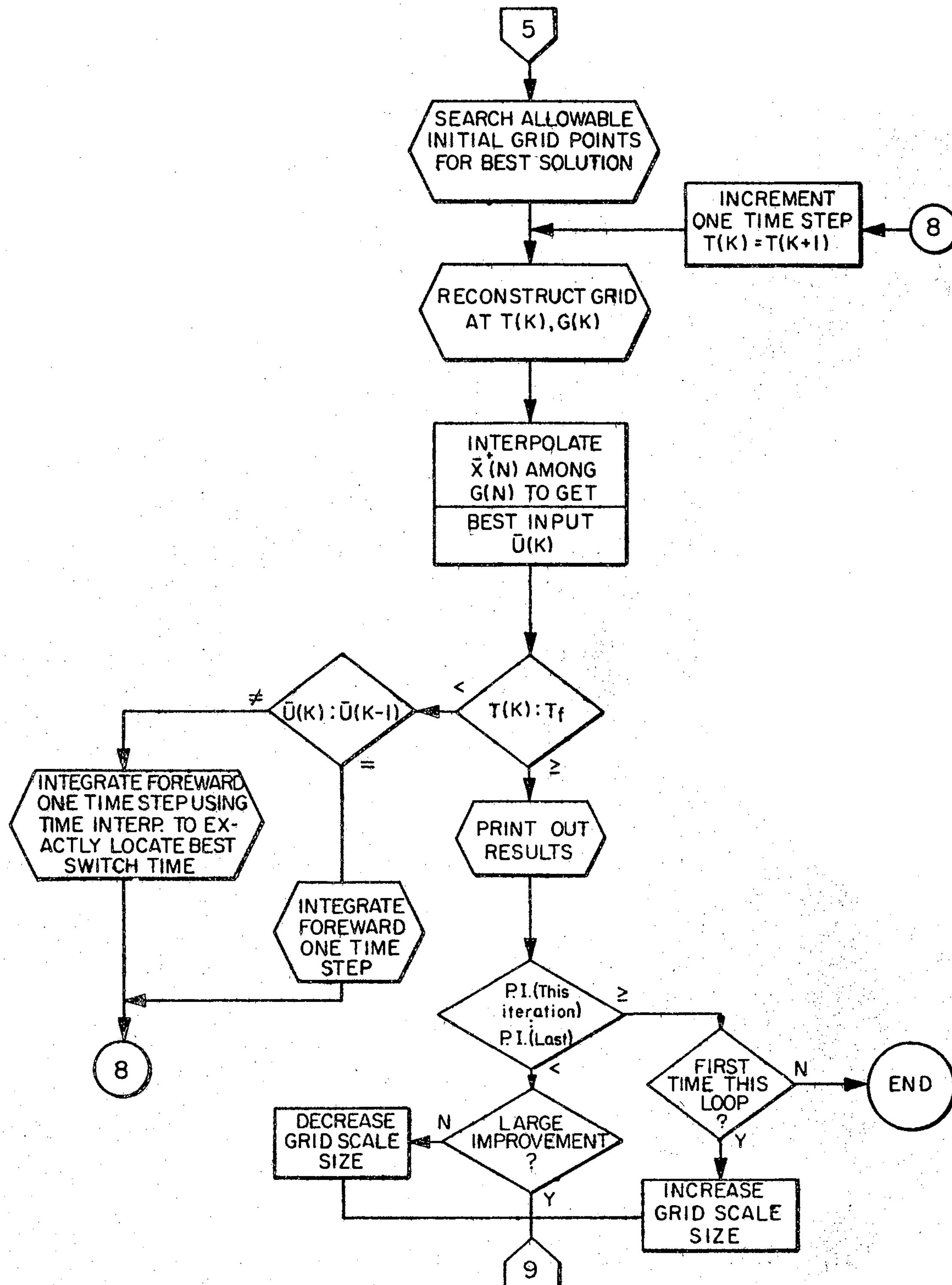
The approach chosen to help minimize the second difficulty (inaccurate switching times), rather than simply decreasing time-step was a time step interpolation scheme. This interpolation scheme will


START
INPUT DATA FOR INITIAL SETUP
N = ORDER OF SYSTEM
M = ORDER OF INPUT VECTOR
TIMEDV = NUMBER OF TIME-STEPS DESIRED
DEFINE STATE EQNS. & P.I.
$\bar{X}(K+1) = \bar{F}[\bar{X}(K), \bar{U}(K)]$
$P.I.(K) = I[\bar{X}(K), \bar{U}(K)]$
9
INITIALIZE AT FINAL TIME
$t(K) = t_f$
6
DECREMENT TIME
$t(K) = t(K-1)$
CONSTRUCT STATE VARIABLE GRID SET G(K)
$G(K) = \begin{bmatrix} X_{1i}(K) & i = 1, \cdots X_1 max \\ X_{2j}(K) & j = 1, \cdots X_2 max \\ \vdots & \\ X_{n\ell}(K) & K = 1, \cdots X_n max \end{bmatrix}$
4
SELECT A NEW GRID POINT BY INCREMENTING i, j, k
3
SELECT A NEW INPUT COMBINATION
$\bar{u}(k) = \begin{bmatrix} u_1 f & f = 1, \cdots u_1 max \\ u_2 g & g = 1, \cdots u_2 max \\ \vdots & \\ umh & h = 1, \cdots u_m max \end{bmatrix}$
1


