NASA CONTRACTOR REPORT

NASA CR-1009

GUIDANCE, FLIGHT MECHANICS AND TRAJECTORY OPTIMIZATION

Volume X - Dynamic Programming

by A. S. Abbott and J. E. McIntyre

Prepared by
NORTH AMERICAN AVIATION, INC.
Downey, Calif.
for George C. Marshall Space Flight Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1968

NASA CR-1009

GUIDANCE, FLIGHT MECHANICS AND TRAJECTORY OPTIMIZATION

Volume X - Dynamic Programming

By A. S. Abbott and J. E. McIntyre

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Issued by Originator as Report No. SID 66-1678-2

Prepared under Contract No. NAS 8-11495 by
NORTH AMERICAN AVIATION, INC.
Downey, Calif.

for George C. Marshall Space Flight Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

FOREWORD

This report was prepared under contract NAS 8-11495 and is one of a series intended to illustrate analytical methods used in the fields of Guidance, Flight Mechanics, and Trajectory Optimization. Derivations, mechanizations and recommended procedures are given. Below is a complete list of the reports in the series.

Volume I	Coordinate Systems and Time Measure
Volume II	Observation Theory and Sensors
Volume III	The Two Body Problem
Volume IV	The Calculus of Variations and Modern Applications
Volume V	State Determination and/or Estimation
Volume VI	The N-Body Problem and Special Perturbation Techniques
Volume VII	The Pontryagin Maximum Principle
Volume VIII	Boost Guidance Equations
Volume IX	General Perturbations Theory
Volume X	Dynamic Programming
Volume XI	Guidance Equations for Orbital Operations
Volume XII	Relative Motion, Guidance Equations for Terminal Rendezvous
Volume XIII	Numerical Optimization Methods
Volume XIV	Entry Guidance Equations
Volume XV	Application of Optimization Techniques
Volume XVI	Mission Constraints and Trajectory Interfaces
Volume XVII	Guidance System Performance Analysis

The work was conducted under the direction of C. D. Baker, J. W. Winch, and D. P. Chandler, Aero-Astro Dynamics Laboratory, George C. Marshall Space Flight Center. The North American program was conducted under the direction of H. A. McCarty and G. E. Townsend.

TABLE OF CONTENTS

SECTION PAGE

1.0 STATEMENT OF THE PROBLEM

This monograph will present both the theoretical and computational aspects of Dynamic Programming. The development of the subject matter in the text will be similar to the manner in which Dynamic Programming itself developed. The first step in the presentation will be an explanation of the basic concepts of Dynamic Programming and how they apply to simple multi-stage decision processes. This effort will concentrate on the meaning of the Principle of Optimality, optimal value functions, multistage decision processes and other basic concepts.

After the basic concepts are firmly in mind, the applications of these techniques to simple problems will be useful in acquiring the insight that is necessary in order that the concepts may be applied to more complex problems. The formulation of problems in such a manner that the techniques of Dynamic Programming can be applied is not always simple and requires exposure to many different types of applications if this task is to be mastered. Further, the straightforward Dynamic Programming formulation is not sufficient to provide answers in some cases. Thus, many problems require additional techniques in order to reduce computer core storage requirements or to guarantee a stable solution. The user is constantly faced with trade-offs in accuracy, core storage requirements, and computation time. All of these factors require insight that can only be gained from the examination of simple problems that specifically illustrate each of these problems.

Since Dynamic Programming is an optimization technique, it is expected that it is related to Calculus of Variations and Pontryagin's Maximum Principle. Such is the case. Indeed, it is possible to derive the Euler-Lagrange equation of Calculus of Variations as well as the boundary condition equations from the basic formulation of the concepts of Dynamic Programming. The solutions to both the problem of Lagrange and the problem of Mayer can also be derived from the Dynamic Programming formulation. In practice, however, the theoretical application of the concepts of Dynamic Programming present a different approach to some problems that are not easily formulated by conventional techniques, and thus provides a powerful theoretical tool as well as a computational tool for optimization problems.

The fields of stochastic and adaptive optimization theory have recently shown a new and challenging area of application for Dynamic Programming. The recent application of the classical methods to this type of problem has motivated research to apply the concepts of Dynamic Programming with the hope that insights and interpretations afforded by these concepts will ultimately prove useful.

2.0 STATE OF THE ART

2.1 Development of Dynamic Programming

The mathematical formalism known as "Dynamic Programming" was developed by Richard Bellman during the early 1950's with one of the first accounts of the method given in the 1952 Proceedings of the Academy of Science (Reference 2.1.1). The name itself appears to have been derived from the related discipline of Linear Programming, with the over-riding factor in the selection of this name stemming more probably from the abundance of research funding available for linear programming type problems, than from the limited technical similarity between the two.

Dynamic Programming did not take long to become widely applied in many different types of problems. In less than 15 years after its origination it has found its way into many different branches of science and is now widely used in the chemical, electrical and aerospace industries. However, even the most rapid perusal of any of Bellman's three books on the subject (Reference 2.1.2, 2.1.3, and 2.1.4) makes one point very clear: the field in which Dynamic Programming finds its most extensive application is not that of science, but of economics, with the problems here all rather loosely groupable under the heading of getting the greatest amount of return from the least amount of investment. Of the several factors contributing to this rapid growth and development, no small emphasis should be placed on the vigorous application program conducted by Bellman and his collegues at Rand in which a multitude of problems were analyzed using the method, and the results published in many different Journals, both technical and non-technical. A brief biographical sketch accompanying an article of Bellman's in a recent issue of the Saturday Review, (Ref. 2.1.5) states that his publications include 17 books and over 400 technical papers, a not-insignificant portion of which deal with the subject of Dynamic Programming.

Historically, Dynamic Programming was developed to provide a means of optimizing multi-stage decision processes. However, after this use was finally established, the originators of Dynamic Programming began to use their mathematical licenses by considering practically all problems as multistage decision processes. There were sound reasons behind such attempts. First, the solution of many practical problems by the use of the classical method of Calculus of Variations was extremely complicated and sometimes impossible. Second, with the fields of high speed computers and mass data processing systems on the threshold, the idea of treating continuous systems in a multi-stage manner was very feasible and promising. This new breakthrough for Dynamic Programming gave rise to a study of the relationships between the Calculus of Variation and Dynamic Programming and applications to trajectory processes and feedback control.

The extension of Dynamic Programming to these other fields, however, presented computational problems. For example, it became necessary to study topics such as accuracy, stability and storage in order to handle these more complicated problems. One of the beauties of Dynamic Programming came to rescue in solving some of these problems. It is idiosyncrasy exploitation. Whereas problem peculiarities usually are a burden to classical techniques, they are usually blessings to the dynamic programmer. It is possible to save computation time, to save storage and/or to increase accuracy by exploiting problem peculiarities in Dynamic Programming.

An understanding of Dynamic Programming hinges on an understanding of the concept of a multi-stage decision process, a concept which is most easily described by means of an example. Consider a skier at the top of a hill who wishes to get down to the bottom of the hill as quickly as possible. Assume that there are several trails available which lead to the bottom and that these trails intersect and criss-cross one another as the slope is descended. The down hill path which is taken will depend only on a sequence of decisions which the skier makes. The first decision consists of selecting the trail on which to start the run. Each subsequent decision is made whenever the current trail intersects some new trail, at which point the skier must decide whether to take the new trail or not. Thus, associated with each set of decisions is a path leading to the bottom of the hill, and associated with each path is a time, namely the time it takes to negotiate the hill. The problem confronting the skier is that of selecting that sequence of decisions (i.e., the particular combination of trails) which result in a minimum run time.

From this example, it is clear that a multi-stage decision process possesses three important features:

(1) To accomplish the objective of the process (in the example above, to reach the bottom of the hill) a sequence of decisions must be made.

(2) The decisions are coupled in the sense that the n_th_ decision is affected by all the prior decisions, and it, in turn, effects all the subsequent decisions. In the skier example, the very existence of an n_th_ decision depends on the preceding decisions.

(3) Associated with each set of decisions there is a number which depends on all the decisions in the set (e.g., the time to reach the bottom of the hill). This number which goes by a variety of names will be referred to here as the performance index. The problem is to select that set of decisions which minimizes the performance index.

There are several ways to accomplish the specified objective and at the same time minimize the performance index. The most direct approach would involve evaluating the performance index for every possible set of decisions. However, in most decision processes the number of different decision sets is so large that such an evaluation is computationally impossible. A second approach would be to endow the problem with a certain mathematical structure (e.g., continuity, differentiability, analyticity, etc.), and then use a standard mathematical technique to determine certain additional properties which the optimal decision sequence must have. Two such mathematical techniques are the maxima-minima theory of the Differential Calculus and the Calculus of Variations. A third alternative is to use Dynamic Programming.

Dynamic Programming is essentially a systematic search procedure for finding the optimal decision sequence; in using the technique it is only necessary to evaluate the performance index associated with a small number of all possible decision sets. This approach differs from the well-known variational methods, in that it is computational in nature and goes directly to the determination of the optimal decision sequence without attempting to uncover any special properties which this decision sequence might have. In this sense the restrictions on the problem's mathematical structure, which are needed in the variational approach, are totally unnecessary in Dynamic Programming. Furthermore, the inclusion of constraints in the problem, a situation which invariably complicates a solution of the variational methods, facilitates solution generation in the Dynamic Programming approach since the constraints reduce the number of decision sets over which the search must be conducted.

The physical basis for Dynamic Programming lies in the "Principle of Optimality," a principle so simple and so self-evident that one would hardly expect it could be of any importance. However, it is the recognition of the utility of this principle along with its application to a broad spectrum of problems which constitutes Bellman's major contribution.

Besides its value as a computational tool, Dynamic Programming is also of considerable theoretical importance. If the problem possesses a certain mathematical structure, for example, if it is describable by a system of differential equations, then the additional properties of the optimal decision sequence, as developed by the Maximum Principle or the Calculus of Variations, can also be developed using Dynamic Programming. This feature gives a degree of completeness to the area of multi-stage decision processes and allows the examination of problems from several points of view. Furthermore, there is a class of problems, namely stochastic decision processes, which appear to lie in the variational domain, and yet which escape analysis by means of the Variational Calculus or the Maximum Principle. As will be shown, it is a rather straightforward matter to develop the additional properties of the optimal stochastic decision sequence by using Dynamic Programming.

The purpose of this monograph is to present the methods of Dynamic Programming and to illustrate its dual role as both a computational and theoretical tool. In keeping with the objectives of the monograph series, the problems considered for solution will be primarily of the trajectory and control type arising in aerospace applications. It should be mentioned that this particular class of problems is not as well suited for solution by means of Dynamic Programming as those in other areas. The systematic search procedure inherent in Dynamic Programming usually involves a very large number of calculations often in excess of the capability of present computers. While this number can be brought within reasonable bounds, it is usually done at the expense of compromising solution accuracy. However, this situation should change as both new methods and new computers are developed.

The frequently excessive number of computations arising in trajectory and control problems has somewhat dampened the initial enthusiasm with which Dynamic Programming was received. Many investigators feel that the extensive applications of Dynamic Programming have been over-stated and that computational procedures based upon the variational techniques are more suitable for solution generation. However, it should be mentioned that the originators of these other procedures can not be accused of modesty when it comes to comparing the relative merits of their own technique with some other. The difficulty arises in that each may be correct for certain classes of problems and unfortunately, there is little which can be used to determine which will be best for a specific problem since the subject is relatively new and requires much investigation.

Without delineating further the merits of Dynamic Programming in the introduction it is noted that current efforts are directed to its application to more and more optimization problems. Since an optimization problem can almost always be modified to a multi-stage decision processes, the extent of application of Dynamic Programming has encompassed business, military, managerial and technical problems. A partial list of applications appears in Ref. 2.1.2. Some of the more pertinent fields are listed below.

Allocation processes
Calculus of Variations
Cargo loading
Cascade processes
Communication and Information Theory
Control Processes
Equipment Replacement
Inventory and Stock Level
Optimal Trajectory Problems
Probability Theory
Reliability
Search Processes
Smoothing
Stochastic Allocation
Transportation
Game Theory
Investment

2.2 Fundamental Concepts and Applications

Section 2.1 presented the example of a skier who wishes to minimize the time required to get to the bottom of the hill. It was mentioned that the Dynamic Programming solution to this problem resulted in a sequence of decisions, and that this sequence was determined by employing the Principle of Optimality. In this section, the Principle of Optimality and other basic concepts will be examined in detail, and the application of these concepts will be demonstrated on some elementary problems.

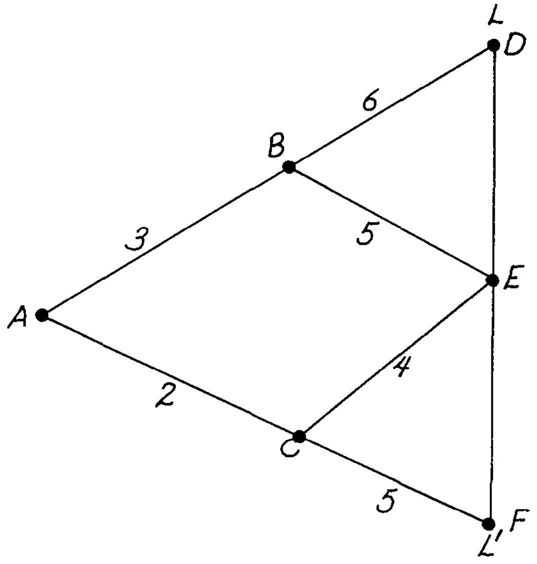
The Principle of Optimality is stated formally in Ref. 0.4 as follows:

> An optimal policy has the property that whatever the initial state and the initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

It is worthy to note that the Principle of Optimality can be stated mathematically as well as verbally. The mathematical treatment has been placed in section 2.4 in order that the more intuitive aspects can be stressed without complicating the presentation. The reader interested in the mathematical statement of the Principle is referred to Sections 2.4.1 and 2.4.2

Before this principle can be applied, however, some measure of the performance which is to be optimized must be established. This requirement introduces the concept of the optimal value function. The optimal value function is most easily understood as the relationship between the parameter which will be optimized and the state of the process. In the case of the skier who wishes to minimize the time required to get to the bottom of the hill, the optimal value function is the minimum run time associated with each intermediate point on the hill. Here the state of the process can be thought of as the location of the skier on the hill. The optimal value function is referred to by many other names, depending upon the physical nature of the problem. Some of the other names are "cost function." "performance index," "profit," or "return function." However, whatever the name, it always refers to that variable of the problem that is to be optimized.

Now that the concept of an optimal value function has been presented, the Principle of Optimality can be discussed more easily. In general, the n stage multi-decision process is the problem to which Dynamic Programming is applied. However, it is usually a very difficult problem to determine the optimal decision sequence for the entire n stage process in one set of computations. A much simplier problem is to find the optimum decision of a one stage process and to employ Dynamic Programming to treat the n stage process as a series of one stage processes. This solution requires the investigation of the many one stage decisions that can be made from each state of the process. Although this procedure at first may appear as the "brute force" method (examining all the combinations of the possible decisions), it is the Principle of Optimality that saves this technique from the unwieldy number of computations involved in the "brute force" method. This reasoning is most easily seen by examining a two stage process. Consider the problem of finding the optimal path from a point A to the LL' in the following sketch.



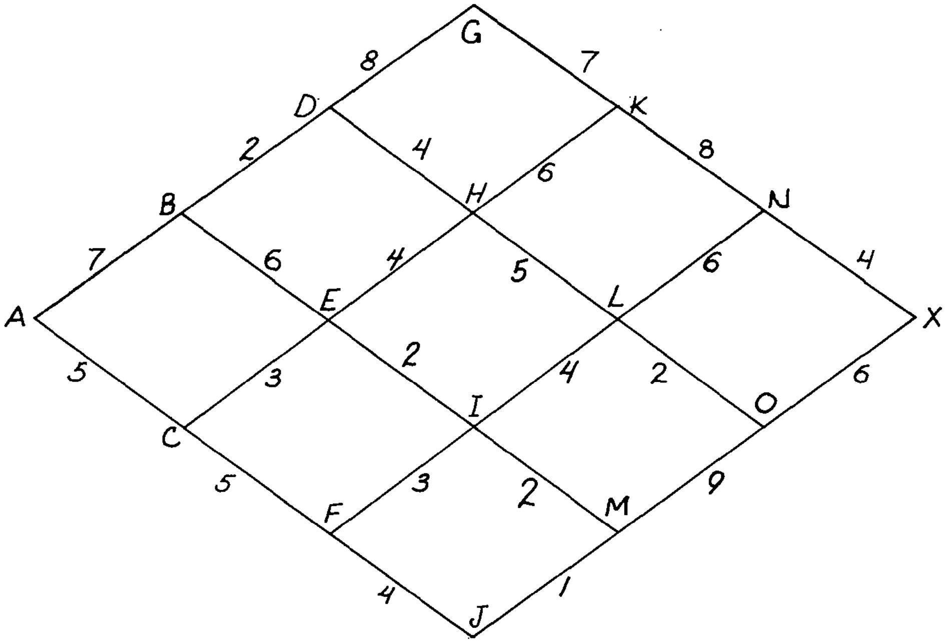
The numbers on each line represent the "cost" of that particular transition. This two stage process will now be treated as two one-stage processes. The Principle of Optimality will then be used to determine the optimal decision sequence. Starting at point A, the first decision to be made is whether to connect point A to point B or point C. The Principle of Optimality states, however, that whichever decision is made the remaining choices must be optimal. Hence, if the first decision is to connect A to B, then the remaining decision must be to connect B to E since it is the optimal path from B to line LL'. Similarly, if the first decision is to connect A to C, then the remaining decision must be to connect C to E. These decisions enable an optimal cost to be associated with each of the points B and C; that is, the optimal cost from each of these points to the line LL'. Hence, the optimal value of B is 5 and of C is 4 since these are the minimum costs from each of the points to line LL'.

The first decision can be found by employing the Principle of Optimality once again. Now, however, the first decision is part of the remaining sequence, which must be optimal. The optimal value function must be calculated for each of the possibilities for the first decision. If the first decision is to go to B, the optimal value function at point A is the cost of that decision plus the optimal cost of the remaining decision, or, $3 + 5 = 8$. Similarly, the optimal value function at point A for a choice of C for the first decision is $2 + 4 = 6$. Hence, the optimal first decision is to go to C and the optimal second decision is to go to E. The optimal path is thus, A-C-E.

Although the previous problem was very simple in nature, it contains all the fundamental concepts involved in applying Dynamic Programming to a multi-stage decision process. The remainder of this section uses the same basic concepts and applies them to problems with a larger number of stages and dimensions.

2.2.1 Multi-Stage Decision Problem

The basic ideas behind Dynamic Programming will now be applied to a simple travel problem. It is desired to travel from a certain city, A, to a second city, X, well removed from A.



Since there are various types of travel services available to the minimum cost from one intermediate city to another will vary depending upon the nature of the transportation. In general, this cost will not be strictly linear with distance. The intermediate cities appear in the sketch above as the letters B, C, D, etc. with the cost in traveling between any two cities entered on the connecting diagonal. The problem is to determine that route for which the total transportation costs are a minimum. A similar problem is treated in Ref. 2.4.1.

Obviously, one solution to this problem is to try all possible paths from A to X, calculate the associated cost, and select the least expensive. Actually, for "small"problems this approach is not unrealistic. If, on the other hand, the problem is multi-dimensional, such a "brute force" method is not feasible.

First, consider the two ways of leaving city A. It is seen that the minimum cost of going to city B is 7, and city C is 5. Based upon this information a cost can be associated with each of the cities. Since there are no less expensive ways of going from city A to these cities, the cost associated with each city is optimum. A table of costs can be constructed for cities B and C as follows:

City	Optimum Cost	Path for Optimum Cost
B	7	A–B
C	5	A–C

Now, the cost of cities D, E, and F will be found. The cost of D is 7 + 2=9. Since there are no other ways of getting to D, 9 is the optimum value. The cost of city E, on the other hand, is 13 by way of B and only 8 by way of C. So the optimum value of city E is 8. The cost for city F is 10 by way of city C. A table can now be constructed for cities D, E, and F as follows:

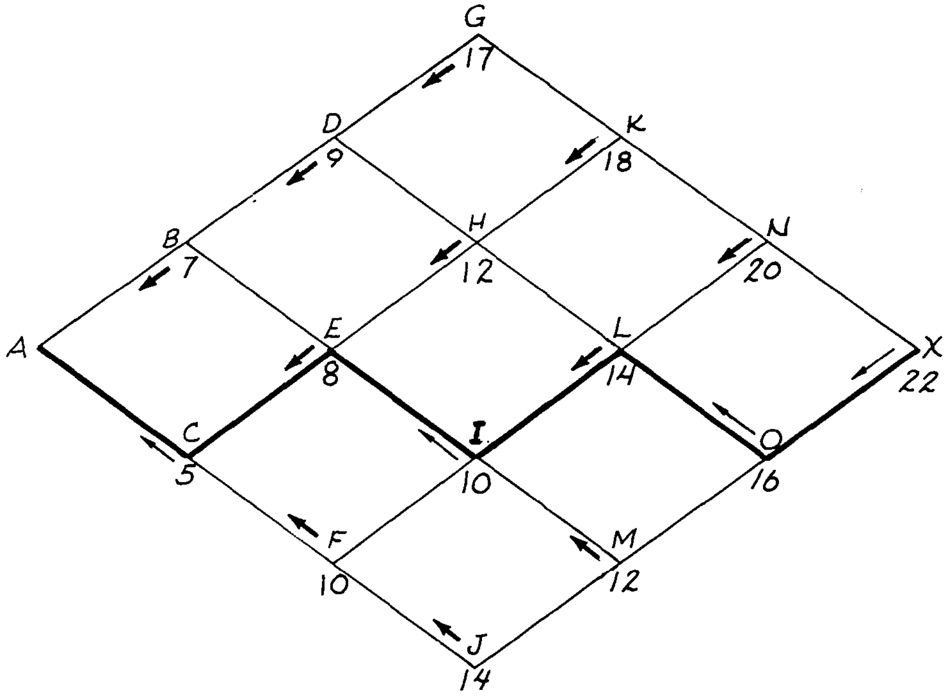
City	Optimum Cost	Via
D	9	B
E	8	C
F	10	C

At this point, it is worthy to note two of the basic concepts that were used. Although they are very subtle in this case, they are keys to understanding Dynamic Programming.

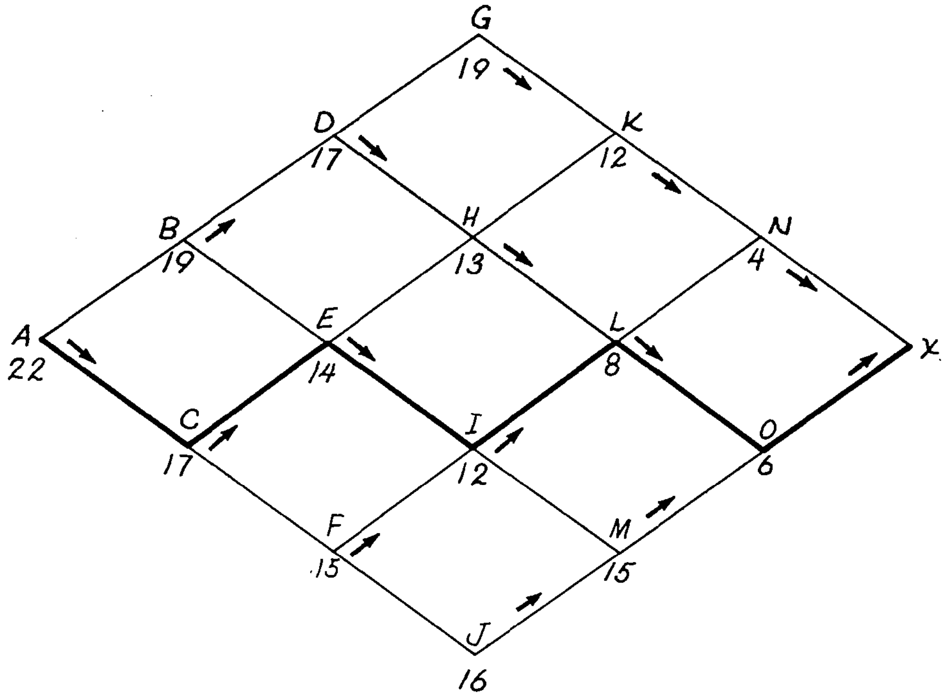
First, the decision to find the cost of city E to be a minimum by choosing to go via city C is employing the Principle of Optimality. In this case, the optimal value function, or cost, was optimized by making the current decision such that all the previous decisions (including the recent one) yield an optimum value at the present state. In other words, there was a choice of going to city E via city B or C and city C was chosen because it optimized the optimal value function, which sums the cost of all previous decisions. One more stage will now be discussed so that the principles are firmly in mind. Consider the optimum costs of cities G, H, I, and J. There is no choice on the cost of city G. It is merely the optimum cost of city D (=9) plus the cost of going to city G from city D (=8), or 17. City H can be reached via city D or city E. In order to determine the optimum value for city H, the optimum cost of city D plus the cost of travel from D to H is compared to the optimum cost of E plus the cost of travel from E to H. In this case the cost via city E is 8 + 4 = 12 whereas the cost via D is 9 + 14 = 23. Hence, the optimal value of city H is 12 and the optimum path is via city E. By completely analogous computations the optimal cost and optimum path for the remaining cities can be found and are shown below:

City	Optimum Cost	Via
G	17	D
H	12	E
I	10	E
J	14	F
K	18	H
L	14	I
M	12	I
N	20	L
O	16	L
X	22	O

The previous computations are sufficient for determining optimum path. From the tables that have been constructed the optimum decision can be found. The following sketch shows the optimum decision for each point by an arrow.



The optimum path, shown by a heavy line, can be found by starting at city X and following the arrows to the left. It should be noted that the preceding computations were made from left to right. This construction then resulted in an optimum path which was determined from right to left. Identical results could have been obtained if the computations are performed from right to left. The following sketch shows the optimum decisions for this method of attack.



The optimum path can be found by starting at city A and following the arrows from left to right. This path is shown by a heavy line in the sketch.

There is an advantage to each of these computational procedures depending upon the nature of the problem. In some problems, the terminal constraints are of such a nature that it is computationally advantageous to start computing at the end of the problem and progress to the beginning. In other problems, the reverse may be true. The preceding sample problem was equally suitable to either method. Depending upon the formulation of the problem, the costs for typical transitions may not be unique (the cost could depend upon the path as in trajectory problems) as they were in the sample problem. This may be a factor that will influence the choice of the method to be used. To summarize, the optimal value function and the Principle of Optimality have been used to determine the best decision policy for the multi-stage decision process: the optimal value function kept track of at least expensive possible cost for each city while the Principle of Optimality used this optimum cost as a means by which it could make a decision for the next stage of the process. Then, a new value for the optimal value function was computed for the next stage. After the computation was complete, each stage had a corresponding decision that was made and which was used to determine the optimum path.

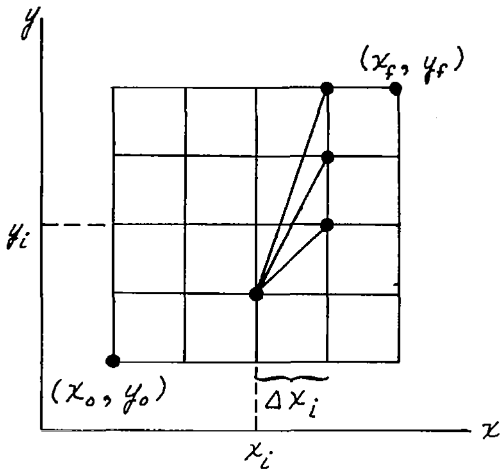
2.2.2 Applications to the Calculus of Variations

So far, the use of Dynamic Programming has been applied to multi-stage decision processes. The same concepts can, however, be applied to the solution of continuous variational problems providing the problem is formulated properly. As might be expected, the formulation involves a discretizing process. The Dynamic Programming solution will be a discretized version of the continuous solution. Providing there are no irregularities, the discretized solution converges to the continuous solution in the limit as the increment is reduced in size. It is interesting to note that the formal mathematical statement of the concepts already introduced can be shown to be equivalent to the Euler-Lagrange equation in the Calculus of Variations in the limit (see Section 2.4). The two classes of problems that are considered in this section are the problem of Lagrange and the and the problem of Mayer. The general computational procedure for the application of Dynamic Programming to each of these problem classes will be discussed in the following paragraphs. Some illustrative examples are included in Sections 2.2.2.1, 2.2.2.2, and 2.2.2.3 so that the specific applications can be seen.

The problem of Lagrange can be stated as finding that function y(x) such that the functional

$$J = \int_{x_0, y_0}^{x_f, y_f} f(x, y, y')\, dx \tag{2.2.1}$$

is a minimum. That is, of all the functions passing through the points (x_o, y_o) and (x_f, y_f), find that particular one that minimizes J. The classical treatment of this problem is discussed in Reference (2.1). The approach taken here is to discretize this space in the region of interest. The following sketch indicates how the space could be divided.



The integral in Equation 2.2.1 can now be written in its discrete form as

$$J = \sum_{i=1}^{N} f_i \left(x_i, y_i, y_i'\right) \Delta x_i \tag{2.2.2}$$

The evaluation of the ith term can be seen for a typical transition in the above sketch. The choice of y'_i can be thought of as being the decision parameter. The similarities to the previous examples should now be evident. Each transition in the space has an associated "cost" just as in the previous travel problem. The problem is to find the optimum path from (x_o, y_o) to (x_f, y_f) such that J, or the total cost, is minimized. Obviously, if a fairly accurate solution is desired, it is not advantageous to choose big increments when dividing the space. It must be kept in mind, however, that the amount of computation involved increase quite rapidly as the number of increments increases. A trade-off must be determined by the user in order to reach a balance between accuracy and computation time.

The problem of Mayer can be shown to be equivalent to the problem of Lagrange (see Ref. 2.1). This problem will be included in this discussion because it is the form in which guidance, control and trajectory optimization problems usually appear. The general form of the equations for a problem of the Mayer type can be written as

$$\dot{x} = f(x, u) \tag{2.2.3}$$

where x is an n dimentional state vector and u is a r dimensional control vector. It is desired to minimize a function of the terminal state and terminal time, i.e.,

$$\phi(x_f, t_f) \qquad = \text{minimum}$$

subject to the terminal constraints

$$\psi_j = (x_f, t_f) \qquad j = 1, m$$

(A more detailed statement of the problem of Mayer can be found in Section 2.4.8 or Reference 2.1).

The approach that is used to solve this problem with Dynamic Programming is quite similar to the Lagrange formulation. The state space component is divided into many increments. The "cost" of all the allowable transitions is then computed. Each diffierent path eminating of the same point in the state space corresponds to a different control, which can be thought of as being analogous to the decision at that point. With these preliminary remarks in mind, some illustrative examples will now be presented.

2.2.2.1 Shortest Distance Between Two Points

The previous travel problem was intentionally made simple so that the concepts of Dynamic Programming could be conveyed easily. Most practical problems involve many more decisions and many more choices for each decision. To give the reader an idea of how much is involved in a slightly more complicated problem, an example of a 3 dimensional problem will be given.

The problem to be considered is the Lagrange Problem of Calculus of Variations, i.e., minimize the following functional

$$J = \min \int_{a,b}^{c,d} F(x, y, y')dx \tag{2.2.4}$$

This is the basic problem of the Calculus of Variations with fixed end points. The classical methods of the solution are well known and are shown in Ref. 2.2.1. The approach of Dynamic Programming is to break the interval into many segments. Each segment corresponds to one stage of a multistage decision process. The object is to find the optimum choice of y' for each segment such that

$$\sum_{i=1}^{n} F_i \Delta x_i$$

is minimized. An example of this kind of problem is that of finding the shortest path between two points. Although the solution to this problem is obvious, it is informative to try to solve the problem with the techniques of Dynamic Programming. It should be noted that the answer from the Dynamic Programming approach will not be exact because of the discretizing that must be performed in order to formulate the problem as a multistage decision process. The answer will approach the correct answer in the limit as the number of grids is increased. The specific problem to be considered is the shortest path from the origin of a rectangular 3 space coordinate system to the point (5,6,7). The discretizing is performed by constructing cubic layers around the origin, with each layer representing a decision stage. The cost of going from a point on one layer to a point on the next layer is the length of a line connecting the two points, i.e.,

$$J = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{2.2.5}$$

where (x_1, y_1, z_1) is the point on one layer and (x_2, y_2, z_2) is the point on the other layer.

In order to keep the problem manageable only two such layers will be used. The first layer will be a cube with one vertex at the origin and the other vertices at (0,0,1) (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), and (1,1,1). The permissible transitions from the origin to the first layer are shown below with the corresponding costs.

From	To	Cost
(0, 0, 0)	(0, 0, 1)	1.000
	(0, 1, 0)	1.000
	(0, 1, 1)	1.414
	(1, 0, 0)	1.000
	(1, 0, 1)	1.414
	(1, 1, 0)	1.414
(0, 0, 0)	(1, 1, 1)	1.732

The second layer chosen is the cube with one vertex at the origin and the others (0, 0, 4), (0, 4, 0), (0, 4, 4), (4, 0, 0), (4, 0, 4), (4, 4, 0), and (4, 4, 4). In addition to the transitions from the vertices of the first layer to the vertices of the second layer, transitions will also be allowed to points between the vertices of the second layer, e.g. (4, 0, 2). This allows more possible choices for the transitions and thus permits the Dynamic Programming solution to be closer to the actual solution.

As mentioned earlier, one of the beauties of Dynamic Programming is that the problem peculiarities can be used to simplify the problem. This advantage will be utilized here by eliminating some of the possible transitions from the first layer to the second layer. The philosophy behind this elimination is that a certain amount of continuity is assumed in the solution. It is not expected that the solution will consist of arcs which go in one direction for the first transition and then in the opposite direction for the second transition. For this reason, only the transitions from layer "one" to layer "two" that has been permitted are those that correspond to light rays that would propagate from the first point of the transition.

With these considerations in mind, the permissible transitions from the first layer to the second will be found. The various points of the second layer that are allowable transition points are listed below:

(0, 0, 0)
(0, 0, 2)
(0, 0, 4)
(0, 2, 0)
(0, 2, 2)
(0, 2, 4)
(0, 4, 0)
(0, 4, 2)
(0, 4, 4)

(2, 0, 0)
(2, 0, 2)
(2, 0, 4)
(2, 2, 0)
(2, 2, 2)
(2, 2, 4)
(2, 4, 0)
(2, 4, 2)
(2, 4, 4)

(4, 0, 0)
(4, 0, 2)
(4, 0, 4)
(4, 2, 0)
(4, 2, 2)
(4, 2, 4)
(4, 4, 0)
(4, 4, 2)
(4, 4, 4)

The cost of the allowable transitions from the first layer to the second layer are shown below:

TO \ FROM	011	100	001	010	101	110	111
004	3.162		3.000		3.162		3.316
024	3.162		3.605		3.741		3.316
040	3.162			3.000		3.162	3.316
042	3.162			3.605		3.714	3.316
044	4.242		5.000	5.000	5.099	5.099	4.358
204	3.741		3.605		3.162		3.316
224	3.741		4.123		3.741		3.316
240	3.741			3.605		3.162	3.316
242	3.741			4.123		3.714	3.316
244	4.690		5.385		5.099	5.099	4.358
400		3.000			3.162	3.162	3.316
402		3.605			3.162	3.741	3.316
404	5.009	5.000	5.000		4.242	5.009	4.358
420		3.605			3.741	3.162	3.316
422		4.123			3.741	3.741	3.316
424	5.099	5.385	5.385		4.690	5.099	4.358
440	5.099	5.000		5.000	5.099	4.242	4.358
442	5.099	5.385		5.385	5.099	4.609	4.358
444	5.830	6.403	6.403	6.403	5.830	5.830	5.196

The blank areas represent transitions that are not allowed because of reasons previously stated. The transitions from the second layer to the terminal point are shown in the following table:

FROM	TO	COST
(0, 0, 4)	(5, 6, 7)	8.366
(0, 2, 4)		7.071
(0, 4, 0)		8.831
(0, 4, 2)		7.348
(0, 4, 4)		6.164
(2, 0, 4)		7.348
(2, 2, 4)		5.830
(2, 4, 0)		9.591
(2, 4, 2)		6.164
(2, 4, 4)		4.690
(4, 0, 0)		9.273
(4, 0, 2)		7.874
(4, 0, 4)		6.782
(4, 2, 0)		8.124
(4, 2, 2)		6.480
(4, 2, 4)		5.099
(4, 4, 0)		7.348
(4, 4, 2)		5.477
(4, 4, 4)	(5, 6, 7)	3.741

Now that the cost of each transition has been established, the methods of Dynamic Programming can be used to find the optimum path from the origin to point (5, 6, 7). The first step is the definition of the optimum cost for each point. Working backwards from point (5, 6, 7) the optimum cost of the points on the second layer are shown in the previous table. The optimum cost of the points on the first layer can be found by finding the path that gives the minimum value of the total cost of going from (5, 6, 7) to layer 2 and from layer 2 to layer 1. As an example, consider the optimum cost of point (0, 0, 1). Table 2.2.1 shows the various paths from (5, 6, 7) to (0, 0, 1) through layer 2.

POINT (0, 0, 1)	PATH	COST	MINIMUM COST
	(0, 0, 4)	3.000 + 8.366 = 11.366	
	(0, 2, 4)	3.605 + 7.071 = 10.676	
	(0, 4, 4)	5.000 + 6.164 = 11.164	
	(2, 0, 4)	3.605 + 7.348 = 10.943	
	(2, 2, 4)	4.123 + 5.830 = 9.953	9.953
	(2, 4, 4)	5.385 + 4.690 = 10.075	
	(4, 0, 4)	5.000 + 6.782 = 11.782	
	(4, 2, 4)	5.385 + 5.099 = 10.484	
	(4, 4, 4)	6.403 + 3.741 = 10.144	

Table 2.2.1

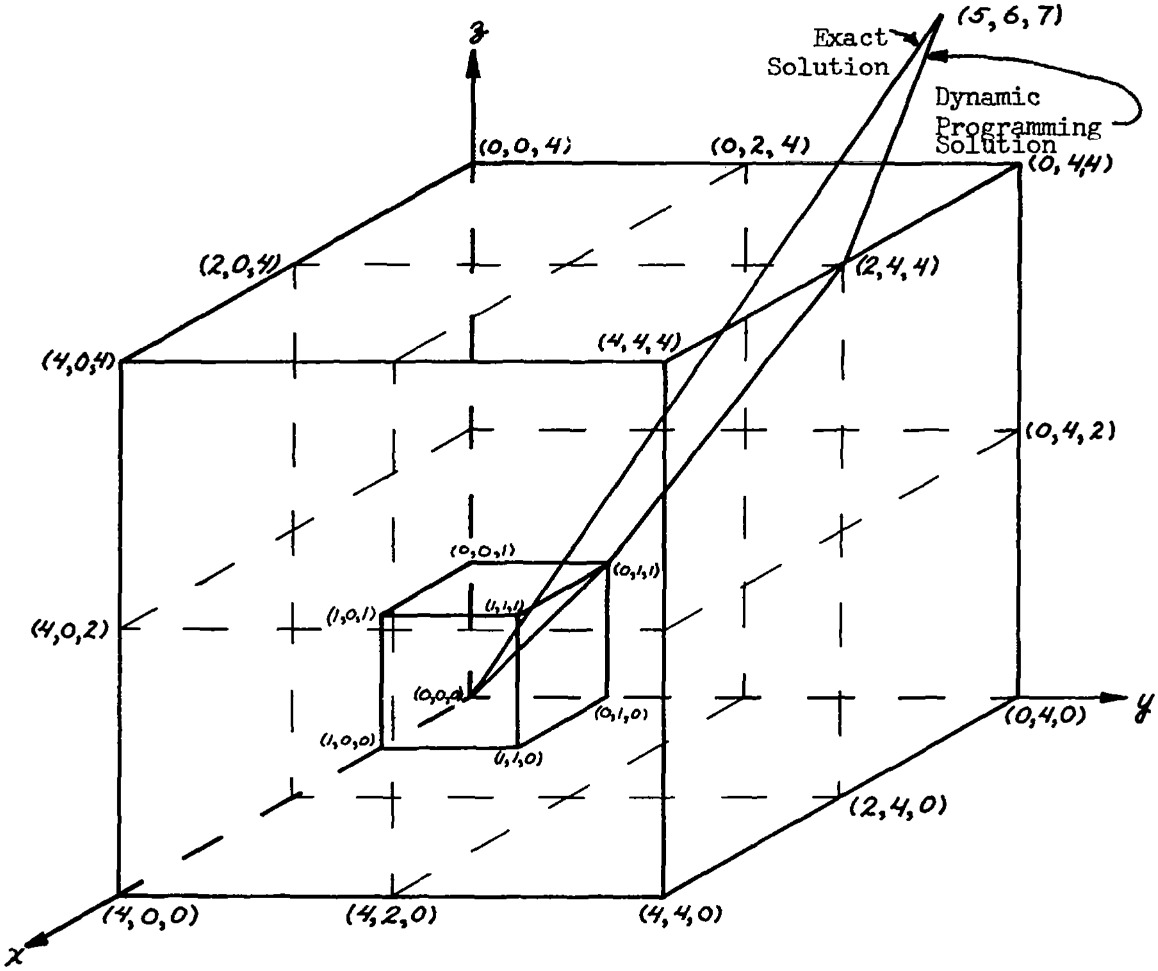
In a completely analogous manner the minimum cost for each of the other points on the first layer can be found. They are shown below.

	MINIMUM	
POINT	COST	VIA
(0, 0, 1)	9.953	(2, 2, 4)
(0, 1, 0)	10.075	(2, 4, 4)
(0, 1, 1)	9.380	(2, 4, 4)
(1, 0, 0)	10.144	(4, 4, 4)
(1, 0, 1)	9.571	(2, 2, 4)
(1, 1, 0)	9.571	(4, 4, 4)
(1, 1, 1)	9.145	(2, 2, 4)

The optimum value of the cost of the origin is found similarly by computing its cost for the various paths from layer 1 and by using the optimal values of those points. The following table shows those values:

			MINIMUM
POINT (0, 0, 0)	PATH	COST	COST
	(0, 0, 1)	1,000 + 9.953 = 10.953	
	(0, 1, 0)	1.000 + 10.075 = 11.075	
	(0, 1, 1)	1.414 + 9.380 = 10.794	10.794
	(1, 0, 0)	1.000 + 10.144 = 11.144	
	(1, 0, 1)	1.414 + 9.571 = 10.985	
	(1, 1, 0)	1.414 + 9.571 = 10.985	
	(1, 1, 1)	1.732 + 9.146 = 10.878	

The solution is now complete. The optimum path can be found by tracing back the optimum values from the previous tables. The optimum path to the origin from layer 1 is seen to be via point (0, 1, 1) from the previous table. With this information the optimum path to (0, 1, 1) can be found to be via point (2, 4, 4) from Table 2.2.1. This path is shown in the following sketch along with the exact solution.



The exact value of the minimal distance between (0, 0, 0) and (5, 6, 7) can easily be found from the Pathagorean Theorem as

$$J = \sqrt{(5-0)^2 + (6-0)^2 + (7-0)^2} = \sqrt{110} = 10.488$$

This value can be compared to the 10.794 that was obtained from Dynamic Programming approach. This difference is consistent with previous comments which were made on the accuracy of Dynamic Programming solutions and the effects of discretizing the space.

2.2.2.2 Variational Problem with Movable Boundary

Dynamic Programming will now be applied to the solution of a variational problem with a movable boundary. Consider the minimization of the functional

$$J(x,y) = \int_0^{x_f} \frac{\sqrt{1-(y')^2}}{y}\, dx \tag{2.2.6}$$

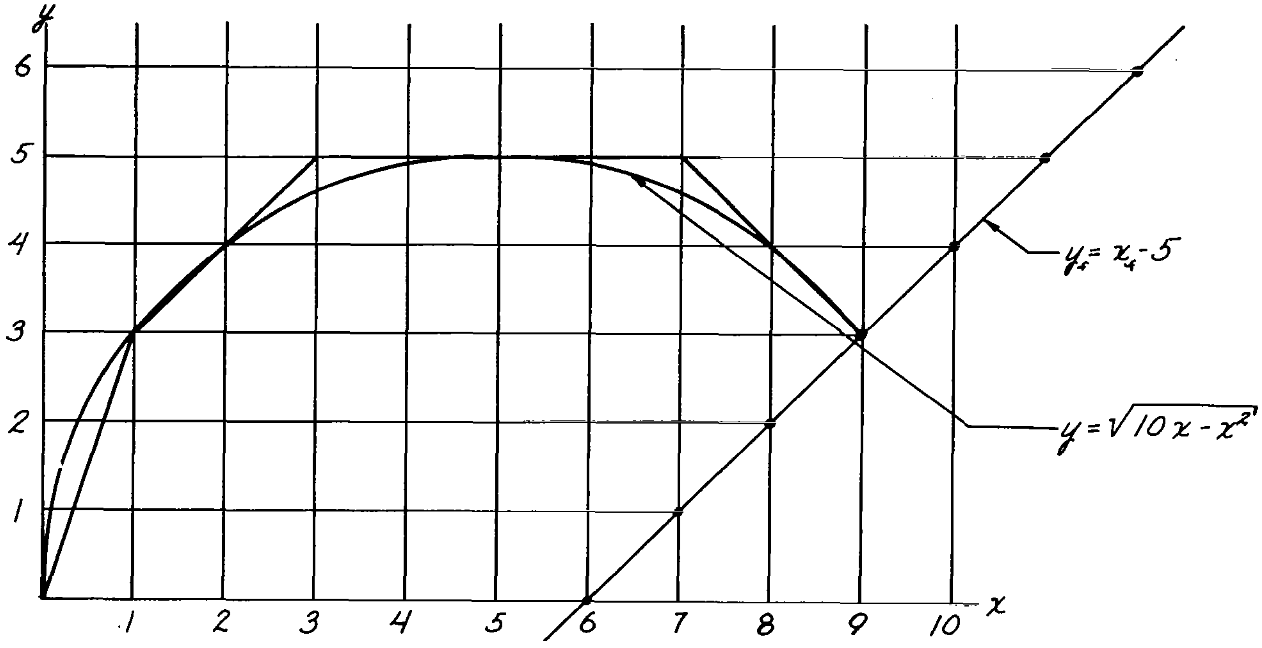
subject to the constraint that

$$y(0) = 0$$

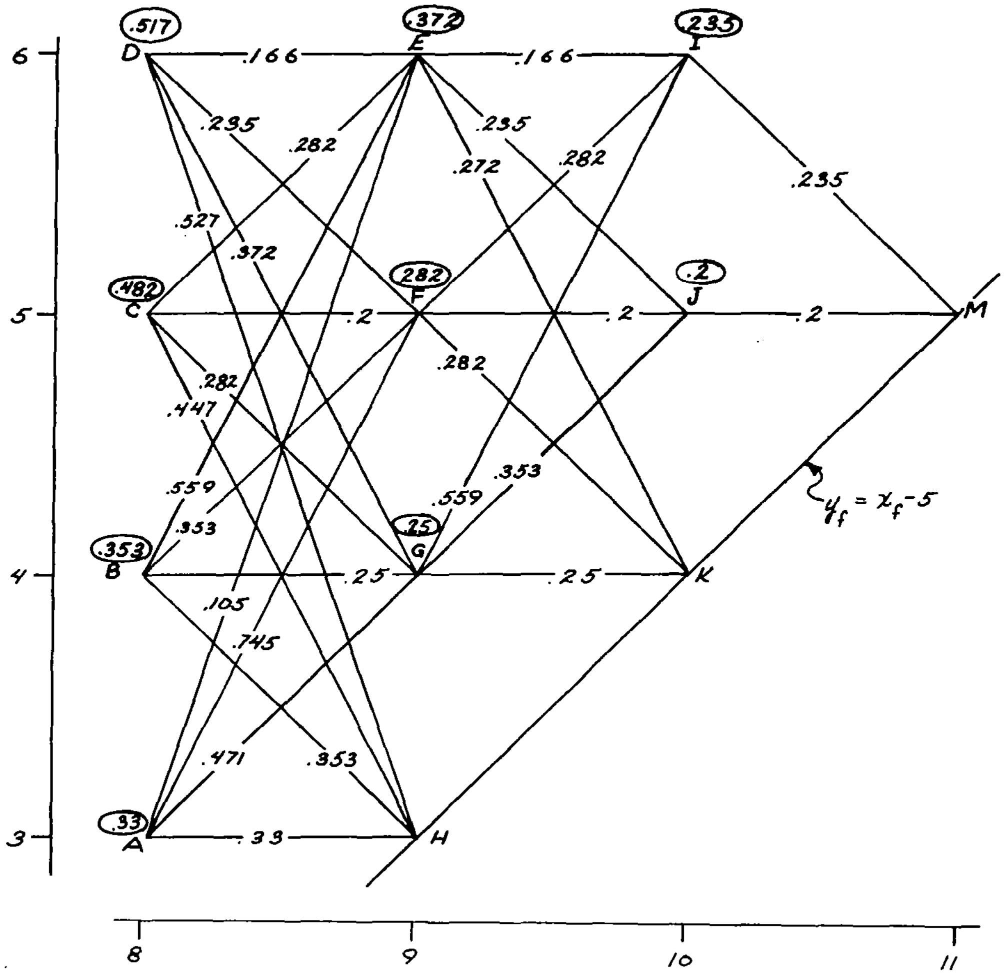
$$y_f = x_f - 5$$

This problem appears as an example in Ref. 2.2.2 to illustrate the classical solution of a problem with a movable boundary. Note that this problem differs in concept from the preceding problem in that the upper limit of integration is not explicitly specified. However, as would be suspected from previous problems, the Dynamic Programming approach still involves the division of the space into segments and the calculation of the cost of each transition. The set of end points is located on the line $y_f = x_f - 5$. As mentioned earlier, there are two ways to perform the Dynamic Programming calculations in most problems. One method initiates the computation at the first stage and progresses to the last stage; the second method begins the computation at the end of the process and progresses to the first stage. Both methods are equivalent and yield the same answers as shown in an earlier example. The following example will be partially solved by using the second method. (The number of computations prohibits the complete manual solution.) The other problems in this section use the first method.