1
t(h) : $t_f$
≥
<
PI(K) = 0
PI(K+1) = 0
FACTOR = 0
INTEGRATE FOREWARD ONE TIME-STEP COMPUTE PI(K)
RESULTS WITHIN ALLOWABLE BOUNDARIES ?
N
Y
IMPOSE PENALTY FACTOR
$F(K) = \sum_{i=1}^{N} \|X_i - X_i \text{ Lim}\|$
FAC(K) = 0
INTERPOLATE $\bar{X}$(K+1) AMONG GRID(K+1)
DETERMINE PISUM(K,P)
$PISUM(K,P) = \sum_{i=K}^{TIMEDV+1} F(i)+PI(i)$
PITRIAL(K,ℓ) = PI(K,ℓ) + F(K,ℓ) + PISUM(K+1,P)
SAVE LOWEST VALUE OF PI TRIAL(K,P) AND CORRESPONDING INPUT CODE
2



2
ALL INPUT COMBINATIONS USED ?
N
3
Y
ALL GRID POINTS COVERED ?
N
4
t(K):t(F)
≥
POINT TERMINAL CONSTRAIN ?
Y
N
<
t(K):t(F-1)
≥
POINT TERMINAL CONSTRAINT ?
Y
N
<
SELECT A NEW INPUT COMBINATION
PERFORM TIME INTERPOLATION TO OBTAIN GOOD P.I. (F-1) VALUES
STORE INPUT SENSITIVITIES FOR USE IN GRID SCALING
INTEGRATE BACKWARD ONE STEP
SAVE (MAX-MIN) X̄(K-1) VALUES
INPUT LIST FINISHED ?
N
Y
6
>
t(K):t(0)
≤
5


5
SEARCH ALLOWABLE INITIAL GRID POINTS FOR BEST SOLUTION
INCREMENT ONE TIME STEP T(K) = T(K+1)
8
RECONSTRUCT GRID AT T(K), G(K)
INTERPOLATE $\vec{X}$(N) AMONG G(N) TO GET
BEST INPUT $\bar{U}$(K)
T(K) : $T_f$
<
≥
$\bar{U}$(K) : $\bar{U}$(K-1)
≠
=
INTEGRATE FOREWARD ONE TIME STEP USING TIME INTERP. TO EXACTLY LOCATE BEST SWITCH TIME
INTEGRATE FOREWARD ONE TIME STEP
8
PRINT OUT RESULTS
P.I. (This iteration) : P.I. (Last)
≥
<
FIRST TIME THIS LOOP ?
N
Y
END
LARGE IMPROVEMENT ?
N
Y
DECREASE GRID SCALE SIZE
INCREASE GRID SCALE SIZE
9

determine the best switching time to within one-sixteenth of a time step. This switching time is then retained for use in the next iteration. Accomplishment of the time interpolation is provided by division of the time step in question into sixteen parts and comparing performance indexes realized for switching at each of the sixteen time increments.

The algorithm, as flow chartered, has been used and is usable for obtaining solutions to the previously mentioned very difficult class of problems. However, convergence problems still exist in the algorithm. Present efforts being made to improve convergence of the method are: improved methods for choosing grid scaling factors and time-step sizes and elimination of penalty-introduced artificial local minima. Computation times for the present implementation of the algorithm are comparable to that required for the quasi-linearization two-point boundary value solution of a continuous system of the same order.

VITA

James David Engelland

Candidate for the Degree of

Doctor of Philosophy

Thesis: OPTIMAL DISCRETE-LEVEL CONTROL

Major Field: Engineering

Biographical:

Personal Data: Born in Sterling, Kansas, April 3, 1940, the son of Alvin J. and Evelyn V. Engelland.

Education: Graduated from Sterling High School in 1958; received the Bachelor of Arts degree from Sterling College in May, 1962; the Bachelor of Science degree, with a major in Mechanical Engineering from the University of Kansas in January, 1963; and the Master of Science degree in May, 1964 with a major in Mechanical Engineering from the University of Kansas. Completed the requirements for the Degree of Doctor of Philosophy in July, 1970.

Professional Experience: Employed by American Telephone and Telegraph Long Lines Division during the summer of 1963. Worked as a staff member technical for Sandia Corporation, Albuquerque, New Mexico from July, 1964 to August, 1966. Employed by the School of Mechanical and Aerospace Engineering at Oklahoma State University as a graduate research assistant on the Army Tank Automotive Command Project from September, 1967 to June, 1969. Employed as an Aerosystems Project Engineer with General Dynamics, Fort Worth Division, from November, 1969 to May, 1970.

Organizations: Member of Tau Beta Pi, Sigma Tau, Pi Tau Sigma, American Society of Mechanical Engineers and Institute of Electrical and Electronic Engineers.