To begin the Dynamic Programming solution, the space is divided as shown in the following sketch.



The circle $y = \sqrt{10x - x^2}$ is the classical solution to the stated problem. The line segments that follow the solution represent the expected Dynamic Programming solution. The computation begins with the calculation of the cost of all the possible transitions between the various points in the space. The minimum cost of each point is then determined in the same manner as previous problems. The difference between this and previous problems is that there is a set of possible terminal points. This generality does not introduce any problems in the method of attack. It merely means that the optimum value of all the possible terminal points must be investigated and the best one must be selected. The following sketch shows the details for part of the computation that begins on the line y = x-5 and progress to the left.



The cost integral is represented by

$$\sum_i \frac{\sqrt{1+(y_i')^2}}{y_i}\Delta x_i$$

instead of the continuous form in Equation 2.2.6. The cost of each transition is shown, and the optimal value of the cost of each point is encircled. The possible transitions that must be considered for each of the above points are shown below.

I

I-M	.235	= [✓ .235]

J

J-M	.2	= [✓ .200]

E

E-I	.166 + .235	= .401
E-J	.235 + .2	= [✓ .435]
E-K	.372	= .372

F

F-I	.282 + .235	= .517
F-J	.2 + .2	= .4
F-K	.282	= [✓ .282]

G

G-I	.559 + .235	= .794
G-J	.352 + .2	= .552
G-K	.25	= [✓ .250]

D

D-E	.166 + .372	= .538
D-F	.235 + .282	= [✓ .517]
D-G	.372 + .25	= .622
D-H	.527	= .527

C

C-E	.282 + .372	= .654
C-F	.2 + .282	= [✓ .482]
C-G	.282 + .25	= .532
C-H	.447	= .447

B

B-E	.559 + .372	= .931
B-F	.353 + .282	= .635
B-G	.25 + .25	= .50
B-H	.353	= [✓ .353]

A

A-E	1.05 + .372	= 1.422
A-F	.745 + .282	= 1.027
A-G	.471 + .25	= .721
A-H	.33	= [✓ .33]

This process continues until the optimal path can be found by following the decisions that were made by starting at the origin and progressing to the right.

2.2.2.3 Simple Guidance Problem

As an example of the application of Dynamic Programming to a problem of the Mayer type, a simple guidance problem will be examined. Consider a throttlable vehicle with some initial condition state vector X(0) where

$$X(0) = \begin{bmatrix} x(0) \\ y(0) \\ \dot{x}(0) \\ \dot{y}(0) \end{bmatrix} \tag{2.2.7}$$

and some initial mass m_o. It is desired to guide the vehicle to some terminal point x(f), y(f) subject to the constraint that its terminal velocity vector is a certain magnitude, i.e.

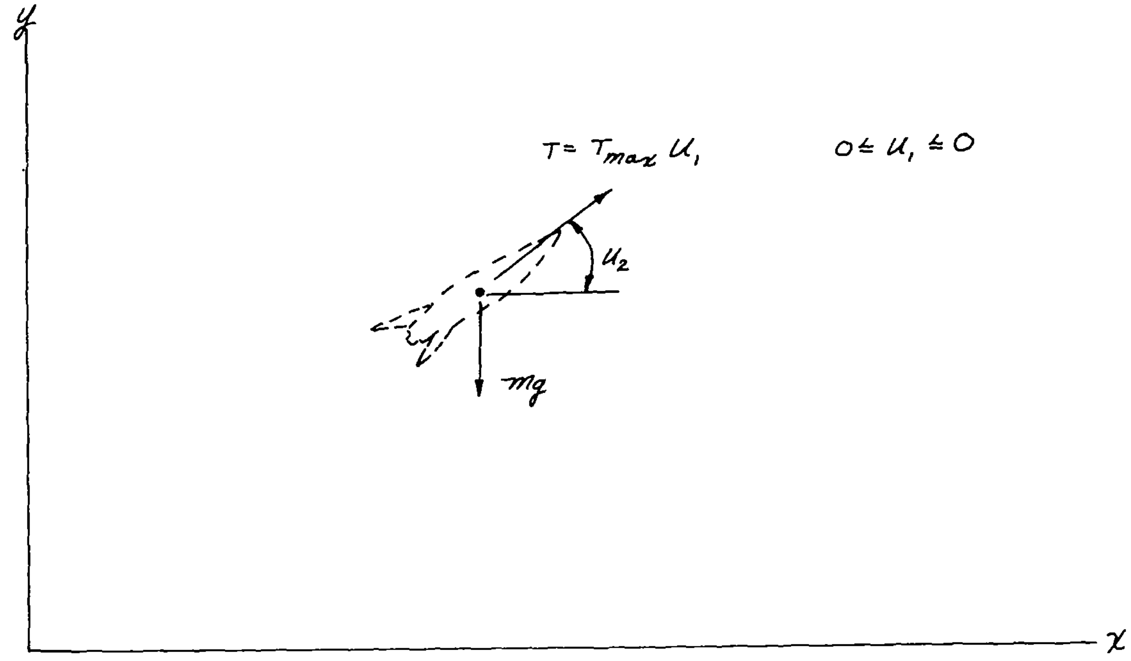
$$\left.\begin{aligned} x &= x_f \\ y &= y_f \\ \dot{x}^2 + \dot{y}^2 &= V_f^2 \end{aligned}\right\} \text{at} \quad t = t_f \tag{2.2.8}$$

where t_f is not explicitly specified.

Further, it is desired to minimize the amount of propellant that is used in order to acquire these terminal conditions (this problem is equivalent to maximizing the burnout mass). In order to simplify the problem, a flat earth will be assumed and the vehicle being considered will be restricted to two control variables U_1 and U_2. U_1 is a throttle setting whose range is $0 \leq U_1 \leq 1$. This variable applies a thrust to the vehicle equal to

$$T = T_{max} U_1 \tag{2.2.9}$$

where T_{max} is the maximum thrust available. U_2 is the control variable that governs the direction of thrust. This variable is defined as the angle between the thrust vector and the horizontal. The following sketch shows the geometry of these parameters.


y
$T = T_{max} u_1$
$0 \leq u_1 \leq 0$
u_2
mg
x

From this sketch, the following differential equations can be written for the motion of the vehicle

$$\ddot{x} = \frac{T_{max}}{m} u_1 \cos u_2 \qquad (2.2.10)$$

$$\ddot{y} = \frac{T_{max}}{m} u_1 \sin u_2 - g \qquad (2.2.11)$$

and

$$\dot{m} = -\frac{T_{max} u_1}{V} \qquad (2.2.12)$$

where T_{max} = the maximum thrust available

V = the exhaust velocity of the rocket.

There are several ways to formulate this problem for a Dynamic Programming solution. The method used here is to represent the state of the vehicle by four parameters, x, y, $\dot{x}$, and $\dot{y}$. The mass is used as a cost variable. The four dimensional state space is divided into small intervals in each coordinate direction. The coordinates designated by all the combinations of various intervals form a set of points in the state space. The vehicle starts at the initial point in the state space with some initial mass. The control and mass change that are necessary to move the vehicle from this point to the first allowable set of points in the state space are then computed. This computation corresponds to the first set of possible control decisions. Each end point of the set of possible first decisions is assigned a mass (cost) and the path that gave the cost (for the first decision the path is obvious since it must have come from the origin).

The second decision is now investigated. The required control and the corresponding mass change required to go from the set of points at the end of the first decision to the set of all possible points at the end of the second decision must now be calculated. (The initial mass used in this second stage calculation is the mass remaining at the end of the first stage.) However, each point corresponding to the end of the second decision will have more than one possible value of mass (depending on the point from which it came). Thus, since it is desired to minimize the fuel consumed or maximize the burnout mass, the largest mass is chosen as the optimum value for that particular point. The point from which this optimum path came is then recorded.

This process continues in the same manner until an end point is reached. In this case, the end point is a set of points all of which have the same coordinates for x and y but have many combinations of $\dot{x}$ and $\dot{y}$ subject to the constraint that

$$\dot{x}^2 + \dot{y}^2 = V_f^2$$

After the optimum mass is calculated for all possible terminal points, the best one is selected. The optimum path is then traced to the initial point just as was done in previous problems.

Formulation

The equations to be used to calculate the cost of each transition can be developed from Equations 2.2.10, 2.2.11 and 2.2.12. Since the transitions from one point to another point in the state space are assumed to be short in duration, it is assumed that the vehicle's mass is constant during the transition and that the acceleration in the x and y direction is constant. This is a reasonable assumption since the state space is divided into many smaller parts and the mass change is not very significant during a typical transition from one point to another. Thus, since the mass is practically constant and the control by the nature of its computation is constant during a transition, a constant acceleration is a reasonable assumption for a typical transition.

The laws of constant acceleration motion can now be used for each short transition. The acceleration that is required in order to force a particle to position x_2 with the velocity $\dot{x}_2$ at that position from a position x_1 with an initial velocity of x_1 is

$$\ddot{x} = \frac{\dot{x}_2^2 - \dot{x}_1^2}{2(x_2 - x_1)} \tag{2.2.13}$$

Similarly, in the y direction

$$\ddot{y} = \frac{\dot{y}_2^2 - \dot{y}_1^2}{2(y_2 - y_1)} \tag{2.2.14}$$

Now, recalling that the x and y components of thrust are

$$\ddot{x} = \frac{T_{max}}{m} u_1 \cos u_2$$

$$\ddot{y} = \frac{T_{max}}{m} u_1 \sin u_2 - g$$

equations 2.2.13, 2.2.14, 2.2.10 and 2.2.11 yield

$$u_1 \cos u_2 = \frac{m}{T_{max}}\left[\frac{\dot{x}_2^2 - \dot{x}_1^2}{2(x_2 - x_1)}\right] = C_x \qquad (2.2.15)$$

$$u_1 \sin u_2 = \frac{m}{T_{max}}\left[\frac{\dot{y}_2^2 - \dot{y}_1^2}{2(y_2 - y_1)} + g\right] = C_y \qquad (2.2.16)$$

The control parameters are thus,

$$u_1 = (C_x^2 + C_y^2)^{1/2} \qquad (2.2.17)$$

$$u_2 = \arctan \frac{C_y}{C_x} \qquad (2.2.18)$$

Now that the throttle control is known, $\dot{m}$ for the particular transition is

$$\dot{m} = -\frac{T_{max}\, u_1}{V} \qquad (2.2.19)$$

and the mass at the end of the transition is found to be

$$m_{i+1} = m_i + \dot{m}\,\Delta t \qquad (2.2.20)$$

where Δt can be found to be

$$\Delta t = \frac{2(x_2 - x_1)}{\dot{x}_2 + \dot{x}_1} \qquad (2.2.21)$$

The algorithm for solving this problem will now be discussed. First, the state space must be divided properly. To do this, an increment measure for each coordinate must be defined. Let $\hat{x}_i$ be the increment measure of the $i\underline{th}$ coordinate so the extent of each coordinate is $L\hat{x}_1, M\hat{x}_2, N\hat{x}_3, P\hat{x}_4$ where L, M, N, and P are integers that are large enough to make the maximum value of each coordinate as close as possible to the maximum value needed by that coordinate without exceeding that maximum value. For instance, if the maximum value required by x_1 is 51, 324 ft. and it was decided to use $\hat{x}_1$ = 1,000 ft., then L would be chosen as 51. Since there is a set of terminal points, N and P must be chosen to accommodate max ($\dot{x}_f$) and max ($\dot{y}_f$), respectively.

The cost of all points $(l\hat{x}_1 , m\hat{x}_2 , n\hat{x}_3 , p\hat{x}_4)$

$$l = 0, 1, \ldots, L$$
$$m = 0, 1, \ldots, M$$
$$n = 0, 1, \ldots, N$$
$$p = 0, 1, \ldots, P$$

must be found as previously discussed. For this particular case, the initial point will be assumed to be (0,0,0,0). The set of states that can result from the first decision includes the following points:

$(0, 0, 0, \hat{x}_4)$

$(0, 0, \hat{x}_3, 0)$

$(0, 0, \hat{x}_3, \hat{x}_4)$

$(0, \hat{x}_2, 0, 0)$

$(0, \hat{x}_2, 0, \hat{x}_4)$

$(0, \hat{x}_2, \hat{x}_3, 0)$

$(0, \hat{x}_2, \hat{x}_3, \hat{x}_4)$

$(\hat{x}_1, 0, 0, 0)$

$(\hat{x}_1, 0, 0, \hat{x}_4)$

$(\hat{x}_1, 0, \hat{x}_3, 0)$

$(\hat{x}_1, 0, \hat{x}_3, \hat{x}_4)$

$(\hat{x}_1, \hat{x}_2, 0, 0)$

$(\hat{x}_1, \hat{x}_2, 0, \hat{x}_4)$

$(\hat{x}_1, \hat{x}_2, \hat{x}_3, 0)$

$(\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4)$

As mentioned previously, the mass of the vehicle is computed for each point and is stored along with the control that was needed to get there.

The second decision must come from a wealth of possibilities. If an approach similar to the shortest distance problem is taken (where the only permissible transitions eminate in rays from the initial point of the particular transition), a reasonable set of transitions for each point is obtained. To give the reader an idea of the number of points which are possible, the following table was constructed to show this set of points by using a shorthand notation for convenience (where $\textcircled{2} = 0, X_i, 2X_i$).

FROM	TO
$(0, 0, 0, X_4$	$(\textcircled{2}, \textcircled{2}, \textcircled{2}, 2X_4)$
$(0, 0, X_3, 0)$	$(\textcircled{2}, \textcircled{2}, 2X_3, \textcircled{2})$
$(0, 0, X_3, X_4)$	$(\textcircled{2}, \textcircled{2}, \textcircled{2}, 2X_4)$ or $(\textcircled{2}, \textcircled{2}, 2X_3, \textcircled{2})$
$(0, X_2, 0, 0)$	$(\textcircled{2}, 2X_2, \textcircled{2}, \textcircled{2})$
$(0, X_2, 0, X_4)$	$(\textcircled{2}, 2X_2, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, \textcircled{2}, \textcircled{2}, 2X_4)$
$(0, X_2, X_3, 0)$	$(\textcircled{2}, \textcircled{2}, 2X_3, \textcircled{2})$ or $(\textcircled{2}, 2X_2, \textcircled{2}, \textcircled{2})$
$(0, X_2, X_3, X_4)$	$(\textcircled{2}, 2X_2, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, \textcircled{2}, 2X_3, \textcircled{2})$
	or $(\textcircled{2}, \textcircled{2}, \textcircled{2}, 2X_4)$
$(X_1, 0, 0, 0)$	$(2X_1, \textcircled{2}, \textcircled{2}, \textcircled{2})$
$(X_1, 0, 0, X_4)$	$(2X_1, \textcircled{2}, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, \textcircled{2}, \textcircled{2}, 2X_4)$
$(X_1, 0, X_3, 0)$	$(2X_1, \textcircled{2}, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, \textcircled{2}, 2X_3, \textcircled{2})$
$(X_1, 0, X_3, X_4)$	$(2X_1, \textcircled{2}, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, \textcircled{2}, 2X_3, \textcircled{2})$
	or $(\textcircled{2}, \textcircled{2}, \textcircled{2}, 2X_4)$
$(X_1, X_2, 0, 0)$	$(2X_1, \textcircled{2}, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, 2X_2, \textcircled{2}, \textcircled{2})$
$(X_1, X_2, 0, X_4)$	$(2X_1, \textcircled{2}, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, 2X_2, \textcircled{2}, \textcircled{2})$
	or $(\textcircled{2}, \textcircled{2}, \textcircled{2}, 2X_4)$
$(X_1, X_2, X_3, 0)$	$(2X_1, \textcircled{2}, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, 2X_2, \textcircled{2}, \textcircled{2})$
	or $(\textcircled{2}, \textcircled{2}, 2X_3, \textcircled{2})$
(X_1, X_2, X_3, X_4)	$(2X_1,, \textcircled{2}, \textcircled{2}, \textcircled{2})$ or $(\textcircled{2}, 2X_2, \textcircled{2}, \textcircled{2})$
	or $(\textcircled{2}, \textcircled{2}, 2X_3, \textcircled{2})$ or $(\textcircled{2}, \textcircled{2}, \textcircled{2}, 2X_4)$

The reader, no doubt, has a reasonable idea of the number of points that must be investigated for the second decision. This number continues to grow at a tremendous rate for subsequent decisions since each point at the end of the second stage is an initial point for the third stage and because the ② becomes a ③(③ = 0, x_i, 2 x_i, 3 x_i) for the terminal points of the third stage. This fantastic increase in computation points is called the "curse of dimensionality" of Dynamic Programming. It stems from the fact that as the number of dimensions of the state space increases; the number of computation points of the problem increases as a^n where "n" is the dimension of the state space and "a" is the number of increments used for a typical coordinate. Section 2.3.3.1 will discuss dimensionality in more detail.

2.2.3 Maximum - Minimum Problem

In order to demonstrate the more analytical applications of Dynamic Programming, a simple Maxima-Minima Problem will be examined. The procedure utilized to formulate a problem for the application of Dynamic Programming is not always immediately obvious. Many times the problem formulation for a Dynamic Programming solution is quite different from any other approach. The following problem will be attacked in a manner such that the Dynamic Programming formulation and method of attack can be seen.

The problem is to minimize the expression

$$S = x_1^2 + 2x_2^2 + x_3^2 \tag{2.2.22}$$

subject to the constraints

$$x_1 + x_2 + x_3 = 10$$

$$x_i \geq 0$$

(A problem similar to this is often used by Dr. Bellman to introduce the concepts of Dynamic Programming). At first glance, the methods of Dynamic Programming do not seem to apply to this problem. However, if the problem is reduced to several smaller problems, the use of Dynamic Programming becomes apparent. Consider the minimization of the following three functions:

$$f_1 = x_1^2 \tag{2.2.23}$$

$$f_2 = f_1 + 2x_2^2 \tag{2.2.24}$$

$$f_3 = f_2 + x_3^2 \tag{2.2.25}$$

Applying the constraints $\{x_1 = 10,\ x_1 \geq 0\}$ to the first function gives the trivial result $x_1 = 10$. This result is not so helpful. However, if the constraints

$$x_1 = A_1 \qquad\qquad 0 \leq A_1 \leq 10$$

$$x_1 \geq 0$$

are applied to Min (f_1), the range of Min (f_1) can be found for various admissible values of x_1. This step can be thought of as the first stage of a three stage decision process, where the various choices for x_1 can be as many as desired within the limits $0 \leq x_1 \leq 10$.

Now that all the choices for the first decision have been investigated, the second decision must be considered. Again, care must be taken in the specification of the constraint equations. The optimal value function for the second decision is

$$f_2 = f_1 + 2x_2^2 \tag{2.2.26}$$

The constraints on x_1 and x_2 are chosen to be

$$x_1 + x_2 = A_2 \qquad 0 \leq A_2 \leq 10 \tag{2.2.27}$$

For each value of A_2 that is to be investigated, there are many combinations of x_1 and x_2 to be considered. More precisely, the number of combinations of x_1 and x_2 to be considered is the same as the number of A_1's that are less than or equal to the A_2 being considered, e.g., if A_1 was investigated for $A_1 = 0, 5, 10$ and $A_2 = 5$ is being considered, then the following combinations must be examined

x_1	x_2
0	5
5	0

The third decision does not require as much computation as the others in this case because of the original constraint equations.

$$\begin{gathered} x_1 + x_2 + x_3 = 10 \\ x_i \geq 0 \end{gathered} \tag{2.2.28}$$

Since the first two decisions were investigated for many possible values of x_1 and $x_1 + x_2$, it is only necessary to consider

$$x_1 + x_2 + x_3 = 10$$

because the various choices for x_3 will specify $x_1 + x_2 = 10 - x_3$ and these possibilities have already been investigated.

The arithmetic solution will now be shown so that the previous discussion will be clear. For simplicity, only integers will be considered for the allowable values for x_1, x_2, and x_3. A table can be constructed for the first decision as follows:

x_1	0	1	2	3	4	5	6	7	8	9	10
f_1	0	1	4	9	16	25	36	49	64	81	100

The table for various combinations of x_1 and x_2 for f_2 is shown below:

f_2

x_2 \ x_1	0	1	2	3	4	5	6	7	8	9	10
0	0	[✓ 1]	4	9	16	25	36	49	64	81	100
1	2	[✓ 3]	[✓ 6]	[✓ 11]	18	27	38	51	66	83	
2	8	9	12	[✓ 17]	[✓ 24]	[✓ 33]	44	57	72		
3	18	19	22	27	34	[✓ 43]	[✓ 54]	[✓ 67]			
4	32	33	36	41	48	57	68				
5	50	51	54	59	66	75					
6	72	73	76	81	88						
7	98	99	102	107							
8	128	129	132								
9	162	163									
10	200										

Note that each diagonal corresponds to a particular value for $x_1+x_2 = A_1$. The optimum value for f_2 in each diagonal is encircled. This chart will be useful as soon as the graph for the third decision is found. It is shown below:

f_3

x_3 \ x_1+x_2	0	1	2	3	4	5	6	7	8	9	10
0											67
1										55	
2									47		
3								42			
4							40				
5						42					
6					47						
7				55							
8			67								
9		82									
10	100										

NOT NEEDED FOR $x_1 + x_2 + x_3 = 10$

Since it was specified that $x_1 + x_2 + x_3 = 10$, only one diagonal is needed. The procedure to find the optimum values is now straightforward. From the previous table it is seen that optimum decision for x_3 is $x_3 = 4$, which means $x_1 + x_2 = 6$. This corresponds to a value of 40 for f_3. The optimum values for x_1 and x_2 can now be determined by referring to the table for the second decision. Since $x_1 + x_2 = 6$, the best value in the sixth diagonal must be selected. It is 24, which corresponds to $x_1 = 4$ and $x_2 = 2$. Thus, the optimum values for x_1, x_2, and x_3 have been determined to be $x_1 = 4$, $x_2 = 2$, $x_3 = 4$.

The question arises: is any savings realized by using Dynamic Programming for this problem? In order to answer this question, the number of computations using Dynamic Programming will be compared to that using the "brute force" method. (It should be noted, however, that small problems do not demonstrate the beauty of Dynamic Programming as well as larger problems. It will be shown in Section 2.3.3.1 that some problems that are virtually impossible to solve by the "brute force" method become reasonable once again due to Dynamic Programming concepts.) The number of additions performed in the previous problem were 66 for the second table, and 11 for the third table - for a total of 77 additions. The "brute force" method would require the calculation of $S = X_1^2 + 2X_2^2 + X_3^2$ for all possible permutations of x_1, x_2, and x_3 where $0 \leq x_i \leq 10$, $x_1 + x_2 + x_3 = 10$, and x_i is an integer. For this particular problem the "brute force" method requires 66 cases or 132 additions. It is seen that even on this simple problem the savings in additions is quite significant.

In order to compare and contrast the Dynamic Programming solution of this problem with the classical solution, the same problem will not be solved using classical techniques. First, the constraint equation is joined to the original problem by a Lagrange Multiplier.

$$S = x_1^2 + 2x_2^2 + x_3^2 - \lambda(x_1 + x_2 + x_3) \tag{2.2.29}$$

Now the partial derivatives are taken with respect to the independent variables and equated to zero.

$$\frac{\partial S}{\partial x_1} = 2x_1 - \lambda = 0 \tag{2.2.30a}$$

$$\frac{\partial S}{\partial x_2} = 4x_2 - \lambda = 0 \tag{2.2.30b}$$

$$\frac{\partial S}{\partial x_3} = 2x_3 - \lambda = 0 \tag{2.2.30c}$$

This yields

$$x_1 = \frac{\lambda}{2} \tag{2.2.31a}$$

$$x_2 = \frac{\lambda}{4} \tag{2.2.31b}$$

$$x_3 = \frac{\lambda}{2} \tag{2.2.31c}$$

The value of λ can be found by employing the constraint equation

$$x_1 + x_2 + x_3 = 10 = \frac{\lambda}{2} + \frac{\lambda}{4} + \frac{\lambda}{2} \tag{2.2.32}$$

Hence,

$$\lambda = 8$$

and finally

$$x_1 = \frac{8}{2} = 4 \tag{2.2.33a}$$

$$x_2 = \frac{8}{4} = 2 \tag{2.2.33b}$$

$$x_3 = \frac{8}{2} = 4 \tag{2.2.33c}$$

It is interesting to compare these two solutions. First, it should be noted that solutions obtained using the two methods on the same problem need not be the same. That the answers are identical for both methods in this problem results from the fact that the answers to the continuous problem happened to be integers and the Dynamic Programming method searched over all the permissible integers. Had the solution not consisted of a set of integers, the Dynamic Programming solution could have been forced to converge to the continuous solution by increasing the number of values employed for the variables in the process.

On the other hand, if it is desired that the solution consist of integers, the continuous method would not be a very effective way of determining the solution. The Dynamic Programming solution, of course, would be constructed without modification.

2.2.4 Equipment Replacement Problem

The following problem is included to illustrate the use of Dynamic Programming in solving problems in which the variables are in a tabular form rather than expressed analytically. The problem was presented by R. E. Kalaba in a course taught at U.C.L.A. during the spring of 1962 and is shown in Ref. 2.2.3.

Consider the position of the person who must decide whether to purchase a new machine for a factory or keep the old machine for another year. It is known that the profit expected from the machine in question decreases every year as follows:

Age at Start of Year	Net Profit from Machine During the Year
0	$10,000
1	9,000
2	8,000
3	7,000
4	6,000
5	5,000
6	4,000
7	3,000
8	2,000
9	1,000
10	0

Table 2.2.2 Expected Profit of Machine

A new machine costs $10,000. It is assumed that the old machine cannot be sold when it is replaced, and the junk value is exactly equal to the cost that is necessary to dismantle it. If the machine is now 3 years old, it is desired to find the yearly decision of keeping or replacing the machine such that the profit is maximized for the next 15 years.

The solution of this problem proceeds in a manner quite similar to previous problems. Instead of solving the specific problem for 15 years, the more general problem is solved for k years. The results of the k th year then provides information for the $k+1$ th decision. The mathematical statement of the optimization problem is as follows:

$$F_k(\alpha) = \max \begin{Bmatrix} \text{KEEP}: & P(\alpha) & + F_{k-1}(\alpha+1) \\ \text{REPLACE}: & \underbrace{P(0)-C}_{\substack{\text{PROFIT DURING}\\ \text{NEXT YEAR OF}\\ \text{OPERATION}}} & + \underbrace{F_{k-1}(1)}_{\substack{\text{PROFIT FROM}\\ \text{REMAINING}\\ \text{K-1 YEARS}}} \end{Bmatrix} \tag{2.2.34}$$

The initial condition is:

$$F_0(\alpha) = \max \begin{Bmatrix} \text{KEEP}: & 0 \\ \text{REPLACE}: & 0 \end{Bmatrix} = 0 \tag{2.2.35}$$

where $F_k(\alpha)$ = the expected profit obtained during the next k years when the machine is α years old at the start, and an optimal policy is used.

$P(\alpha)$ = the expected profit obtained during one year of operation of a machine α years old at the start of the year.

C = the cost of a new machine.

The numerical soltuion begins by evaluation $P(\alpha)$. This is easily done from Table 2.2.2.

P(0) =	$10,000	P(6) =	$ 4,000
P(1) =	9,000	P(7) =	3,000
P(2) =	8,000	P(8) =	2,000
P(3) =	7,000	P(9) =	1,000
P(4) =	6,000	P(10) =	0
P(5) =	5,000		

The cost of a new machine is $10,000 so

$$C = \$10,000$$

Now a table for the one stage process will be formed using all possible values for α .

AGE α		0	1	2	3	4	5	6	7	8	9	10
$f_1(\alpha)$	Keep	10,000	9,000	8,000	7,000	6,000	5,000	4,000	3,000	2,000	1,000	0
	Replace	0	0	0	0	0	0	0	0	0	0	0

Starting with a machine α years old, the profit for the year will be $P(\alpha)$ if the machine is is not replaced. If, on the other hand, the machine is replaced, the profit from the machine is $10,000; but it costs $10,000 to get a new machine, so the net profit for that year is 0. Hence, the result for a one stage process is to keep the machine regardless of how old it is in order to maximize the profit for one year.

Now a 2 stage process will be considered. Here, the question arises whether to keep or replace the machine at the beginning of each year for two years. Using the previous results, the following table for the 2 stage process can be constructed.

AGE α		0	1	2	3	4	5	6	7	8	9	10
$f_1(\alpha)$		10,000	9,000	8,000	7,000	6,000	5,000	4,000	3,000	2,000	1,000	0
DECISION		K	K	K	K	K	K	K	K	K	K	K
$f_2(\alpha)$	Keep	19,000	17,000	15,000	13,000	11,000	9,000	7,000	5,000	3,000	1,000	0
	Replace	9,000	9,000	9,000	9,000	9,000	9,000	9,000	9,000	9,000	9,000	9,000

A closer look at the computation of the numbers in this table will clarify the concepts involved. For an example, consider $\alpha = 2$. The decision faced here is to keep or replace a machine now that is to last for 2 years. If it is decided to keep the machine, the income from the first year is P(2) = $8,000. The decision for the last year has already been made on the 1 stage process (keep). The income from the second year is that of a machine 3 years old or $7,000 for a total income of $15,000 for two years. Now consider the "replace" decision for the beginning of the first year. The income from the machine for the first year is $10,000 and the cost of replacement is $10,000 so the profit during the first year is $0. The second year starts with a machine that is 1 year old, and the profit obtained is $9,000. The total profit for two years is thus $9,000. From Table 2.2.3

it is seen that (for a two stage process) a machine which is less than 5 years old should be kept, a machine which is more than 5 years old should be replaced and a machine which is exactly 5 years old can be kept or replaced. (In the indifferent case the machine will be kept by convention).

Repeating this procedure for a three stage process yields the following table.

AGE α		0	1	2	3	4	5	6	7	8	9	10
F_1 (α) DECISION		K	K	K	K	K	K	K	K	K	K	K
F_2 (α)		19,000	17,000	15,000	13,000	11,000	9,000	9,000	9,000	9,000	9,000	9,000
DECISION		K	K	K	K	K	K	R	R	R	R	R
F_3 (α)	KEEP	27,000	24,000	21,000	18,000	15,000	14,000	13,000	12,000	11,000	10,000	9,000
	REPLACE	17,000	17,000	17,000	17,000	17,000	17,000	17,000	17,000	17,000	17,000	17,000

The optimal policy can now be found by referring to the table. Note that the general solution is given; that is, the problem can begin with a machine of any age (not just 3 years old as in the original problem). This generality is the result of the fact that Dynamic Programming solves a class of problems rather than a specific problem. For a 15 stage process, the correct initial decision for the problem in which the machine is 3 years old is found in the grid $_{15}$ (α) and $\alpha = 3$ (marked by ①). For the next decision, F_{14} (α), the machine is 4 years old since it was kept for an additional year. The correct decision for this stage is again "keep" as shown by the grid marked by ② . The third decision is to "replace" as shown by the grid labeled by ③. The fourth decision is shown by ④ . The unit that was replaced in the third decision is one year old at the beginning of the fourth decision so the grid to use is F_{12} (α) and $\alpha = 1$. This process continues as is shown by the remaining circled numbers. The final policy for the 15-stage process that starts with a unit 3 years old is keep, keep, replace, keep, keep, keep, keep, replace, keep, keep, keep, replace, keep, keep, keep. The maximum profit for this problem is seen to be $91,000.

Similarly, for 15 stages, the following table results:

AGE	0	1	2	3	4	5	6	7	8	9	10
F_1 (α)	K	K	K	K ⑮	K	K	K	K	K	K	K
F_2 (α)	K	K	K ⑭	K	K	K	R	R	R	R	R
F_3 (α)	K	K ⑬	K	K	R	R	R	R	R	R	R
F_4 (α)	K	K	K	K	R ⑫	R	R	R	R	R	R
F_5 (α)	K	K	K	K ⑪	K	R	R	R	R	R	R
F_6 (α)	K	K	K ⑩	K	K	K	R	R	R	R	R
F_7 (α)	K	K ⑨	K	K	K	R	R	R	R	R	R
F_8 (α)	K	K	K	K	R	R ⑧	R	R	R	R	R
F_9 (α)	K	K	K	K	K ⑦	R	R	R	R	R	R
F_{10} (α)	K	K	K	K ⑥	K	R	R	R	R	R	R
F_{11} (α)	K	K	K ⑤	K	K	K	R	R	R	R	R
F_{12} (α)	K	K ④	K	K	R	R	R	R	R	R	R
F_{13} (α)	K	K	K	K	K	R ③	R	R	R	R	R
F_{14} (α)	K	K	K	K	K ②	R	R	R	R	R	R
F_{15} (α)	K	K	K	K ①	K	R	R	R	R	R	R
	100,000	96,000	93,000	91,000	90,000	90,000	90,000	90,000	90,000	90,000	90,000

2.3 COMPUTATIONAL CONSIDERATIONS

So far, the principles of Dynamic Programming have been applied to both discrete and continuous problems. It was shown in Section 2.2.2 that Dynamic Programming is an alternative method of solving certain variational problems. In fact, the use of Dynamic Programming sometimes enables the solution of problems that are normally very difficult, if not impossible, by classical techniques. It should not be assumed, however, that its use is free of difficulties. Dynamic Programming does indeed suffer from difficulties that are inherent in any scheme that discritizes a problem or performs a combinational search. This section discusses the relative advantages and disadvantages of Dynamic Programming as applied to both continuous and discrete problems.

2.3.1 MERITS OF DYNAMIC PROGRAMMING

The classical techniques used in optimization theory are subject to many complications when they are applied to physical problems. These difficulties result from applications of the theory based on continuous, well-behaved functions to problems involving discontinuities and relationships for which there are no closed-form analytical expressions. This section deals with these classical techniques and discusses the relative merits of Dynamic Programming on these points.

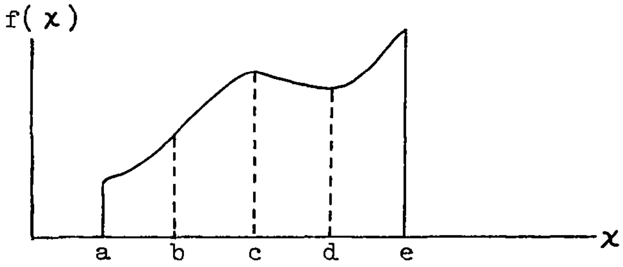
2.3.1.1 Relative Extrema

The difficulty in trying to distinguish between relative extrema, absolute extrema, and inflection is well known to the calculus student who sets the first derivative equal to zero. This difficulty, which is a nuisance in functions of one variable, becomes almost unbearable for functions of many variables. (Such cases are encountered in the optimization of a multi-variable problem.) The use of Dynamic Programming on problems such as these avoids this difficulty completely. The very nature of Dynamic Programming deals only with absolute maxima or minima; so far as the Dynamic Programming solution is concerned, other extrema do not even exist.

This property of Dynamic Programming turns out to be the only salvation in the solution of multi-dimensional problems in which there are many extrema.

2.3.1.2 Constraints

Classical techniques fail to give the necessary extrema when they occur on a constraint point. This fact can be seen most easily by examining the following sketch of one variable that has an extrema on a constraint point.



If classical techniques were to be used to determine the extrema, the values of f(b), f(c), and f(d) would be obtained. That is, since the derivative at x = e is not zero, the extremum is not located with classical techniques. Such a function is quite common in practical problems such as control problems or economic problems where there is a very distinct limit to the range that a variable can have. This fact poses a problem to the engineer who attempts to optimize a process that includes functions of this sort; therefore, he must be very careful when using classical techniques. If he is aware of the possible existance of other extrema, precautionary measures can be taken to guarantee that the extremum which is located analytically, in fact, is the extremum.

Again, the techniques of Dynamic Programming avoid these problems completely. The reason for this is that all functions are represented discretely and the optimum values are found by a search technique over a set of numbers that represent the cost of a various number of policies. Thus, the procedure escapes the problems associated with the introduction of an imprecise condition by merely selecting the optimum number.

2.3.1.3 Continuity

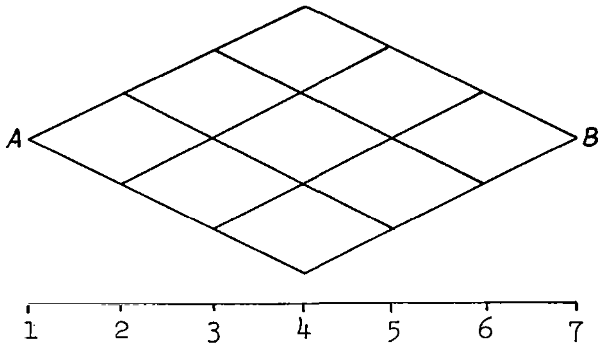
The application of classical techniques on problems involving functions with discontinuities and with discontinuous derivatives also introduces difficulties. Since the tools of calculus are directed at continuous variations in variables, it is sometimes useful to smooth the discontinuities in physical problems so that classical techniques can be used. However, in some cases, the accuracy of the solution is seriously affected by such smoothing. Further, many functions that are ideally represented by discontinuities in the variables must be handled in a special manner in the analytical solution.

The techniques of Dynamic Programming also surmount these problems since the discrete manner in which the functions are used is not affected by discontinuities so long as the representations of the discontinuities are not ambiguous.

2.3.2 DYNAMIC PROGRAMMING VERSUS STRAIGHTFORWARD COMBINATIONAL SEARCH

The application of Dynamic Programming techniques to a problem of more than two dimensions usually provokes some thought on the advantages of Dynamic Programming over the so-called "brute force" method of searching all of the possible combinations of decisions and selecting the best. Surely, the overwhelming number of computations involved appear to classify this approach as a near "brute force" method even when using the techniques of Dynamic Programming. If a calculation comparison is made, however, it will be seen that such a statement is not justified. The computational savings offered by Dynamic Programming makes soluble some problems that are physically impossible to attempt with a straightforward combinational search because the exorbitant number of computations.

In order to see the relative merits of Dynamic Programming in a small problem, consider the problem of finding the optimum path from point A to point B in the following sketch.



Decision Points (Stages)

The brute force method of solving this problem would be to evaluate the cost of each of the 20 possible paths that could be taken. Since there are six segments per path, there will be five additions per path or a total of 100 additions and one search over 100 numbers for a complete solution. The same problem can be solved by Dynamic Programming (see Section 2.2.1) by performing two additions and one comparison at each of the nine points where a decision was needed and one addition at the remaining six points. This approach results in 24 additions and six comparisons as opposed to 100 additions and one search which were necessary with the brute force method.

This comparison can be performed for an n stage process (the previous example was a six stage process). The expression for the number of additions for the Dynamic Programming approach is $\frac{n^2}{2}+n$. The brute force method involves $(n-1)\,n!/(\frac{n}{2}!)^2$ additions. Using these expressions, the merits of Dynamic Programming begin to become very evident as n increases. For instance, the 20-stage process would require 220 additions using Dynamic Programming as opposed to 3,510,364 additions by the brute force method.

2.3.3 DIFFICULTIES ENCOUNTERED IN DYNAMIC PROGRAMMING

It should not be assumed that because Dynamic Programming overcomes the difficulties discussed in Section 2.2.1, that it is the answer to all optimization difficulties. To the contrary, many problems are created by its use. The following section discusses some of the difficulties encountered when Dynamic Programming is applied to multi-dimensional optimization problems.

2.3.3.1 The Curse of Dimensionality

In Section 2.2.2.3 a simple guidance problem is presented. It is pointed out in that section that the number of computations involved was quite large because of the four dimensional nature of the state space. In general, the number of computation points increases as a^n, where a is the number of increments in one dimension and n is the number of dimensions in the space. With the limited storage capabilities of modern digital computers, it is not difficult to realize that a modest multi-dimensional problem can exceed the capacity of the computer very easily, even with the methods of Dynamic Programming. This impairment does not prevent the solution of the problem; however, it means that more sophisticated techniques must be found in order to surmount this difficulty. Although this field has had several important contributions, it is still open for original research.

One of the more promising techniques that can be used to overcome dimensionality difficulties is the method of successive approximations. In analysis, this method determines the solution to a problem by first assuming a solution. If the initial guess is not the correct solution, a correction is applied. The correction is determined so as to improve the previous guess. The process continues until it reaches a prescribed accuracy.

The application of successive approximations to Dynamic Programming takes form as an approximation in policy space. The two important unknown functions of any Dynamic Programming solution are the cost function and the policy function. These two equations are dependent on each other, i.e., one can be found from the other. This relation is used to perform a successive approximation on the solution of the policy function by guessing at an initial solution and iterating to the correct solution. (This technique is called approximation in policy space.) It should be noted that such a procedure sacrifices computation time for the sake of reducing storage requirements.

The use of approximation in policy space will be illustrated via an allocation problem. Mathematically, the two dimensional allocation problem can be stated as finding the policy $\{x_i\}$ that minimizes

$$f_N(x, y) = \sum_{i=1}^{N} g_i(x_i, y_i) \qquad (2.3.1)$$

subject to the condition

$$\sum_{i=1}^{N} x_i = x \qquad x_i \geq 0 \qquad (2.3.2)$$

$$\sum_{i=1}^{N} y_i = y \qquad y_i \geq 0 \qquad (2.3.3)$$

In order to give an appreciation for the need for more sophisticated techniques, a sample problem will be worked by the Dynamic Programming techniques which have been discussed. The presentation will serve two purposes: first, it will illustrate the use of Dynamic Programming on a multi-dimensional allocation problem and, second, it will demonstrate the rapid growth of storage requirements as a function of the dimension of the problem. The method of approximation in policy space will then be discussed in order to illustrate the savings in storage requirements and the increase in computation time.

Consider the problem of minimizing the function

$$f = (x_1^2 + y_1^2) + (x_2^2 + y_2^2) + (x_3^2 + y_3^2) \qquad (2.3.4)$$

subject to the constraint that

$$x_1 + x_2 + x_3 = 3 \qquad (2.3.5)$$

and

$$y_1 + y_2 + y_3 = 3 \qquad (2.3.6)$$

Obviously, using Dynamic Programming to find a solution to this problem is not very efficient. The method of Lagrange multipliers is by far a more suitable method. However, the Dynamic Programming solution will be shown for illustrative purposes.

First, the problem is reduced to a series of simpler problem.

$$f_1 = x_1^2 + y_1^2 \qquad \left\{ \begin{array}{ll} x_1 \leq A_1 & A_1 \leq 3 \\ y_1 \leq B_1 & B_1 \leq 3 \end{array} \right. \qquad (2.3.7)$$

$$f_2 = f_1 + x_2^2 + y_2^2 \left\{ \begin{array}{ll} x_1 + x_2 \leq A_2 & A_2 \leq 3 \\ y_1 + y_2 \leq B_2 & B_2 \leq 3 \end{array} \right. \qquad (2.3.8)$$

$$f_3 = f_2 + x_3^2 + y_3^2 \left\{ \begin{array}{l} x_1 + x_2 + x_3 = 3 \\ y_1 + y_2 + y_3 = 3 \end{array} \right. \qquad (2.3.9)$$

Next, f_1 is evaluated for all allowed values of x_1 and y_1. The results are shown in the following table.

x_1 \ y_1	0	1	2	3
0	0	1	4	9
1	1	2	5	10
2	4	5	8	13
3	9	10	13	18

The second stage must not be evaluated for $x_1 + x_2 = A_2$ where $A_2 = 0, 1, 2, 3$ subject to all the possible values for $y_1 + y_2$. The following table shows the values of f_2 for the second stage.

			$x_1 + x_2 = 0$ ($A_2 = 0$)	$x_1 + x_2 = 1$ ($A_2 = 1$)	
			$x_1 = 0$, $x_2 = 0$	$x_1 = 0$, $x_2 = 1$	$x_1 = 1$, $x_2 = 0$
$y_1 + y_2 = B_2$	y_1	y_2	f_2	f_2	f_2
0	0	0	⓪	①	①
1	0	1	①	①	2
	1	0	◌1	3	2
2	0	2	4	5	5
	1	1	②	③	③
	2	0	4	5	5
3	0	3	9	10	10
	1	2	⑤	⑥	⑥
	2	1	⑤	⑥	⑥
	3	0	9	10	10

○ indicates optimum value for stage

◌ indicates redundant optimal values for different policies at the same stage

			$x_1 + x_2 = 2$ $(A_2 = 2)$			$x_1 + x_2 = 3$ $(A_2 = 3)$			
			$x_1=0$, $x_2=2$	$x_1=1$, $x_2=1$	$x_1=2$, $x_2=0$	$x_1=0$, $x_2=3$	$x_1=1$, $x_2=2$	$x_1=2$, $x_2=1$	$x_1=3$, $x_2=0$
$y_1 + y_2 = B_2$	y_1	y_2	f_2	f_2	f_2	f_2	f_2	f_2	f_2
0	0	0	4	②	4	9	⑤	⑤	9
1	0	1	5	③	5	10	⑥	⑥	10
	1	0	5	③	5	10	⑥	⑥	10
2	0	2	8	6	8	13	9	9	13
	1	1	6	④*	6	11	⑦	⑦	11
	2	0	8	6	8	13	9	9	13
3	0	3	13	11	13	18	14	14	18
	1	2	9	⑦	9	14	⑩	⑩	14
	2	1	9	⑦	9	14	⑩	⑩	14
	3	0	13	11	13	18	14	14	18

* optimal value for problem

So far, the principle of optimality has not been employed. This principle is introduced in the evaluation of the third stage since the optimal values from the second stage must be used. These values are determined by finding the minimum values of f_2 within a particular A_2 classification for a particular B_2. In other words, the use of the optimal value theorem for the third stage requires the knowledge of the optimal value of f_2 for various values of $x_1 + x_2$ as in previous problems. This information must be known for various values of $y_1 + y_2$ because the process is attempting to maximize over two variables. The number of cases that must be examined for the third stage is relatively small since it is no longer required to investigate $A < 3$ and $B < 3$. Instead, only cases for $A = 3$ and $B = 3$ must be considered. The computation results for the third stage are shown below.

	$x_1 + x_2$	0	1	2	3
$y_1 + y_2$	x_3 \ y_3	3	2	1	0
0	3	18	14	12	14
1	2	14	9	8	10
2	1	12	8	6	8
3	0	14	10	8	10

The optimal combination of the x_i's and y_i's is now determined. From the previous table, it is seen that the optimal policy for the third decision is $y_3 = 1$ and $x_3 = 1$ and an optimal value function of 6 results for the entire process. This selection restricts the choice of x_1, x_2, y_1 and y_2 to the cases where $y_1 + y_2 = 2$ and $x_1 + x_2 = 2$ and focuses attention on nine numbers which satisfy these constraints. The optimal value of these numbers has already been selected; it is 4 and is marked with an asterisk. The corresponding values for x_1, x_2, y_1 and y_2 are

$$y_1 = 1$$
$$y_2 = 1$$
$$x_1 = 1$$
$$x_2 = 1$$

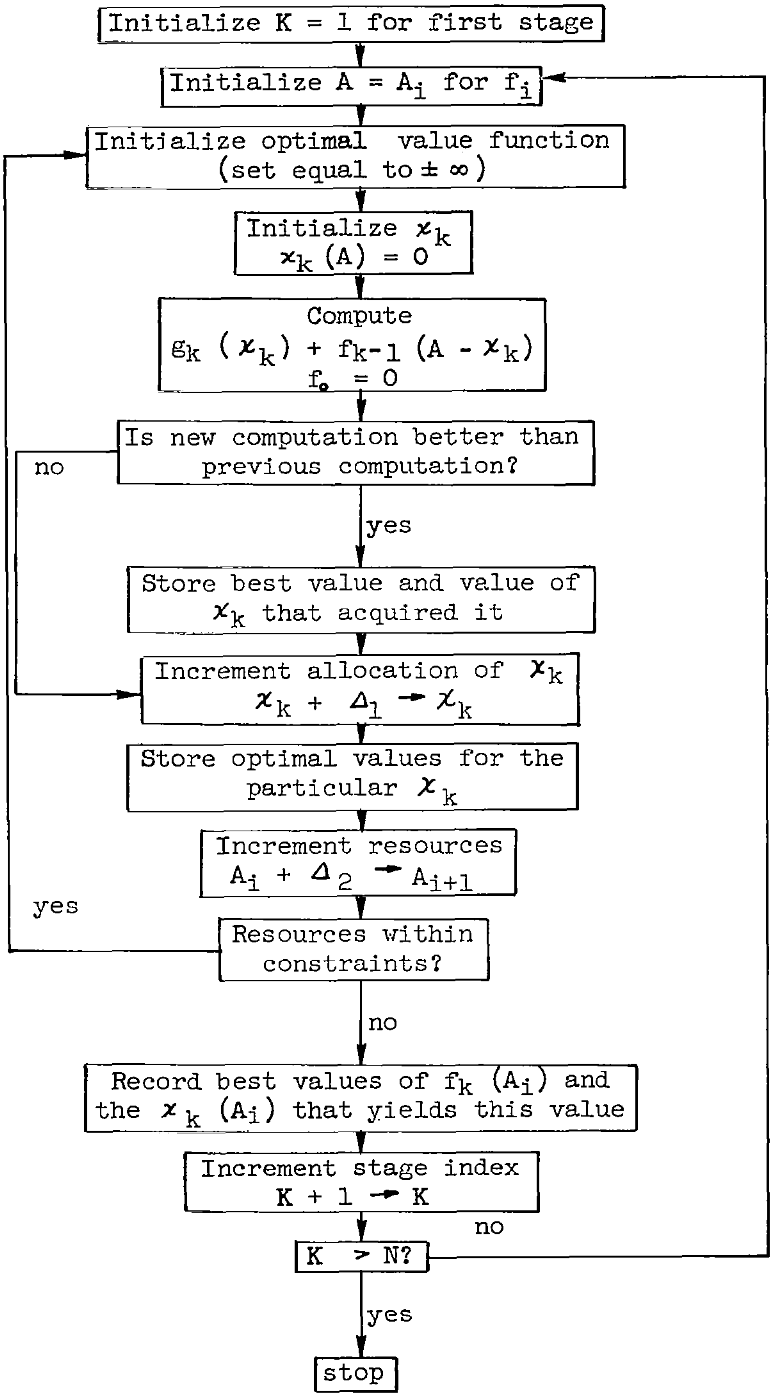
The total solution, including the optimal value of the final result, is now known. It is comforting to know that this result agrees with answers obtained by the use of Lagrange multipliers and intuitive results.

The same problem will now be solved using the method of approximation in policy space. This method starts by assuming a solution for the policy function (y_i). The next step then uses the conventional techniques of Dynamic Programming to find the sequence of (x_i) that minimizes f, assuming the previously mentioned y_i's. The techniques of Dynamic Programming are again employed, now using the sequence (x_i) and finding the sequence (y_i)

that minimizes f. This interchange of the roles of x_i and y_i continues until the change in the value of f reaches some predetermined value (just as a convergent series is truncated after higher order terms are no longer useful to the accuracy desired).

It is seen that the approximation in policy space method sacrifices computation time for storage requirements. This trade-off enables multi-dimensional problems to be solved even though their core storage requirements far exceed current memory capabilities when the straight forward Dynamic Programming approach is used. Hence, the increase in the computation time is a small price to pay the difference between a solution and insoluble problem.

Another method of overcoming the core storage requirements of the computer is to take advantage of the one stage nature of the solution by the use of blocks of logic and thus avoid storing any unnecessary data. This is done by constructing a logical flow chart that is used repetitively by incrementing index numbers for subsequent stages. Also, during the search procedure of the optimal value for a particular state, many unnecessary numbers can be immediately omitted by performing a comparison as soon as the number is computed. If it is the best value so far, it is retained. If it is not the best so far, it is immediately omitted. Thus many core locations can be saved as opposed to a maximum search over a section of the core memory. Still, it must be remembered that two pieces of information must be retained for each decision point. They are the optimal value at that point and the optimal decision at that point. The following sketch shows how a typical allocation problem would be formulated by using a flow chart and an immediate search procedure in order to conserve storage requirements. (Illustration on following page.)


Initialize K = 1 for first stage
Initialize A = A_i for f_i
Initialize optimal value function (set equal to $\pm \infty$)
Initialize x_k
$x_k(A) = 0$
Compute
$g_k(x_k) + f_{k-1}(A - x_k)$
$f_0 = 0$
Is new computation better than previous computation?
no
yes
Store best value and value of x_k that acquired it
Increment allocation of x_k
$x_k + \Delta_1 \rightarrow x_k$
Store optimal values for the particular x_k
Increment resources
$A_i + \Delta_2 \rightarrow A_{i+1}$
yes
Resources within constraints?
no
Record best values of $f_k(A_i)$ and the $x_k(A_i)$ that yields this value
Increment stage index
$K + 1 \rightarrow K$
no
$K > N$?
yes
stop

2.3.3.2 Stability and Sensitivity

It was previously noted that the Dynamic Programming approach solves a family of problems rather than a specific problem. Although this may appear to be wasteful at first, a closer evaluation would point out some definite advantages of this type of solution. The construction of mathematical models to represent physical phenomenon frequently involve approximations and uncertainties and hence the parameters of the models are not exactly known. It is, therefore, desirable to conduct studies for a variety of parameter values in order to determine the sensitivity of the results to these parameter changes. The uncertainties of the solution can then be evaluated. These solutions are in effect families of solutions and are obtained from Dynamic Programming applications, in many cases without extra effort beyond that required for a specific problem.

A precautionary note on the approximation of functions is in order at this point because of stability considerations. A very popular technique in many analyses involves the approximations of discrete functions by continuous functions or vice versa depending on the demands of the analytical tools being used. In many cases, such approximations are perfectly valid and the results are acceptable. In other cases, care must be taken to insure that the small differences between the actual function and its approximation do not introduce unacceptable variations in the solution. In general there are no stability theories available for Dynamic Programming and one must experiment with a particular problem to determine its peculiarities.

2.4 LIMITING PROCESS IN DYNAMIC PROGRAMMING

The previous sections have delt exclusively with the computational aspects of Dynamic Programming and have shown how the Principle of Optimality can be used to systematize the search procedure for finding an optimal decision sequence. As mentioned in Section 2.1, Dynamic Programming is also a valuable theoretical tool in that it can be used to develop additional properties of the optimal decision sequence. For example, it is well known that the optimal solution for the problem of Lagrange (Section 2.2.2) must satisfy the Euler-Lagrange equation. This differential equation, as well as other conditions resulting from either an application of the classical Calculus of Variations or the Pontryagin Maximum Principle, can also be developed through Dynamic Programming.

To develop these additional properties, the multi-stage decision process must be considered in the limit as the separation between neighboring states and decisions go to zero (i.e., as the process becomes continuous). That is, the problem is first discretized and a finite number of states and decisions considered just as in the computational approach of the previous sections. The Principle of Optimality is then used to develop a recursive equation by which the numerical values of the optimal decision sequence are computed. (This equation was not given an explicit statement in the previous sections since it was reasonably obvious there how the Principle of Optimality was to be used in the search process.) By considering the discretized process in the limit (i.e., allowing it to become a continuous process again), the recursive equation which governs the search procedure in the discrete case becomes a first-order, partial differential equation. From this partial differential equation, many additional properties of the optimal decision sequence can be developed.

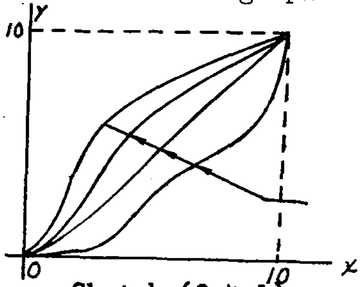
It should be mentioned that in some cases the limiting process outlined does not exist and the passage to the limit leads to an erroneous result. While this situation does occur in physically meaningful problems and, therefore, cannot be classed as pathological, it occurs infrequently enough as to cause little concern. Some examples of this phenomenon will be given later on.

2.4.1 Recursive Equation for the Problem of Lagrange

Consider the one-dimensional Lagrange problem of minimizing the integral

$$J = \int_{x_0, y_0}^{x_f, y_f} f(x, y, y')\, dx \qquad (2.4.1)$$

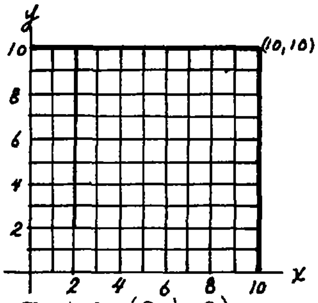
Let the lower limit of integration (x_0, y_0) be the point $(0, 0)$ and the upper limit be the point $(10, 10)$. As illustrated in the graph to the right, the problem consists of selecting from all curves $y(x)$ connecting the points $(0,0)$ and $(10, 10)$, that one curve for which the integral J in Eq. (2.4.1) is a minimum.



Sketch (2.4.1)

Consider the discrete form of the problem with the integral in Eq. (2.4.1) replaced by the summation

$$J = \sum_{i=0}^{9} f(x_i, y(x_i), y'(x_i))\Delta x$$



Sketch (2.4.2)

and with the grid selected, as shown to the right, so that $\Delta x = 1$. Now, let $R(x_0, y_0) = R(0,0)$ denote the minimum value of the summation in (2.4.2); that is

$$R(x_0, y_0) = R(0, 0) = \mathit{Min} \sum_{i=0}^{9} f(x_i, y(x_i), y'(x_i)\Delta x \qquad (2.4.2)$$

The arguments of R, x_0 and y_0 indicate the starting point of the curve which is the point $(0,0)$. Since the minimization is achieved by selecting the appropriate slope y' at each state of the process (that is, the decision variable is the slope y'), Eq. (2.4.2) is sometimes written

$$R(x_0, y_0) = R(0, 0) = \underset{y'}{\mathit{Min}} \sum_{i=0}^{9} f(x_i, y(x_i), y'(x_i))\Delta x \qquad (2.4.3)$$

Let y_0' denote the slope on the section of the curve from x_0 to $x_1 \{x_1 = x_0 + \Delta x\}$. Then $y_1 = y(x_1)$ is given by

$$y(x_1) = y_1 = y_0 + y_0' \Delta x \qquad (2.4.4)$$

with

$$x_1 = x_0 + \Delta x \qquad (2.4.5)$$

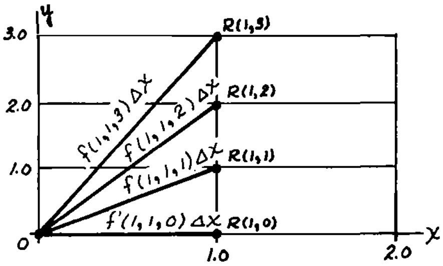
Using this notation, Eq. (2.4.3) can be rewritten as

$$R(x_0, y_0) = \underset{y'}{MIN}\left\{f(x_0, y_0, y_0')\Delta x + \sum_{i=1}^{9} f(x_i, y(x_i), y'(x_i))\Delta x\right\} \quad (2.4.6)$$

Let $R(x_1, y_1)$ be the minimum value of the summation $\sum_{i=1}^{9} f(x_i, y(x_i), y'(x_i))\Delta x$ where the arguments x_1 and y_1, again denote the starting point of the curve. Thus

$$R(x_1, y_1) = \underset{y'}{MIN}\sum_{i=1}^{9} f(x_i, y(x_i), y'(x_i))\Delta x \cong \underset{y'}{MIN}\int_{x_1, y_1}^{x_f, y_f} f(x, y, y')dx . \quad (2.4.7)$$

Note from the grid size in Sketch (2.4.2) and Eqs. (2.4.4) to (2.4.5) that $x_1 = 1$ $\{x_1 = x_0 + \Delta x = 0+1\}$ but that y_1 can take any value from 0 to 10. Suppose the optimal curve connecting the points (x_1, y_1) and (x_f, y_f) has been calculated and the function $R(x_1, y_1)$ evaluated for $x_1 = 1$ and $y_1 = 0, 1, 2, \ldots, 10$. Then, using the Principle of Optimality and the grid (which is partially shown to the right) allows the optimal solution to the original problem (namely the value of $R(x_0, y_0)$ to be located.



Sketch (2.4.3)

Again letting

$$y_1 = y_0 + y_0' \Delta x$$

and

$$R(x_1, y_1) = R(x_1, y_0 + y_0' \Delta x)$$

it follows that $R(x_0, y_0)$ is given by

$$R(x_0, y_0) = \underset{y_0'}{MIN}\left\{f(x_0, y_0, y_0')\Delta x + R(x_1, y_0 + y_0'\Delta x)\right\} \quad (2.4.8)$$

That is, the slope y_0' at the point (x_0, y_0) would be selected so that the sum of the two terms $f(x_0, y_0, y_0')\Delta x + R(x_1, y_1)$ is a minimum where $y_1 = y_0 + y_0'\Delta x$. This is exactly the computational procedure which was followed in the example problems of the preceding sections.

Equation (2.4.8) can be developed directly from Eq. (2.4.6) by noting that the operation $\underset{y'}{MIN}$ means the minimization is to be performed over all slopes y_i' with i running from 0 to 9. Thus,

$$\underset{y'}{MIN} = \underset{y_i' \ (i=0,9)}{MIN}$$

Now, substituting this expression into (2.4.6) provides

$$R(x_0, y_0) = \underset{y_i' \ (i=0,9)}{MIN}\left\{f(x_0, y_0, y_0')\Delta x + \sum_{i=1}^{9} f(x_i, y(x_i), y'(x_i))\Delta x\right\} \quad (2.4.9)$$

$$= \underset{y_0'}{MIN}\left\{\underset{y_i' \ (i=1,9)}{MIN}\left\{f(x_0, y_0, y_0')\Delta x + \sum_{i=1}^{9} f(x_i, y(x_i), y'(x_i))\Delta x\right\}\right\}$$

But

$$\underset{y_i' \; (i=1,9)}{MIN} f(x_0, y_0, y_0')\Delta x = f(x_0, y_0, y_0')\Delta x \qquad (2.4.10)$$

since the function on which the MIN operator is operating is not depended on $y_i'(i = 1, 9)$. Also, from the definition of $R(x_1, y_1)$ given in (2.4.7)

$$R(x_1, y_1) = \underset{y_i' \; (i=1,9)}{MIN} \sum_{i=1}^{9} f(x_i, y(x_i), y'(x_i))\Delta x \qquad (2.4.11)$$

where $$y_1 = y_0 + y_0'\Delta x$$

Thus, substituting (2.4.10) and 2.4.11) into (2.4.9) provides the desired result

$$R(x_0, y_0) = \underset{y_0'}{MIN}\left\{ f(x_0, y_0, y_0')\Delta x + R(x_1, y_0 + y_0'\Delta x)\right\} \qquad (2.4.12)$$

Again, it is to be emphasized that this equation is simply the mathematical statement for the search procedure as suggested by the Principle of Optimality.

To develop the solution of the problem using Eq. (2.4.12), the values of $R(x_1, y_1)\{y_1 = y_0 + y_0'\Delta x\}$ must be calculated. However, these quantities can be calculated in precisely the same manner as $R(x_0, y_0)$; that is, $R(x_1, y_1)$ must, according to the Principle of Optimality, satisfy a recursive equation of the form

$$R(x_1, y_1) = \underset{y_1'}{MIN}\left\{ f(x_1, y_1, y_1') + R(x_2, y_1 + y'\Delta x)\right\} \qquad (2.4.13)$$

and similarly, for all points (x_i, y_i) in the grid,

$$R(x_i, y_i) = \underset{y_i'}{MIN}\left\{ f(x_i, y_i, y_i') + R(x_{i+1}, y_i + y_i'\Delta x)\right\} \qquad (2.4.14)$$

Thus Eq. (2.4.14) represents a computation algorithm for finding the optimal decision sequence. Note that all curves must terminate at the point (x_f, y_f), the upper limit of integration, which for the particular problem here is the point $(10, 10)$. This condition can be expressed mathematically as

$$R(x_{10}, y_{10}) = R(x_f, y_f) = \underset{y'}{MIN}\int_{x_f, y_f}^{x_f, y_f} f(x, y, y')dx = 0 \qquad (2.4.15)$$

2.4.2 Recursive Equation in Limiting Form

In this section the recursive equation of (2.4.14) will be considered in the limit as $\Delta x \rightarrow 0$. It will be shown that under certain relatively unrestrictive assumptions, the limiting form of this equation becomes a first-order, partial differential equation.

Again, the problem under consideration is that of minimizing the integral

$$J = \int_{x_0,\, y_0}^{x_f,\, y_f} f(x, y, y')\,dx \qquad (2.4.16)$$

as in the preceding section, let $R(x_0,y_0)$ denote the minimum value of this integral. Thus

$$R(x_0, y_0) = \mathrm{Min} \int_{x_0,\, y_0}^{x_f,\, y_f} f(x, y, y')\,dx$$

or alternately

$$R(x_0, y_0) = \underset{y'}{\mathrm{Min}} \int_{x_0,\, y_0}^{x_f,\, y_f} f(x, y, y')\,dx \qquad (2.4.17)$$

where the y' under the Min symbol denotes the value of the decision variable y' on the interval $(x_0 \le x \le x_f)$; that is

$$y' = \{\, y'(\xi)\ ;\quad x_0 \le \xi \le x_f \,\} \qquad (2.4.18)$$

Now, note that $R(x_0, y_0)$ is simply a number; namely, the minimum value of integral J, with the arguments (x_0, y_0) denoting the point at which the integration begins.

Proceeding as in the discrete case, let $R(x, y)$ denote the minimum value of the same integral but with the integration beginning at the point (x, y) : that is

$$R(x, y) = \underset{y'}{\mathrm{Min}} \int_{x,\, y}^{x_f,\, y_f} f(x, y, y')\,dx \ ;\ y' = \{\, y'(\xi)\, ;\, x \le \xi \le x_f \,\} \qquad (2.4.19)$$

Again, R is simply a number but a number which changes in value as the initial point of integration (the argument of R) changes. Now

$$\int_{x,\,y}^{x_f,\, y_f} f(x, y, y')\,dx = \int_{x,\,y}^{x+\Delta x,\, y+\Delta y} f(x, y, y')\,dx + \int_{x+\Delta x,\, y+\Delta y}^{x_f,\, y_f} f(x, y, y')\,dx \qquad (2.4.20)$$

$$= f(x, y, y')\Delta x + O(\Delta x) + \int_{x+\Delta x,\, y+\Delta y}^{x_f,\, y_f} f(x, y, y')\,dx$$

where $O(\Delta x)$ denotes terms of order Δx ; that is,

$$\lim_{\Delta x \to 0} \frac{O(\Delta x)}{\Delta x} = O$$

Substituting (2.4.20) into (2.4.19) provides

$$R(x, y) = \underset{y'}{\mathrm{Min}} \left\{ f(x, y, y')\Delta x + O(\Delta x) + \int_{x+\Delta x,\, y+\Delta y}^{x_f,\, y_f} f(x, y, y')\,dx \right\}$$

$$= \underset{\substack{y'(\xi_1) \\ (x \le \xi_1 \le x+\Delta x)}}{\mathrm{Min}} \ \underset{\substack{y'(\xi_2) \\ (x+\Delta x \le \xi_2 \le x_f)}}{\mathrm{Min}} \left\{ f(x, y, y')\Delta x + O(\Delta x) + \int_{x+\Delta x,\, y+\Delta y}^{x_f,\, y_f} f(x, y, y')\,dx \right\}$$

Now, noting that the second MIN operator on the right does not operate on $f(x,y,y')\Delta x$, it follows that

$$R(x,y) = \underset{\substack{y' \\ (x \le \xi_1 \le x+\Delta x)}}{MIN} \left\{ f(x,y,y')\Delta x + O(\Delta x) + R(x+\Delta x, y+\Delta y) \right\} \qquad (2.4.21)$$

This equation is essentially the same as the recursive relationship of Eq. (2.4.14). However, it can be reduced to a simpler form under the assumption that the second derivatives of $R(x,y)$ are bounded; that is

$$\left|\frac{\partial^2 R}{\partial x^2}\right| < \infty$$

$$\left|\frac{\partial^2 R}{\partial y^2}\right| < \infty$$

$$\left|\frac{\partial^2 R}{\partial x\, \partial y}\right| < \infty \qquad (2.4.22)$$

This assumption allows for the expansion

$$\begin{aligned} R(x+\Delta x, y+\Delta y) &= R(x,y) + \frac{\partial R}{\partial x}\Delta x + \frac{\partial R}{\partial y}\Delta y + O(\Delta x) \\ &= R(x,y) + \frac{\partial R}{\partial x}\Delta x + \frac{\partial R}{\partial y} y'\Delta x + O(\Delta x) \qquad (2.4.23) \end{aligned}$$

since $\Delta y = y'\Delta x$. Substituting (2.4.23) into (2.4.21), yields

$$R(x,y) = \underset{\substack{y'(\xi_1) \\ (x \le \xi \le x+\Delta x)}}{MIN} \left\{ f(x,y,y')\Delta x + R(x,y) + \frac{\partial R}{\partial x}\Delta x + \frac{\partial R}{\partial y} y'\Delta x + O(\Delta x) \right\}$$

Noting that the MIN operator does not operate on $R(x,y)$ and factoring out Δx, this expression becomes

$$\underset{\substack{y(\xi_1) \\ (x \le \xi \le x+\Delta x)}}{MIN} \left\{ f(x,y,y') + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y} y' + \frac{O(\Delta x)}{\Delta x} \right\} = 0$$

Finally, taking the limit as $\Delta x \to 0$ provides the desired result

$$\underset{y'(x)}{MIN} \left\{ f(x,y,y') + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y} y' \right\} = 0 \qquad (2.4.24)$$

Equation (2.4.24) is the continuous analog of the recursive computational algorithm of Eq. (2.4.14). Since it is a first-order (non-classical) partial differential equation, one boundary condition must be specified. This boundary condition is the same as that which was applied in the discrete case; namely,

$$R(x_f, y_f) = \underset{y'}{MIN} \int_{x_f, y_f}^{x_f, y_f} f(x,y,y')dx = 0 \qquad (2.4.25)$$

The combined solution of (2.4.24) and (2.4.25) yields $R(x,y)$ which is the minimum value of the integral starting at the point (x,y). Evaluating R at the point (x_0,y_0) provides the solution to the problem.

Two questions arise at this point. First, how are Eqs. (2.2.24) and (2.4.25) solved; and secondly, once the function $R(x,y)$ is known, how is the optimal curve $y(x)$ determined? Both questions are interrelated and can be answered by putting the partial differential equation in (2.4.24) in a more usable form.

Note that the minimization in Eq. (2.4.24) is a problem in maxima - minima theory: that is, the slope $y'(x)$ is to be selected so that the quantity $f(x,y,y')+\frac{\partial R}{\partial x}+\frac{\partial R}{\partial y}y'$ is a minimum. Assuming that f is differentiable and noting that R does not depend on y', it follows that

$$\frac{\partial}{\partial y'}\left\{f(x,y,y')+\frac{\partial R}{\partial x}+\frac{\partial R}{\partial y}y'\right\}=0$$

or

$$\frac{\partial f}{\partial y'}+\frac{\partial R}{\partial y}=0 \qquad (2.4.26)$$

Thus, Eq. (2.4.24) is equivalent to the two equations

$$f(x,y,y')+\frac{\partial R}{\partial x}+\frac{\partial R}{\partial y}y'=0$$

$$\frac{\partial f}{\partial y'}+\frac{\partial R}{\partial y}=0 \qquad (2.4.27)$$

which, when combined, lead to a classical partial differential equation in the independent variables x and y $\{y'$ is eliminated by Eq. (2.4.26)$\}$ and the dependent variable $R(x,y)$. This equation can be solved either analytically or numerically, and then Eq. (2.4.26) used to determine the optimal decision sequence $y'(x)$ for $(x_0 \le x \le x_f)$.

2.4.3 An Example Problem

The problem of minimizing the integral

$$J=\int_{x_0,y_0}^{x_f,y_f} f(x,y,y')\,dx$$

has been shown to be equivalent to solving the partial differential equations

$$f(x,y,y')+\frac{\partial R}{\partial x}+\frac{\partial R}{\partial y}y'=0 \qquad (2.4.28)$$

where y' is determined from

$$\frac{\partial f}{\partial y'} + \frac{\partial R}{\partial y} = 0 \tag{2.4.29}$$

and with the boundary condition.

$$R(x_f, y_f) = 0 \tag{2.4.30}$$

To illustrate this equivalence, consider the shortest distance problem treated in Section 2.2.2.1 where

$$f(x, y, y') = \sqrt{1+y'^2} \tag{2.4.31}$$

In this case, Eqs. (2.4.28) and (2.4.29) become

$$\sqrt{1+y'^2} + \frac{\partial R}{\partial y} y' + \frac{\partial R}{\partial x} = 0$$

$$\frac{y'}{\sqrt{1+y'^2}} + \frac{\partial R}{\partial y} = 0$$

By straight-forward substitution, it can be verified that the function satisfying these equations and the boundary condition of (2.4.30) is given by

$$R(x, y) = \sqrt{(x-x_f)^2 + (y-y_f)^2} \tag{2.4.32}$$

with

$$y' = \frac{y_f - y}{x_f - x} \tag{2.4.33}$$

At the initial point

$$y'(x_0) = \frac{y_f - y_0}{x_f - x_0} \tag{2.4.34}$$

while at the succeeding point along the solution,

$$x_1 = x_0 + \Delta x$$

$$y_1 = y_0 + y_0' \Delta x$$

$$y'(x) = \frac{y_f - y_1}{x_f - x_1} = \frac{y_f - y_0 - y_0'\Delta x}{x_f - x_0 - \Delta x} = y'(x_0)$$

In a similar fashion, it can be shown that for each succeeding point along the optimal solution emanating from (x_0, y_0)

$$y'(x) = y'(x_0)$$

thus, verifying that the solution is a straight line with slope given by Eq. (2.4.34).

2.4.4 Additional Properties of the Optimal Solution

The solution to the problem of minimizing the integral

$$J = \int_{x_0, y_0}^{x_f, y_f} f(x, y, y')\,dx \tag{2.4.35}$$

is usually developed by means of the Calculus of Variations with the development consisting of the establishment of certain necessary conditions which the optimal solution must satisfy. In this section, it will be shown that four of these necessary conditions resulting from an application of the Calculus of Variations can also be derived through Dynamic Programming.

In the previous sections it was shown that the function $R(x,y)$ defined by

$$R(x,y) = \underset{y'}{\mathrm{MIN}} \int_{x,y}^{x_f, y_f} f(x,y,y')\,dx \tag{2.4.36}$$

satisfies the partial differential equation

$$\underset{y'(x)}{\mathrm{MIN}} \left\{ f(x,y,y') + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y} y' \right\} = 0 \tag{2.4.37}$$

Setting the first derivative with respect to y' to zero in this equation provides the additional condition

$$\frac{\partial f}{\partial y'} + \frac{\partial R}{\partial y} = 0 \tag{2.4.38}$$

Also, if y' is to minimize the bracketed quantity in (2.4.37), then the second derivative of this quantity with respect to y' must be greater than or equal to zero. Hence the condition,

$$\frac{\partial^2 f}{\partial y'^2} \geq 0 \tag{2.4.39}$$

must be satisfied along the optimal solution. This condition is referred to as the Legendre condition in the Calculus of Variations.

A slightly stronger condition than that in (2.4.39) can be developed by letting y'_{opt} denote the optimal solution and Y' denote any other solution. Then from (2.4.37)

$$f(x, y, y'_{opt}) + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial x} y'_{opt} = 0$$

$$f(x, y, Y') + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y} Y' \geq 0$$

Now, combining these two expressions provides

$$f(x,y,Y') - f(x,y,y'_{opt}) + \left\{ Y' - y_{opt} \right\} \frac{\partial R}{\partial y} \geq 0 \qquad (2.4.40)$$

But from Equation (2.4.38)

$$\frac{\partial R}{\partial y} = \frac{\partial R}{\partial y}(x,y) = -\frac{\partial f}{\partial y'}(x,y,y'_{opt})$$

Thus, substituting this equation into (2.4.40), yields the Weierstrass condition of the Calculus of Variations.

$$f(x,y,Y') - f(x,y,y'_{opt}) - (Y' - y_{opt}) \frac{\partial f}{\partial y'}(x,y,y'_{opt}) \geq 0 \qquad (2.4.41)$$

When the slope y' in (2.4.37) is computed according to the optimizing condition of (2.4.38) it follows that

$$f(x,y,y') + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y} y' = 0 \qquad (2.4.42)$$

Note that y' as developed from (2.4.38) will be a function of x and y. At points (x,y) for which $y'(x,y)$ is differentiable, (i.e., $\frac{\partial y'}{\partial x}$ and $\frac{\partial y'}{\partial y}$ exist). Eqs. (2.4.42) and 2.4.38) can be combined to yield a third necessary condition. Taking the total derivative of (2.4.38) with respect to x and the partial derivative of (2.4.42) with respect to y yields

$$\frac{d}{dx}\left(\frac{\partial f}{\partial y'}\right) + \frac{\partial^2 R}{\partial x \partial y} + \frac{\partial^2 R}{\partial y^2} y' = 0$$

$$\frac{\partial f}{\partial y} + \frac{\partial f}{\partial y'}\frac{\partial y'}{\partial y} + \frac{\partial^2 R}{\partial x \partial y} + \frac{\partial^2 R}{\partial y^2} y' + \frac{\partial R}{\partial y}\frac{\partial y'}{\partial y} = 0$$

Subtracting and making use of (2.4.38) provides

$$\frac{d}{dx}\left(\frac{\partial f}{\partial y'}\right) - \frac{\partial f}{\partial y} = 0 \qquad (2.4.43A)$$

which is the Euler-Lagrange equation; an equation which must be satisfied at points (x,y) where y' is differentiable. Across discontinuities in y', the required derivatives do not exist, and (2.4.43A) does not hold. However, at such points $R(x,y)$ is continuous and so is $\frac{\partial R}{\partial y}$ according to the original assumptions of (2.4.22). Thus, from Eq. (2.4.38) $\frac{\partial f}{\partial y'}$, is also continuous and the Weierstrass-Erdman corner condition

$$\frac{\partial f}{\partial y'}(x,y,y'^{(+)}) = \frac{\partial f}{\partial y'}(x,y,y'^{(-)}) \qquad (2.4.43B)$$

must hold.

Collecting the results of this section, the curve $y(x)$ which minimizes the integral

$$J = \int_{x_0, y_0}^{x_f, y_f} f(x, y, y')\,dx$$

must satisfy

(1) Euler-Lagrange Equation

$$\frac{d}{dx}\left(\frac{\partial f}{\partial y'}\right) - \frac{\partial f}{\partial y} = 0 \qquad (2.4.44A)$$

(2) Weierstrass-Erdman Corner Condition

$$\frac{\partial f}{\partial y'}(x, y, y'^{(+)}) = \frac{\partial f}{\partial y'}(x, y, y'^{(-)}) \qquad (2.4.44B)$$

(3) Weierstrass Condition

$$f(x, y, Y') - f(x, y, y') - (Y' - y')\frac{\partial f}{\partial y'}(x, y, y') \geq 0 \qquad (2.4.44C)$$

(4) Legendre Condition

$$\frac{\partial^2 f}{\partial y'^2}(x, y, y') \geq 0 \qquad (2.4.44D)$$

In addition to these four conditions, a fifth necessary condition, the classical Jacobi condition, can also be developed by means of Dynamic Programming. Since this condition is rather difficult to apply and frequently does not hold in optimal control problems, it will not be developed here. The interested reader should consult Reference (2.4.1), page 103.

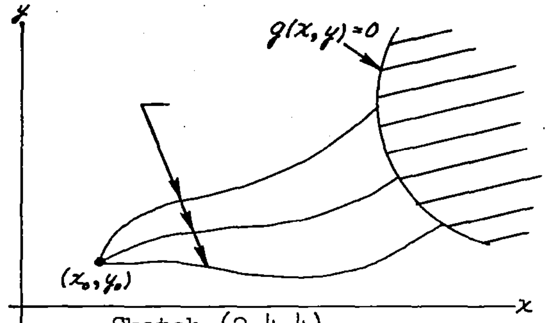
2.4.5 Lagrange Problem with Variable End Points

In the preceeding sections the problem of minimizing the integral

$$J = \int_{x_0, y_0}^{x_f, y_f} f(x, y, y')\,dx$$

was considered where the limits of integration, (x_0, y_0) and (x_f, y_f) were fi‸ed. In this section a minor variation on this problem will be considered in which the upper limit of integration is not fixed precisely, but is required to lie in the curve

$$g(x, y) = g(x_f, y_f) = 0 \qquad (2.4.45A)$$



Sketch (2.4.4)

The situation is pictured to the right. Note that the minimization of the integral involves both the selection of the optimal curve $y(x)$ and the optimal terminal point (x_f, y_f) along the curve $g(x_f, y_f)=0$. As in the fixed end point case, let

$$R(x,y) = \underset{y'}{Min} \int_{x,y}^{x_f, y_f} f(x,y,y')\,dx$$

where the terminal point (x_f, y_f) lies on the curve of Eq. (2.4.45A). Following the procedure of Section (2.4.2), it can again be shown that $R(x,y)$ satisfies the partial differential equation

$$\underset{y'(x)}{Min} \left\{ f(x,y,y') + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y} y' \right\} \qquad (2.4.45B)$$

However, the boundary condition on R is slightly different in this case.

Since $R(x,y)$ is the minimum value of the integral starting from the point (x,y) and terminating on the curve $g(x,y)=0$, it follows that $R(x,y)$ is zero for any (x,y) satisfying $g(x,y)=0$; that is, the value of the integral is zero since both limits of integration are the same. Hence, the boundary condition for Eq. (2.4.45B) is

$$R(x,y) = 0 \quad \text{on} \quad g(x,y)=0 \qquad (2.4.46)$$

This condition can be put in an alternate form if the equation $g(x,y)=0$ can be solved for $y \left\{ i.e.,\ \frac{\partial g}{\partial y} \neq 0 \right\}$. In this case

$$g(x,y) = 0 \Longleftrightarrow y - h(x) = 0 \qquad (2.4.47)$$

and Eq. (2.4.46) becomes

$$R(x,y) = 0 \qquad \text{on} \quad y = h(x) \qquad (2.4.48)$$

or

$$R(x, h(x)) = 0 \qquad (2.4.49)$$

Alternately, for two neighboring points along the curve $y = h(x)$, say x and x + δx .

$$R(x+\delta x; h(x+\delta x)) - R(x, h(x)) = 0 = R(x+\delta x, y+\delta y) - R(x; y)$$

where

$$\delta y = \frac{dh}{dx}\,\delta x$$

Thus, dividing by δx and taking the limit provides

$$\frac{\partial R}{\partial x} + \frac{\partial R}{\partial y}\frac{dy}{dx} = 0 = \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y}\frac{dh}{dx} \qquad (2.4.50)$$

But from (2.4.47)

$$\frac{dy}{dx} = \frac{dh}{dx} = \frac{-\frac{\partial g}{\partial x}}{\frac{\partial g}{\partial y}}$$

Hence, the boundary condition

$$\frac{\partial R}{\partial x} - \frac{\partial R}{\partial y}\left(\frac{\frac{\partial g}{\partial x}}{\frac{\partial g}{\partial y}}\right) = 0 \quad \text{ON} \quad g(x,y) = 0 \qquad (2.4.51)$$

results. Setting $\frac{\partial R}{\partial y} / \frac{\partial g}{\partial y} = \mu$ and substituting into (2.4.51) provides

$$\left.\begin{aligned} \frac{\partial R}{\partial x} - \mu \frac{\partial g}{\partial x} &= 0 \\ \frac{\partial R}{\partial y} - \mu \frac{\partial g}{\partial y} &= 0 \end{aligned}\right. \quad \text{ON} \quad g(x,y) = 0 \qquad (2.4.52)$$

which indicates that the gradient of $R(x,y)$ and the gradient of $g(x,y)$ are co-linear along the curve $g(x,y)=0$.

Eqs. (2.4.46), (2.4.49), (2.4.51) and (2.4.52) are different but equivalent representations of the boundary condition that the R function must satisfy when the terminal point is required to lie on the curve $g(x,y)=0$. From this boundary condition the transversality condition which the Calculus of Variations requires can be derived. This is shown next.

From Eq. (2.4.45B) it follows that the optimal slope must satisfy

$$\frac{\partial f}{\partial y'} + \frac{\partial R}{\partial y} = 0 \qquad (2.4.53)$$

at all points (x,y) including the terminal point. Using this equation, Eq. (2.4.45B) becomes

$$f(x,y,y') + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y} y' = 0 \qquad (2.4.54)$$

and must also hold at every point including the terminal point. Combining Eqs. (2.4.51), (2.4.53) and (2.4.54) provides

$$\left.\begin{aligned} f(x,y,y') - y' \frac{\partial f}{\partial y'} + \mu \frac{\partial g}{\partial x} &= 0 \\ \frac{\partial f}{\partial y'} + \mu \frac{\partial g}{\partial y} &= 0 \end{aligned}\right\} \quad \text{ON} \quad g(x,y) = 0 \qquad (2.4.55)$$

which is the transversality condition which the optimal solution must satisfy; that is, Eq. (2.4.55) specifies which of the points along $g(x,y)=0$ is the point for which the integral J is a minimum.

2.4.6 N-Dimensional Lagrange Problem

The concepts of Sections (2.4.1) to (2.4.5) which have been developed in connection with minimizing the integral

$$J = \int_{x_0, y_0}^{x_f, y_f} f(x, y, y')\,dx$$

where y is a scalar (1-dimensional) variable, can be extended to the case in which y is an n dimensional vector

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{2.4.56}$$

and the integral to be minimized is

$$J = \int_{x_0, y_0}^{x_f, y_f} f(x, y, y')\,dx = \int_{x_0, y_{1_0}, y_{2_0}, \cdots, y_{n_f}}^{x_f, y_{1_f}, y_{2_f}, \cdots y_{n_f}} f(x, y_1, y_2 \cdots y_n, y_1', y_2', \ldots, y_n')\,dx \tag{2.4.57}$$

Again, let $R(x,y)$ satisfy the equation

$$R(x, y) = R(x, y_1, y_2, \ldots, y_n) = \underset{y'}{Min} \int_{x, y}^{x_f, y_f} f(x, y, y')\,dx \tag{2.4.58}$$

Then, following a procedure identical to that employed in Eqs. (2.4.20) to (2.4.24) but with the scalar y replaced by the vector y as indicated in Eq. (2.4.56), it can be shown that $R(x,y)$ satisfies the equation

$$\underset{y'}{Min} \left\{ f(x, y_1, y_2, \ldots, y_n, y_1', y_2', \ldots, y_n') + \frac{\partial R}{\partial x} + \sum_{i=1}^{n} \frac{\partial R}{\partial y_i} y_i \right\} = 0 \tag{2.4.59}$$

or in the vector notation

$$\underset{y'}{Min} \left\{ f(x, y, y') + \frac{\partial R}{\partial x} + \left(\frac{\partial R}{\partial y}\right)^T y' \right\} = 0 \tag{2.4.60}$$

where superscript T denotes the transpose and $\frac{\partial R}{\partial y}$ and y' are the vectors

$$y' = \begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_n' \end{pmatrix} \quad ; \qquad \frac{\partial R}{\partial y} = \begin{pmatrix} \frac{\partial R}{\partial y_1} \\ \frac{\partial R}{\partial y_2} \\ \vdots \\ \frac{\partial R}{\partial y_n} \end{pmatrix} \tag{2.4.61}$$

The boundary condition to be satisfied by $R(x,y)$ will in all cases take the form

$$R(x_f, y_f) = 0 \qquad (2.4.62A)$$

whether the point (x_f, y_f) is fixed or allowed to vary on some surface in the (x, y) space. In the latter case, however, Eq. (2.4.62A) has several alternate representations similar to those developed for the 1-dimensional problem e.g. Eqs. (2.4.46) to (2.4.52) . For example, if the terminal point (x_f, y_f) is required to lie in the surface specified by the m constraint equations

$$G(x, y) = \begin{pmatrix} g_1(x, y) \\ g_2(x, y) \\ \vdots \\ g_m(x, y) \end{pmatrix} = 0 \; ; \; m \leq n+1 \qquad (2.4.62B)$$

the boundary condition of R as given in (2.4.62A) can also be written as

$$R(x, y) = 0 \quad \text{on} \quad G(x, y) = 0 \Rightarrow \{ g_i(x, y) = 0 \; ; \; i = 1, m \} \qquad (2.4.63A)$$

or analogous to Eq. (2.4.52), as

$$\left.\begin{aligned} \frac{\partial R}{\partial x} - \mu^T \left(\frac{\partial G}{\partial x}\right) &= 0 = \frac{\partial R}{\partial x} - \sum_{j=1}^{m} \mu_j \frac{\partial g_j}{\partial x} = 0 \\ \frac{\partial R}{\partial y_i} - \mu^T \frac{\partial G}{\partial x} &= 0 = \frac{\partial R}{\partial y_i} - \sum_{j=1}^{m} \mu_j \frac{\partial g_j}{\partial y_i} = 0 \end{aligned}\right\} \text{on } G(x,y) = \begin{pmatrix} g_1(x,y) \\ g_2(x,y) \\ \vdots \\ g_m(x,y) \end{pmatrix} = 0 \qquad (2.4.63B)$$

where μ is the m dimensional vector

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix} \qquad (2.4.64)$$

Thus, by combining (2.4.63B) with (2.4.60) and using the optimization condition inherent in (2.4.60), that

$$\frac{\partial f}{\partial y_i'} + \frac{\partial R}{\partial y_i} = 0 \quad ; \quad i = 1, n \qquad (2.4.65)$$

the transversality conditions of the Calculus of Variations, corresponding to the terminal constraints of Eq. (2.4.62B), can be developed. These conditions, which are essentially the m dimensional equivalent of the one-dimensional transversality condition of Eq. (2.4.55), take the form

$$\left.\begin{aligned} f - \sum_{i=1}^{n} y_i' \frac{\partial f}{\partial y_i'} + \sum_{j=1}^{m} \mu_j \frac{\partial g_j}{\partial x} &= 0 \\ \frac{\partial f}{\partial y_i'} + \sum_{j=1}^{m} \mu_j \frac{\partial g_j}{\partial y_i} &= 0 \end{aligned}\right\} \text{ on } \begin{pmatrix} g_1(x,y) \\ g_2(x,y) \\ \vdots \\ g_m(x,y) \end{pmatrix} = 0 \ . \qquad (2.4.66)$$

One final remark regarding the n-dimensional Lagrange problem is appropriate. In Section (2.4.4) it was shown that in the 1-dimensional problem, the partial differential equation governing the function R could be used to develop some of the necessary conditions usually developed by means of the Calculus of Variations. The same thing can be done in the n-dimensional case. The vector form of the necessary conditions in the case which corresponds to Eqs. (2.4.44A) to 2.4.44D), is as follows:

(1) Euler-Lagrange Equations

$$\frac{d}{dx}\left(\frac{\partial f}{\partial y_i'}\right) - \frac{\partial f}{\partial y_i} = 0 \quad ; \quad i = 1, n \qquad (2.4.67\text{A})$$

(2) Weierstrass-Erdman Corner Condition

$$\frac{\partial f}{\partial y_i}(x, y, y'^{(+)}) = \frac{\partial f}{\partial y_i}(x, y, y'^{(-)}) \quad ; \quad i = 1, n \qquad (2.4.67\text{B})$$

(3) Weierstrass Condition

$$f(x, y, Y') - f(x, y, y') - \sum_{i=1}^{n} (Y_i' - y_i') \frac{\partial f}{\partial y'}(x, y, y') \geq 0 \qquad (2.4.67\text{C})$$

where

$$y' = \begin{pmatrix} y_1' \\ y_2' \\ \vdots \\ y_n' \end{pmatrix} \qquad\qquad Y = \begin{pmatrix} Y_1' \\ Y_2' \\ \vdots \\ Y_n' \end{pmatrix}$$

(4) Legendre-Condition

$$\left(\frac{\partial^2 f}{\partial y'^2}\right) = \begin{pmatrix} \dfrac{\partial^2 f}{\partial y_1'^2}, & \dfrac{\partial^2 f}{\partial y_1' \partial y_2'}, & \ldots, & \dfrac{\partial^2 f}{\partial y_1' \partial y_n'} \\ \vdots & & & \\ \dfrac{\partial^2 f}{\partial y_n' \partial y_1'}, & \dfrac{\partial^2 f}{\partial y_n' \partial y_2'}, & \ldots, & \dfrac{\partial^2 f}{\partial y_n'^2} \end{pmatrix} \geq 0 \qquad (2.4.67\text{D})$$

That is, the matrix $\frac{\partial^2 f}{\partial y'^2}$ must be positive semi-definite.

2.4.7. Discussion of the Problem of Lagrange

In the preceeding five sections, it has been shown that the computational algorithm inherent in the Principle of Optimality is, under certain relatively unrestrictive assumptions [see Eq. (2.4.22], equivalent to a first-order, partial differential equation. This partial equation goes by a variety of names, one of which is the "Bellman" equation. The solution to the original problem of minimizing an integral is easily generated once the solution to the Bellman equation is known. It is to be emphasized that the source of this equation is the computational algorithm, that is, the equation is simply the limiting statement for how the computation is to be carried out.

It is a relatively rare case in which the Bellman equation can be solved in closed form, and the optimal solution to the problem developed analytically. In most cases, however, numerical procedures must be employed. The first of two available procedures consists of discretizing the problem and representing the partial differential equation as a set of recursive algebraic equations. This approach is just the reverse of the limiting procedure carried out in Section (2.4.2) where the recursive equation (2.4.15) was shown to be equivalent to the partial equation of (2.4.24). Hence, in this technique the continuous equation (2.4.24) is approximated by the discrete set in (2.4.15) and the solution to (2.4.15) is generated by using the same search techniques that were used in the sample problems of Section (2.2) and (2.3). Thus, the computational process implicit in Dynamic Programming is simply a method for solving a first-order, partial differential equation.

A second procedure for generating a numerical solution for the Bellman equation consists of integrating a set of ordinary differential equations which corresponds to the characteristic directions associated with the partial differential equation. For example, the solution to the partial equation

$$\frac{\partial S}{\partial x} g_1(x,y) + \frac{\partial S}{\partial y} g_2(x,y) = 0 \qquad (2.4.68)$$

for $S(x,y)$ with the boundary condition

$$S(x_0, y_0) = c(x_0, y_0) \quad \text{on} \quad h(x_0, y_0) = 0 \qquad (2.4.69)$$

involves the introduction of the variable t where

$$\frac{dx}{dt} = g_1(x, y)$$

$$\frac{dy}{dt} = g_2(x, y) \qquad (2.4.70)$$

Substituting (2.4.70) into (2.4.68) provides

$$\frac{\partial s}{\partial x}\frac{dx}{dt} + \frac{\partial s}{\partial y}\frac{dy}{dt} = \frac{ds}{dt} = 0 \qquad (2.4.71)$$

Hence, along the characteristic directions in Eq. (2.4.70), which emanate from points (x_o, y_o) satisfying $h(x_o, y_o) = 0$, $S(x, y) = C(x_o, y_o)$. This fact is derived from Eq. (2.4.69) and (2.4.71). Therefore, integration of Eqs. (2.4.70) for all (x_o, y_o) for which $h(x_o, y_o) = 0$ yields the solutions $S(x, y)$ to Eq. (2.4.68). If, in addition, x is monotonic and $g_1(x, y) \neq 0$, the characteristic direction in (2.4.70) can be represented more simply by

$$\frac{dy}{dx} = \frac{g_2(x, y)}{g_1(x, y)} \qquad (2.4.72)$$

A similar procedure to that outlined in the preceding paragraph can be used to solve the Bellman Equation, which for the 1-dimensional Lagrange problem is equivalent to the two equations

$$f(x, y, y') + \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y} y' = 0 \qquad (2.4.73)$$

$$\frac{\partial f}{\partial y'} + \frac{\partial R}{\partial y} = 0 \qquad (2.4.74)$$

The characteristics for this set of nonlinear equations are somewhat more difficult to develop than those for the linear example in Eq. (2.4.68). However, by referring to any standard text on partial differential equations see for example, Ref. (2.4.2), pages 61 to 66 it can be shown that the characteristics associated with Eqs. (2.4.73) and (2.4.74) are

$$\frac{dy}{dx} = y' \tag{2.4.75}$$

$$\frac{dR}{dx} = f(x, y, y') \tag{2.4.76}$$

$$\frac{d}{dx}\left(\frac{\partial R}{\partial x}\right) = - \frac{\partial f}{\partial x} \tag{2.4.77}$$

$$\frac{d}{dx}\left(\frac{\partial R}{\partial y}\right) = - \frac{\partial f}{\partial y} \tag{2.4.78}$$

The meaning of the first two equations is obvious. They are simply a restatement of the definitions of y' and $R(x,y)$. The last equation, when coupled with Eq. (2.4.74), reduced to the Euler-Lagrange equation

$$\frac{d}{dx}\left(\frac{\partial f}{\partial y'}\right) - \frac{\partial f}{\partial y} = 0 \tag{2.4.79}$$

Equation (2.4.77) is also equivalent to the Euler-Lagrange equation. This equivalence can be shown by differentiating (2.4.73) with respect to x and using (2.4.74). Thus, the characteristic directions associated with the Bellman equation are determined by solving the Euler-Lagrange equation. Since the value of R at the point (x_o, y_o) and the associated curve y (x) (i.e., the curve emanting from the point (x_o, y_o)) is of primary interest, it is only necessary to solve for one characteristic; namely, that one starting at (x_o, y_o). Thus, the solution to the problem of minimizing the integral

$$J = \int_{x_o, y_o}^{x_f, y_f} f(x, y, y')dx \tag{2.4.80}$$

can be achieved by integrating Eq. (2.4.79) to determine the optimum curve y(x), and then substituting this value back into (2.4.80) to evaluate J. This is the normal procedure and is followed in the Calculus of Variations. It should be mentioned that the solution to the Euler-Lagrange equation cannot be accomplished directly due to the two-point boundary nature of the problem (i.e., curve y(x) must connect the two points (x_o, y_o) and (x_f, y_f) while the determination of this curve by numerical integration of Eq. (2.4.79) requires a knowledge of the slope $y'(x_o)$. Hence, it may be more efficient to develop the solution by means of the first numerical technique of discretizing the problem and solving a set of recursive algebraic equations.

From this discussion, it is seen that the Bellman equation of Dynamic Programming and the Euler-Lagrange equation of the Calculus of Variations are equivalent approaches to the problem of Lagrange and that the equivalence exists on both the theoretical and computational levels. The other necessary conditions (e.g., Weierstrass, Legendre, etc.), generally enter the optimization problem in a less direct manner, in that once a solution has been developed, they serve to test if the solution is indeed minimizing. The fact that these conditions can be developed from the Bellman Equation lends a degree of completeness to the area of optimization theory.

2.4.8 The Problem of Bolza

The preceding sections have delt with the Dynamic Programming formulation of the problem of Lagrange. In this section the Bolza Problem will be considered, since optimal trajectory and control problems are usually cast in this form. The Bellman equation for this case will be developed and some solutions presented. Also, some comparisons and parallels will be drawn between the Dynamic Programming approach and the Pontryagin Maximum Principle (Ref. 2.4.3).

The problem of Bolza is usually stated in the following form: given the dynamical system

$$\dot{x}_i = f_i(x, u) \quad ; \quad i = 1, n \tag{2.4.81A}$$

or in the vector notation

$$\dot{x} = f(x, u) \tag{2.4.81B}$$

where the state x is a n-dimensional vector,

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{2.4.82}$$

and the control u is a r-dimensional vector,

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} \tag{2.4.83}$$

which is required to lie in some closed set U in the r-dimensional control space; determine the control history u(t) for which the functional

$$J = \int_{t_0}^{t_f} k(x, u)\,dt + \phi(x_f, t_f) = \text{minimum} \tag{2.4.84}$$

is minimized subject to the terminal constraints

$$\psi_y(x_f, t_f) = 0 \quad \text{AT} \quad t = t_f \;;\; y = 1, m \tag{2.4.85}$$

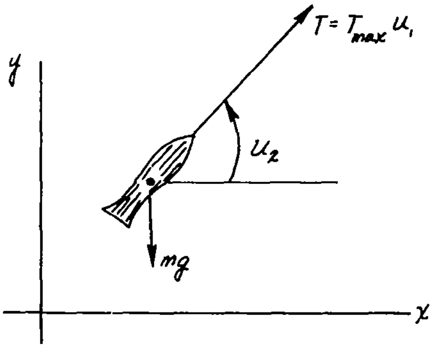
where the final time t_f may or may not be specified. The initial state is assumed specified with

$$x(t_0) = x_0 = \begin{pmatrix} x_{1_0} \\ x_{2_0} \\ \vdots \\ x_{n_0} \end{pmatrix} \quad \text{AT} \quad t = t_0 \tag{2.4.86}$$

If $\phi(x_f, t_f)$ is zero in Eq. (2.4.84) the Problem of Bolza reduces to the problem of Lagrange. If k(x,u) is zero the Mayer problem results. The type of physical situation which is implied by such a problem is illustrated in the following two examples.

Example (1) - Simple Guidance Problem

Consider the problem of maneuvering a rocket over a flat, airless earth which was treated in Section (2.2.2.3). The equation of motion in this case becomes



Sketch (2.4.5)

$$\ddot{x} = \frac{T_{MAX}\, u_1}{m} \cos u_2 \tag{2.4.87}$$

$$\ddot{y} = \frac{T_{MAX}\, u_1}{m} \sin u_2 - g \tag{2.4.88}$$

$$\dot{m} = -\frac{T_{MAX}\, u_1}{V} \tag{2.4.89}$$

where x and y represent the horizontal and vertical position, m the mass, V the exhaust velocity (a constant), and u_1 and u_2 are control variables denoting the throttle setting and steering angle. Since the thrust varies between zero and some maximum value, T_{max}, the throttle setting u_1 must satisfy the inequality

$$0 \leq u_1 \leq 1 \tag{2.4.90}$$

The initial position, velocity and mass are specified by

$$\left.\begin{aligned} x &= x_o \\ \dot{x} &= \dot{x}_o \\ y &= y_o \\ \dot{y} &= \dot{y}_o \\ m &= m_o \end{aligned}\right\} \quad t = t_o \tag{2.4.90B}$$

and at the terminal point, the position vector and the velocity magnitude are specified by

$$\begin{aligned} x &= x_f \\ y &= y_f \\ \dot{x}_f^2 + \dot{y}_f^2 &= V_f^2 \end{aligned} \tag{2.4.91}$$

where the final time itself, t_f, is not specified. The problem is to determine the control u_1 and u_2 such that the fuel expended during the maneuver is a minimum. Since the fuel is equal to the difference between the initial and terminal values for the mass, and since m_o is specified, minimizing the fuel is equivalent to minimizing the negative value of the final mass with

$$\varphi = - m_f = \text{minimum} \tag{2.4.92}$$

To put this problem in the Bolza format of Eqs. (2.4.81) to (2.4.86) define the new variables x_1, x_2, x_3, x_4, and x_5 by

$$\begin{aligned} x_1 &= x \\ x_2 &= y \\ x_3 &= \dot{x} \\ x_4 &= \dot{y} \\ x_5 &= m \end{aligned}$$

The state equations then beome

$$\begin{aligned} \dot{x}_1 &= x_3 & \qquad \dot{x}_4 &= \frac{T_{MAX}\, u_1}{x_5} \sin u_2 - g \\ \dot{x}_2 &= x_4 & & \\ \dot{x}_3 &= \frac{T_{MAX}\, u_1}{x_5} \sin u_2 - g & \qquad \dot{x}_5 &= - \frac{T_{MAX}\, u_1}{V} \end{aligned} \tag{2.4.93}$$

with the initial condition

$$x = x_0 = \begin{pmatrix} x_{1_0} \\ x_{2_0} \\ \vdots \\ x_{3_0} \end{pmatrix} \qquad (2.4.94)$$

and the terminal condition

$$\left.\begin{aligned} \psi_1 &= x_1 - x_f = 0 \\ \psi_2 &= x_2 - y_f = 0 \\ \psi_3 &= x_3^2 + x_4^2 - C = 0 \end{aligned}\right\} \text{AT } t = t_f \qquad (2.4.95A)$$

The admissible control set U is given by

$$u \in U \Longleftrightarrow \begin{array}{l} 0 \leq u_1 \leq 1 \\ u_2 \text{ arbitrary} \end{array} \qquad (2.4.95B)$$

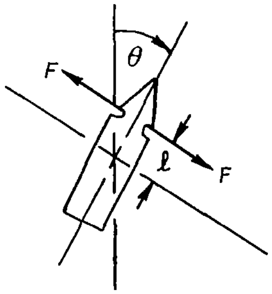
The optimizing criterion is

$$J = \phi(x_f, t_f) = -x_{5_f} \qquad (2.4.96)$$

Note that the integrand in Eq. (2.4.84) (i.e., the function $k(x,u)$) is identically zero for this problem.

Example (2) - Attitude Control Problem

Consider the simple single-axis attitude control system represented by the equation



Sketch (2.4.6)

$$I\ddot{\theta} = F\ell \qquad (2.4.97)$$

where I is the moment of inertia, F is the jet force and ℓ is the lever arm. Letting

$$F = u$$
$$x_1 = \frac{\theta I}{\ell}$$
$$x_2 = \frac{\dot{\theta} I}{\ell}$$

the state equations become

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \qquad (2.4.98A)$$

It is assumed, in addition, that the magnitude of the jet force F can vary from zero to a sufficiently large value so that essentially no constraint on the cantral action need be considered. Hence, the admissible control set U will be taken as the entire control space. The angular position and rate are specified initially with

$$\left.\begin{aligned} x_1 &= x_{1_0} \\ x_2 &= x_{2_0} \end{aligned}\right\} \text{ AT } t = t_0 \qquad (2.4.99)$$

and no terminal constraints are imposed (but, the final time, t_f, is specified). The control action u is to be selected so that the integral

$$J = \int_{t_0}^{t_f} (x_1^2 + x_2^2 + u^2)\, dt = \text{minimum} \qquad (2.4.100)$$

This function corresponds physically to keeping a combined measure of both the fuel and the angular displacement and rate errors as small as possible.

In subsequent sections both the above problem and the simple guidance problem will be analyzed using Dynamic Programming. Next, however, the partial differential equation, analogous to the Bellman equation for the problem of Bolza, will be developed.

2.4.9 Bellman Equation for the Bolza Problem

In this section a procedure very similar to that in Sections (2.4.1) and (2.4.2) will be followed. It will be shown, to begin with, that the Principle of Optimality, when applied to the problem of Bolza, is equivalent to a set of algebraic recursive equations. Next, it will be shown that under relatively unrestrictive assumptions, the limiting form of these recursive equations is a first-order, partial differential equation.

Let $R(t_0, x_0) = R(t_0, x_{1_0}, x_{2_0}, \ldots, x_{n_0})$ denote the minimum value of the performance index

$$R(t_0, x_0) = MIN \left\{ \int_{t_0}^{t_f} \ell(x,u)dt + \phi(x_f, t_f) \right\} \tag{2.4.101}$$

for the solution $x(t)$ which begins at the point

$$x = x_0 = \begin{pmatrix} x_{1_0} \\ x_{2_0} \\ \vdots \\ x_{n_0} \end{pmatrix} \tag{2.4.102}$$

satisfies the differential constraints

$$\dot{x}_i = f_i(x, u) \tag{2.4.103}$$

and the terminal conditions

$$\Psi_j(x_f, t_f) = 0 \quad ; \; j = 1, m \leq n+1 \tag{2.4.104}$$

and for which the control u (t) lies in the required set U . In other words, $R(t_0, x_0)$ is the minimum value of the performance index for the problem of Bolza as expressed in the preceding section. Eq. (2.4.106) is some times written either as

$$R(t_0, x_0) = \underset{\substack{u(\tau) \\ (t_0 \leq \tau \leq t_f)}}{MIN} \left\{ \int_{t_0}^{t_f} \ell(x,u)\,dt + \phi(x_f, t_f) \right\} \tag{2.4.105}$$

or, as

$$R(t_o, x_o) = \underset{\substack{u(\tau)\in U \\ (t_o \leq \tau \leq t_f)}}{MIN} \left\{ \int_{t_o}^{t_f} k(x, u)dt + \phi(x_f, t_f) \right\} \tag{2.4.106}$$

to indicate that the minimization is performed through the selection of the control u and that this control must lie in the set U .

To generalize Eq. (2.4.106), let $R(t,x(t))$ denote the minimum value of the performance index for the solution which starts at the point (t, x(t)) and satisfies the constraint conditions of Eqs. (2.4.103) and (2.4.104); that is,

$$R(t, x(t)) = \underset{\substack{u(\tau)\in U \\ (t \leq \tau \leq t_f)}}{MIN} \left\{ \int_{t}^{t_f} k(x, u)dt + \phi(x_f, t_f) \right\} \tag{2.4.107}$$

Similarly,

$$R(t+\Delta t, x(t+\Delta t)) = \underset{\substack{u(\tau)\in U \\ (t+\Delta t \leq \tau \leq t_f)}}{MIN} \left\{ \int_{t+\Delta t}^{t_f} k(x,u)dt + \phi(x_f, t_f) \right\} \tag{2.4.108}$$

where the solution starts at the point $(t+\Delta t, x(t+\Delta t))$ and satisfies constraints (2.4.103) and (2.4.104). Now, the Principle of Optimality states that if a solution which starts at the point (t, x(t)) is at the point $(t+\Delta t, x(t+\Delta t))$ after the first decision { or the first set of decisions }, all the remaining decisions must be optimal decisions if the solution itself is to be optimal. Putting this statement into mathematical form, leads to the equation

$$R(t, x(t)) = \underset{\substack{u(\tau)\in U \\ (t \leq \tau \leq t+\Delta t)}}{MIN} \left\{ R(t+\Delta t, x(t+\Delta t)) + k(x,u)\Delta t \right\} \tag{2.4.109}$$

Note the similarity between this equation and Eq. (2.4.21) developed for the problem of Lagrange. Again, it is to be emphasized that Eq. (2.4.109) is simply a mathematical statement of how the search procedure for the decision sequence is to be conducted.

To reduce (2.4.109) to a partial differential equation, one must assume that all second derivatives of R with respect to t and x are bounded; that is,

$$\left|\frac{\partial^2 R}{\partial t^2}\right| < \infty$$

$$\left|\frac{\partial^2 R}{\partial t\,\partial x_i}\right| < \infty ;\ i = 1, n \tag{2.4.110}$$

$$\left|\frac{\partial^2 R}{\partial x_i\,\partial x_s}\right| < \infty ;\ i, y = 1, n$$

Under this assumption, $R(t+\Delta t,\ x(t+\Delta t))$ has the series expansion

$$R(t+\Delta t,\ x(t+\Delta t) = R(t, x(t)) + \left\{\left(\frac{\partial R}{\partial x}\right)^T \frac{dx}{dt} + \frac{\partial R}{\partial t}\right\}\Delta t + O(\Delta t) \tag{2.4.111}$$

where T denotes transpose and

$$\frac{dx}{dt} = \dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{pmatrix} ; \quad \frac{\partial R}{\partial x} = \begin{pmatrix} \frac{\partial R}{\partial x_1} \\ \frac{\partial R}{\partial x_2} \\ \vdots \\ \frac{\partial R}{\partial x_n} \end{pmatrix}$$

Substituting (2.4.111) into (2.4.109) along with the values for $\dot{x}$ from (2.4.108), provides

$$\underset{\substack{u(\tau)\in V \\ (t\le\tau\le t+\Delta t)}}{MIN} \left\{ k(x,u) + \sum_{i=1}^{n} \frac{\partial R}{\partial x_i} f_i(x,u) + \frac{\partial R}{\partial t} + \frac{O(\Delta t)}{\Delta t} \right\} \Delta t = 0 \tag{2.4.112}$$

In the limit as $\Delta t \to 0$ this expression becomes

$$\underset{u(t)\in V}{MIN} \left\{ k(x,u) + \sum_{i=1}^{n} \frac{\partial R}{\partial x_i} f_i(x,u) + \frac{\partial R}{\partial t} \right\} = \underset{u(t)\in V}{MIN} \left\{ k(x,u) + \left(\frac{\partial R}{\partial x}\right)^T f(x,u) + \frac{\partial R}{\partial t} \right\} = 0 \tag{2.4.113}$$

which is a first-order, partial differential equation and will be referred to as the Bellman equation for the Problem of Bolza. The boundary condition which R(t, x(t)) must satisfy, will be considered next.

Since R(t, x(t)) is the minimum value of the performance index for the solution which starts at the point (t, x(t)), it follows that R must satisfy the terminal condition

$$R(t_f\,,\,x(t_f)) = R(t_f\,,\,x_f) = \varphi(x_f\,,\,t_f) \tag{2.4.114}$$

However, in addition, the terminal point (t_f, x (t_f)) must satisfy the terminal constraints

$$\psi_j(t_f\,,\,x_f) = 0 \quad \text{AT} \quad t = t_f \;;\; j = 1, m \tag{2.4.115}$$

Hence, the boundary condition on R becomes

$$R(t, x(t)) = \varphi(x(t), t) \quad \text{ON} \quad \psi_j\underset{t=t_f}{(x(t), t)} = 0 \;;\; j = 1, m \tag{2.4.116}$$

Analogous to the development of Eqs. (2.4.63B) from the boundary condition (2.4.63A) for the problem of Lagrange, the above expression can be reworked to yield the equivalent condition

$$\frac{\partial R}{\partial x_i} = \frac{\partial \phi}{\partial x_i} + \sum_{j=1}^{m} \mu_j \frac{\partial \psi_f}{\partial x_i} \quad \text{ON} \quad \psi_j \underset{t=t_f}{(x(t), t)} = 0 \;,\; j = 1, m \tag{2.4.117}$$

where μ is the vector

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{pmatrix} \tag{2.4.118}$$

If the final time t_f is itself not specified, then the additional boundary condition

$$\frac{\partial R}{\partial t} = \frac{\partial \varphi}{\partial t} + \sum_{j=1}^{m} \mu_j \frac{\partial \psi}{\partial t} \tag{2.4.119}$$

must hold.

Collecting the results of this section, the solution to the problem of minimizing the performance index

$$J = \int_{t_0}^{t_f} k(x,u)\,dt + \phi(x_f, t_f) \quad = \text{minimum}$$

subject to the boundary conditions

$$x = x_0 \quad \text{AT } t = t_0$$

$$\psi(x_f, t_f) = 0 \quad \text{AT } t = t_f$$

and the state equations

$$\dot{x}_i = f_i\,(x, u) \qquad i = 1, \ldots, n$$

can be generated by solving the partial differential equation

$$\underset{u(t)\,\epsilon\, U}{\text{MIN}} \left\{ k(x,u)\,\frac{\partial R}{\partial x} \quad f(x,u) + \frac{\partial R}{\partial t} \right\} = 0 \tag{2.4.120}$$

for R(t,x(t)) subject to the boundary conditions of Eq. (2.4.116) or Eqs. (2.4.117) and (2.4.119) where Eq. (2.4.119) is to hold only if the final time t_f is not explicitly specified.

2.4.10 Linear Problem with Quadratic Cost

To illustrate the method of solution by means of the Bellman partial differential equation, consider the following linear problem. Let the system state be governed by

$$\dot{x}_i = \sum_{j=1}^{n} a_{ij} x_j + \sum_{k=1}^{r} g_{ik} u_k \tag{2.4.121A}$$

or in the vector notation

$$\dot{x} = A(t) x + G(t) u \tag{2.4.121B}$$

where A is an n x n matrix and G is an n x r matrix. The initial state is specified, while the terminal state must satisfy the m constraint conditions

$$\left. \sum_{j=1}^{n} C_{ij} \, x_j (t_f) - d_i = 0 \right|_{t=t_f} ; \quad i = 1, m \tag{2.4.122A}$$

which can also be written as

$$C x_f - d = 0 \quad at \quad t = t_f \tag{2.4.122B}$$

where C is an m x n constant matrix and d is an m-dimensional constant vector. The problem is to select the control u so that the integral

$$\begin{aligned} J &= \int_{t_0}^{t_f} \left(\sum_{i,j=1}^{n} q_{i,j}^{(1)} x_i x_j + \sum_{i,j=1}^{r} q_{i,j}^{(2)} u_i u_j \right) dt \\ &= \int_{t_0}^{t_f} \left(x^T Q_1 x + u^T Q_2 u \right) dt \end{aligned} \tag{2.4.123}$$

with Q_1, a n x n symmetric matrix with elements $q^{(1)}_{i,j}$ and Q_2 a r x r symmetric matrix with element $q^{(2)}_{i,j}$. It is required that Q_2 be a positive, definite matrix (i.e., $u^T Q_2 u$ is always greater than zero for any control u not equal to zero). Furthermore, the admissible control set U is the entire r-dimensional control space; or in other words, no constraints are imposed on the components of the control vector u. Also, the final time, t_f, is explicitly specified. Note that the simple attitude controller which was considered in Section (2.4.8) is a special case of the above problem.

Substituting the state expressions of (2.4.121B) into (2.4.120) provides

$$\underset{u(t)\in U}{MIN}\left\{\left(\frac{\partial R}{\partial x}\right)^T(Ax+Gu)+(x^TQ_1x+u^TQ_2u)+\frac{\partial R}{\partial t}\right\}=0 \qquad (2.4.124)$$

Since the admissible set U is the entire control space, the minimization operation in (2.4.124) is accomplished simply by differentiating with respect to u . Thus (2.4.124) is equivalent to the two equations

$$\left(\frac{\partial R}{\partial x}\right)^T(Ax+Gu)+(x^TQ_1x+u^TQ_2u)+\frac{\partial R}{\partial t}=0 \qquad (2.4.125)$$

$$\left(\frac{\partial R}{\partial x}\right)^T G+2u^TQ_2=0 \qquad (2.4.126)$$

Using Eq. (2.4.122B), the boundary condition on R as given in Eq. (2.4.117) reduces to

$$\left(\frac{\partial R}{\partial x}\right)^T=\mu^TC \quad \text{ON} \quad \begin{matrix} Cx-d=0 \\ t=t_f \end{matrix} \qquad (2.4.127)$$

where μ is the m-dimensional constant vector

$$\mu=\begin{pmatrix}\mu_1\\ \mu_2\\ \vdots\\ \mu_m\end{pmatrix} \qquad (2.4.128)$$

Try as a solution, the function

$$R=x^TS(t)x+z(t)^Tx \qquad (2.4.129)$$

where S(t) is some n x n symmetric matrix and z(t) is an n vector. By the appropriate selection of S(t) and z(t), the R function in (2.4.129) can be made to satisfy both the differential equation and the boundary conditions of (2.4.125) to (2.4.127). This point will be illustrated next.

Substituting (2.4.129) into (2.4.125) and (2.4.126), it follows that the optimal control must satisfy

$$\begin{aligned} u &= -\frac{1}{2} Q_2^{-1} G^T \frac{\partial R}{\partial x} \\ &= -Q_2^{-1} G^T S(t)\, x(t) \end{aligned} \tag{2.4.130}$$

with S and z satisfying the ordinary differential equations

$$\dot{S} = -Q_1 - SA - A^T S + S G Q_2^{-1} G^T S \tag{2.4.131}$$

$$Z = -A^T Z + S G Q_2^{-1} G Z \tag{2.4.132}$$

For the boundary condition of (2.4.127) to hold, it follows that

$$2Sx + z = C^T \mu \quad \text{AT} \quad t = t_f \tag{2.4.133}$$

where the m-dimensional multiplier μ is selected so that the constraint conditions

$$Cx - d = 0 \quad \text{AT} \quad t = t_f \tag{2.4.134}$$

are satisfied.

Equation (2.4.131) governing the evolution of the matrix S is nonlinear and, hence, difficult to solve. However, the matrix S need not be explicitly evaluated to determine the optimal solution which from (2.4.130) to (2.4.135) depends only on the terms SX and z. It will be shown next that these terms satisfy a linear equation and can be evaluated rather easily.

Let P be the n-dimensional vector

$$p = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix} = -\frac{\partial R}{\partial x} \tag{2.4.135}$$

Substitution of this variable into (2.4.130) to (2.4.134) and using the state equation for x provides

$$u = \frac{Q_2^{-1} G^T p}{2} \tag{2.4.136}$$

$$\dot{p} = -A^T p + 2 Q_1 x \tag{2.4.137}$$

$$\dot{x} = Ax - \frac{G Q_2^{-1} G^T p}{2} \tag{2.4.138}$$

with the boundary conditions

$$x = x_0 \quad \text{AT} \quad t = t_0 \tag{2.4.139}$$

$$\left.\begin{aligned} p + C^T \mu &= 0 \\ Cx - d &= 0 \end{aligned}\right\} \; t = t_f \tag{2.4.140}$$

Note that the new equations in p and x are linear and that the two-point boundary problem as represented in Eqs. (2.4.137) to (2.4.140) can be solved directly (i.e., without iteration). The optimal control is then evaluated using Eq. (2.4.136). The method will be illustrated next on the simple attitude control problem of Section (2.4.8).

The governing equations for the attitude controller are

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \qquad (2.4.141)$$

with the initial conditions

$$\begin{aligned} x_1(t_0) &= x_{1_0} \\ x_2(t_0) &= x_{2_0} \end{aligned} \qquad (2.4.142)$$

and with no terminal conditions imposed; that is, all the ψ_{ψ} in Eq. (2.4.118B) are identically zero. The matrices A, Q_1, Q_2 and G for this problem are

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} , \quad G = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ Q_1 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} , \quad Q_2 = 1 \end{aligned} \qquad (2.4.143)$$

Hence, Eqs. (2.4.138) to (2.4.140) become

$$\begin{aligned} u &= p_2/2 & \dot{x}_1 &= x_2 \\ \dot{p}_1 &= 2x_1 & \dot{x}_2 &= p_2/2 \\ \dot{p}_2 &= -p_1 + 2x_2 \end{aligned} \qquad (2.4.144)$$

Since no terminal constraints are imposed, it can be shown from Eqs. (2.4.117), (2.4.133), and (2.4.140) that at the terminal point

$$p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = 0 \quad \text{AT} \quad t = t_f \qquad (2.4.145)$$

The solution to this system can be written

$$\begin{pmatrix} x(t) \\ p(t) \end{pmatrix} = \mathcal{X}(t, t_0) \begin{pmatrix} x(t_0) \\ p(t_0) \end{pmatrix} \qquad (2.4.146)$$

where χ is the 4 x 4(matrix fundamental matrix solution)

$$\chi(t,t_0) = \begin{pmatrix} \cosh(\alpha_1) & 0 & \sqrt{2}\sinh(\alpha_1) & 0 \\ 0 & e^{-(\frac{t-t_0}{2})}\cosh(\alpha_2) & 0 & \frac{1}{2\sqrt{5}}e^{-\frac{(t-t_0)}{2}}\sinh(\alpha_2) \\ \sqrt{2}\sinh(\alpha_1) & 0 & \cosh(\alpha_1) & 0 \\ 0 & \frac{1}{2\sqrt{5}}e^{-(\frac{t-t_0}{2})}\sinh(\alpha_2) & 0 & e^{-\frac{(t-t_0)}{2}}\cosh(\alpha_2) \end{pmatrix} \qquad (2.4.147)$$

and where $p(t_0)$ is the initial value of p and is to be selected consistent with (2.4.145).

2.4.11 Dynamic Programming and the Pontryagin Maximum Principle

For the Bolza problem under consideration, the state system is given by

$$\dot{x}_i = f_i(x, u) \tag{2.4.148}$$

and the boundary conditions by

$$x = x_0 \quad \text{AT} \quad t = t_0 \tag{2.4.149}$$

$$\psi_j(x_f, t_f) = 0 \;;\; j = 1, m \quad \text{AT} \quad t = t_f \tag{2.4.150}$$

where t_f may or may not be specified. The control action u is to be selected so that an integral plus the function of the terminal state of the form

$$J = \int_{t_0}^{t_f} k(x, u)\,dt + \phi(x_f, t_f) = \text{minimum} \tag{2.4.151}$$

is a minimum. An application of the Pontryagin Maximum Principle to this problem (Ref. 2.4.3) leads to the following requirements:

(1) the control u is selected from U so that at each instant the quantity H where

$$H = p \cdot f(x, u) - k(x, t) = \sum_{j=1}^{n} p_j f_j(x, u) - k(x, u) \tag{2.4.152}$$

is maximized.

(2) the adjoint vector p satisfies the differential equations

$$\dot{p}_i = -\frac{\partial H}{\partial x_i} = -\sum_{j=1}^{n} p_j \frac{\partial f_j}{\partial x_i} + \frac{\partial k}{\partial x_i} \;;\; i = 1, n \tag{2.4.153}$$

*(3) the boundary conditions on the p vector are

$$p_i + \frac{\partial \phi}{\partial x_i} + \sum_{j=1}^{m} \mu_j \frac{\partial \psi_j}{\partial x_i} = 0 \quad ; \quad t = t_f \quad i = 1, n \tag{2.4.154}$$

where μ is an m-dimensional constant vector which is selected so that the m terminal constraints are satisfied.

(4) $p_i^{(+)} = p_i^{(-)}$ {i.e., p is continuous across discontinuities in u}

(2.4.155A)

It is not difficult to verify that the p vector used in the preceding section, in connection with the linear optimization problem, does satisfy these conditions. It will be shown in this section that the Bellman equation, Eq. (2.4.120), can be used to develop the above equations of the Maximum Principle for the general Bolza Problem. The approach to be taken is essentially the same as that used in Section (2.4.4) to relate the Calculus of Variations and Dynamic Programming for the problem of Lagrange.

From Eq. (2.4.120), the Bellman equation for the Bolza Problem is

$$\underset{u(t)\epsilon U}{MIN} \left\{ k(x,u) + \left(\frac{\partial R}{\partial x}\right)^T f(x,u) + \frac{\partial R}{\partial t} \right\} = 0 \tag{2.4.155B}$$

with the boundary condition

$$\frac{\partial R}{\partial x_i} = \frac{\partial \phi}{\partial x_i} + \sum_{j=1}^{m} \mu_j \frac{\partial \psi_j}{\partial x_i} = 0 \qquad t = t_f \tag{2.4.156}$$

** $$\frac{\partial R}{\partial t} = \frac{\partial \phi}{\partial t} + \sum_{j=1}^{m} \mu_j \frac{\partial \psi_j}{\partial t}$$

* If the final time is not explicitly specified, the terminal condition must hold:

$$\sum_{i=1}^{n} p_i f_i(x,u) - k(x,u) - \frac{\partial \varphi}{\partial t} - \sum_{j=1}^{m} \mu_j \frac{\partial \psi}{\partial t} = 0 \tag{2.4.154A}$$

** this equation is valid only if the final time is unspecified.

Since $\frac{\partial R}{\partial t}$ does not depend on u explicitly, the control u which minimizes the bracketed quantity in Eq. (2.4.155) also minimizes the quantity $\sum_{i=1}^{n} \frac{\partial R}{\partial x_i} f_i(x, u) + k(x,u)$, or alternately, maximizes the quantity $-\sum_{i=1}^{n} \frac{\partial R}{\partial x_i} f_i(x,u) - k(x,u)$. Let the p vector be defined by

$$p_i = -\frac{\partial R}{\partial x_i} \tag{2.4.157}$$

Then, it follows that the optimal control is that control in the set which maximizes $\sum_{i=1}^{n} p_i f_i(x,u) \quad - \quad K(x,u)$. Thus, condition (1) of the Maximum Principle is satisfied. With the definition of Eq. (2.4.157), the boundary condition in Eq. (2.4.156) is

$$p_i + \frac{\partial \varphi}{\partial x_i} + \sum_{j=1}^{n} \mu_j \frac{\partial \psi_j}{\partial x_i} \qquad i = 1,\cdots,n \tag{2.4.158}$$

And since from(2.4.156)

$$\frac{\partial R}{\partial t} = -\sum_{i=1}^{n} \frac{\partial R}{\partial x_i} f_i(x, u_{opt}) - K(x,u) = \sum_{i=1}^{n} p_i f_i - K(x,u)$$

$$\sum_{i=1}^{n} p_i f_i = \frac{\partial \phi}{\partial t} + \sum_{\gamma=1}^{m} \mu_\gamma \frac{\partial \psi_\gamma}{\partial t} + K(x,u) \tag{2.4.158A}$$

which agrees with condition (3) of the Maximum Principle.

To demonstrate that the p vector satisfies Eq. (2.4.153), a technique developed by S. Dreyfus [Ref. (2.4.1)] is used. From (2.4.157),

$$\frac{d}{dt} p_i = -\frac{d}{dt} \frac{\partial R}{\partial x_i} \quad ; \quad i = 1, n \tag{2.4.159}$$

But from (2.4.155B) it follows that the quantity $k(x,u)+\sum \frac{\partial R}{\partial x_i} f_i(x,u)+\frac{\partial R}{\partial t}$ has a minimum of $u = u_{opt}$ and that this minimum value is zero. If u is held fixed at its optimum value which corresponds to some point $(\hat{x},\hat{t})$ $\{i.e., u_{opt} = u_{opt}(x,t)\}$ then this bracketed quantity, considered as a function of x,t , will have a minimum at the point $\hat{x}$, $\hat{t}$. Hence,

$$\frac{\partial}{\partial x_y}\left\} k(x,u)+\sum_{j=1}^{n} \frac{\partial R}{\partial x_j} f_j(x, u_{opt})+\frac{\partial R}{\partial t}\right\}_{x=\hat{x}} = 0 \qquad y = 1, n$$

$$= \frac{\partial k}{\partial x_y} \sum_{j=1}^{n}\left(\frac{\partial^2 R}{\partial x_i \partial x_y} + \frac{\partial R}{\partial x_j} \frac{\partial f_i}{\partial x_y}\right)+\frac{\partial^2 R}{\partial t\, \partial x_y} = 0$$

and substituting this expression into (2.4.159), yields the desired result; namely,

$$\frac{dp_i}{dt} = -\frac{d}{dt}\left(\frac{\partial R}{\partial x_i}\right) = -\left\{\sum_{y=1}^{n} \frac{\partial^2 R}{\partial x_i\, \partial x_y} f_y(x,y)+\frac{\partial^2 R}{\partial t\, \partial x_i}\right\} = \sum_{y=1}^{n} \frac{\partial R}{\partial x_y} \frac{\partial f_y}{\partial x_i}$$

$$= \frac{\partial k}{\partial x_i} - \sum_{y=1}^{n} p_y \frac{\partial f_y}{\partial x_i} \qquad (2.4.160)$$

The fourth condition follows directly from the original assumption on the R function needed to develop the Bellman equation. This assumption [see (2.4.115)] required that the second derivatives of R be bounded; hence, the first derivatives must be continuous. Thus,

$$\left(\frac{-\partial R}{\partial x_i}\right)^{(+)} = p_i^{(+)} = \left(\frac{-\partial R}{\partial x_i}\right)^{(-)} = p_i^{(-)}$$

and condition (4) is satisfied. As discussed at the start of this section, this requirement in the second derivatives is not always satisfied, a point which will be treated later on.

The conditions of the Maximum Principle as developed from the Bellman equation and represented in (2.4.152) to (2.4.155) will now be used to solve the first example problem in Section (2.4.8).

The guidance problem of Section (2.4.8) is represented by the equations

$$\dot{x}_1 = x_3 \qquad \dot{x}_4 = \frac{T_{MAX}\, u_1}{x_5} \sin u_2 - g$$

$$\dot{x}_2 = x_4$$

$$\dot{x}_3 = \frac{T_{MAX}\, u_1}{x_5} \cos u_2 \qquad \dot{x}_5 = \frac{T_{MAX}\, u_1}{V} \qquad (2.4.160A)$$

with

$$U \Longleftrightarrow \begin{matrix} 0 \le u_1 \le 1 \\ u_2 \end{matrix}$$

$$x = x_0 \ , \quad t = t_0$$

$$\phi(x_f, t_f) = - x_{5_f} - M_{IN}$$

$$\left.\begin{aligned} \psi_1 &= x_1 - x_f = 0 \\ \psi_2 &= x_2 - y_f = 0 \\ \psi_3 &= x_3^2 + x_4^2 - C = 0 \end{aligned}\right\} t = t_f \qquad (2.4.160B)$$

The P vector for the system is

$$\begin{aligned} \dot{p}_1 &= 0 \\ \dot{p}_2 &= 0 \\ \dot{p}_3 &= - p_1 \\ \dot{p}_4 &= - p_2 \\ \dot{p}_5 &= \frac{p_3 \, T_{MAX} \, u_1 \cos u_2}{x_5^2} + \frac{p_4 \, T_{MAX} \, u_1 \sin u_5}{x_5^2} \end{aligned} \qquad (2.4.161)$$

and the boundary conditions

$$\begin{aligned} p_{1_f} &= \mu_1 \\ p_{2_f} &= \mu_2 \\ p_{3_f} + 2\mu_3 \, x_{3_f} &= 0 \\ p_{4_f} + 2\mu_3 \, x_{4_f} &= 0 \\ p_{5_f} &= 1 \end{aligned} \qquad (2.4.162)$$

where the μ_i are constants to be selected so that the boundary conditions on the state variables $\{ i.e., \psi_i = 0$ at $t = t_f \}$ are satisfied. Since the final time is not explicitly stated, the additional condition

$$\sum p_i f_i(x, u) = 0 \quad AT \quad t = t_f \qquad (2.4.162A)$$

must also hold.

From Eq. (2.4.152) the control u is to be selected to maximize

$$H = \sum_{i=1}^{n} p_i f_i(x, u) = p_1 x_3 + p_2 x_4 + p_3 \frac{T_{MAX} u_1}{x_5} \cos u_2$$

$$+ p_4 \left\{ \frac{T_{MAX} u_1}{x_5} \sin u_2 - g \right\} - p_5 \frac{T_{MAX} u_1}{V}$$

Let

$$\frac{p_3}{\sqrt{p_3^2 + p_4^2}} = \cos \phi \quad , \quad \frac{p_4}{\sqrt{p_3^2 + p_4^2}} = \cos \phi$$

then

$$H = \sqrt{p_3^2 + p_4^2} \; \frac{T_{MAX} u_1}{x_5} \left\{ \cos(u_2 - \phi) - \frac{p_5 x_5}{V\sqrt{p_3^2 + p_4^2}} \right\} + p_1 x_3 + p_2 x_4 - p_4 g$$

Thus, for the control to maximize H, cos $(u_2 - \phi) = 1$ and

$$\cos u_2 = \frac{p_3}{\sqrt{p_3^2 + p_4^3}}$$

$$\sin u_2 = \frac{p_4}{\sqrt{p_3^2 + p_4^2}} \qquad (2.4.163)$$

with the H function

$$H = \frac{T_{MAX} u_1}{x_5} \left(p_3^2 + p_4^2 - \frac{p_5 x_5}{V} \right) + p_1 x_3 + p_2 x_4 - p_4 g$$

Hence, for u_1 to maximize H

$$u_1 = 1 \; ; \; \phi > 0$$

$$u_1 = 0 \; ; \; \phi < 0 \qquad (2.4.164)$$

where

$$\varphi = \sqrt{p_3^2 + p_4^2} - \frac{p_5 x_5}{V} \tag{2.4.165}$$

With the control known as a function of the state and P vectors, the solution to the problem can be achieved numerically on a digital computer with the boundary conditions of (2.4.160B), (2.4.162) and (2.4.162A) just sufficient to determine a unique solution to the differential equations in (2.4.160A) and (2.4.161). The solution to this problem is considered in some detail in Refs. (2.4.4) and (2.4.5).

(2.4.12) Some Limitations on the Development of the Bellman Equation

The preceding paragraphs of this section have been primarily concerned with reducing the computational algorithm inherent in the Principle of Optimality to a certain partial differential equation called the Bellman equation. From this equation various additional properties of the optimal decision sequence have been developed and shown to be equivalent to the necessary conditions normally developed by means of the Calculus of Variations or the Maximum Principle. In some special cases, however, the Bellman equation, which results from considering the Principle of Optimality in the limit as the separation between states and decision goes to zero, is erroneous.

In developing the Bellman equation which, for the Bolza problem, took the form

$$\underset{u(t)\in U}{MIN}\left\{ \ell(x,u)+\sum_{i=1}^{n}\frac{\partial R}{\partial x_i}\, f_i(x,u)+\frac{\partial R}{\partial t}\right\} = 0 \qquad (2.4.166)$$

it was necessary to assume that all second derivatives of R exist and are bounded (see Eq. 2.4.115) which implies, among other things, that all first derivatives of R exist and are continuous. It is shown in Ref. (2.4.3) that occasionally the derivatives $\frac{\partial R}{\partial x_i}$ do not exist at all points in the (t, x) space and hence, that Eq. (2.4.166) is not always correct. The type of problem in which this may happen is one in which the control action appears linearly in the state equations; that is, the state equations take the form

$$\dot{x}_i = f_i(x,u) = f_i^{(1)}(x) + f_i^{(2)}(x)\, u \qquad (2.4.167)$$

with the result that the optimal control is bang-bang in that it jumps discontinuously from one boundary of the control set U to another boundary. If there exists a curve in the (x, t) space (called a switching curve) with the property that all optimal trajectories when striking the curve experience a control discontinuity, and if furthermore a finite segment of the optimal solution lies along the switching curve, then the derivatives $\frac{\partial R}{\partial x_i}$ may not exist along the switching curve and Eq. (2.4.166) may not be applicable.

As an example of such a problem, consider the second order integrator

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \qquad (2.4.168)$$

with

$$|u| \leq 1 \tag{2.4.169}$$

and with the boundary conditions

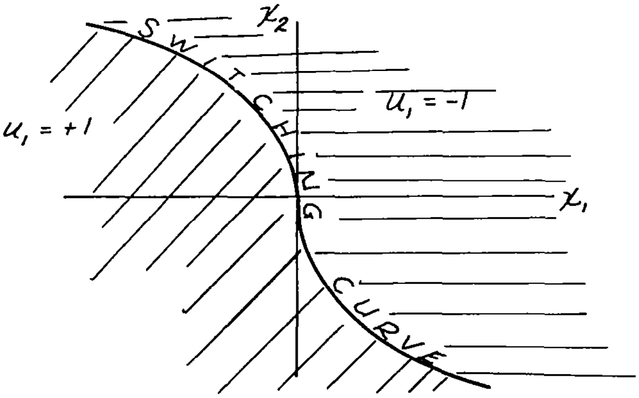
$$\left.\begin{aligned} x_1 &= x_{1_0} \\ x_2 &= x_{2_0} \end{aligned}\right\} \quad t = 0 \tag{2.4.170}$$

$$\left.\begin{aligned} x_1 &= 0 \\ x_2 &= 0 \end{aligned}\right\} \quad t = t_f \tag{2.4.171}$$

The optimizing criterion is time, with

$$\phi(x_f, t_f) = t_f = \text{MIN} \tag{2.4.172}$$

It can be shown using the Maximum Principle that the solution to this problem consists of segments along which u = + 1 and segments along which u = - 1 with the two types of control separated by the switching curve as shown on Sketch (2.4.7). Since the switching curve is the only curve which satisfies both the state equations and the optimal control condition, and which goes through the origin, it follows that all optimal trajectories have segments lying on the switching curve.



Now if the Maximum Principle is used to determine the optimal solution for a variety of initial conditions, the minimum time t_f can be developed as a function of x_{1_0} and x_{2_0} and this time is equal to the function $R(t_0, x_0)$ appearing in the Bellman equation. Thus, $R(t, x(t))$ can be developed from the Maximum Principle, and what's more, the development is straightforward and can be accomplished analytically. It is then just a matter of differentiation to show that $\frac{\partial R}{\partial x_1}$ and $\frac{\partial R}{\partial x_2}$ are discontinuous across the switching curve and that the Bellman equation does not apply along this curve.

Dreyfus, in Chapter 6 of Ref. (2.4.1), shows how problems of this type can be handled using Dynamic Programming. The method consists essentially of solving Eq. (2.4.166) on both sides of the switching curve and then patching the solution across the curve through the use of a specialized form of the Bellman equation which is valid along the switching curve. To use such an approach, however, one must know that the problem possesses a switching curve and also the equation of this curve -- knowledge which one usually does not have until after the solution has been developed. Hence, while a modified Bellman equation can be developed in these special cases from which a solution to the problem can be generated, the development requires a priori knowledge of the solution structure -- a rather imperfect state of affairs to say the least.

This shortcoming of the limiting form of Dynamic Programming is by no means severe. The class of problems to which the Bellman equation of (2.4.166) does not apply appears to be rather small with the problem themselves atypical. Hence, one can feel reasonably confident that the Bellman equation as developed for a particular problem is indeed correct, unless, of course, the problem possesses the linear structure indicated in Eq. (2.4.167) and there is evidence of the existence of a switching curve. In such cases one should exercise some caution in working with the Bellman equation.

2.5 DYNAMIC PROGRAMMING AND THE OPTIMIZATION OF STOCHASTIC SYSTEMS

2.5.1 Introduction

The previous sections of this report have dealt exclusively with the optimization of deterministic systems. In this section, some optimization problems are considered in which the equations describing the system contain stochastic or random elements. This extension is considered desirable, if not necessary, since all phenomena occurring in nature are stochastic. That is, every physical process contains some parameters or elements which are not known exactly but which are known in some statistical sense. Fortunately, in many systems, the total effect of these random parameters on system behavior is negligible and the system can be approximated by a deterministic model and analyzed using standard procedures. In other cases, however, the random elements are not negligible and may dominate those elements which are known precisely. The midcourse correction problem encountered in lunar and planetary transfer maneuvers is a case in point.

Due to injection errors at the end of the boost phase of a planetary transfer, the vehicle's trajectory will differ slightly from the desired nominal condition, and hence, some correction maneuver will be required. To make such a maneuver, the trajectory error must be known; and so radar and optical measurement data are collected. This data will lead to a precise determination of the trajectory error only if the data itself are precise. Unfortunately, the measurements and measuring devices are not perfect. Hence, the midcourse maneuver which is made will not null the trajectory error. Rather, it will null some estimate of the error, for example, the most probable value of the error. The determination of when and how to make these corrections so that the fuel consumed is a minimum is a problem of current interest in stochastic optimization theory. Note that if a deterministic model of the planetary transfer problem were used, the problem itself would cease to exist.

At the present time, the area of optimal stochastic control is just beginning to be examined. Thus, there are no standard equations or standard approaches which can be applied to such systems. In fact, the literature on the subject contains very few problems which have been solved. One reason for this limited amount of literature is that the fundamental equations which are encountered are of the diffusion type; that is, they are second order partial differential equations. Hence, the method of characteristics, which is used in the deterministic case and which reduces the Bellman equation to a set of ordinary differential equation, can not be applied; rather, the partial differential equations must be utilized directly.

A second factor contributing to the difficulty in handling stochastic problems is that the type of feedback being considered must be explicitly accounted for. This situation is just the opposite of that encountered in the deterministic case. If the initial state is known along with the control to be applied in a deterministic system, then all subsequent states can be determined simply by integrating the governing equations. In the stochastic case, the initial state and control are insufficient to determine all subsequent states due to the presence of disturbing forces and other random elements. Hence, only an estimate of the state can be generated and the estimate will be good or bad depending on the rate, quality and type of information which is being gathered. This estimate or feedback loop must be included in the analysis of the stochastic system.

Finally, a third factor which complicates the stochastic problem is the inclusion of terminal constraints. In the deterministic case, the presence or absence of terminal constraints has little effect on the analysis involved. In the stochastic case, the inclusion of terminal constraints makes the analysis much more difficult since the means employed to handle the constraints is not unique. For this reason, most of the literature on optimal stochastic control does not consider the terminal constraint problem.

In the following paragraphs, only one very specialized type of stochastic problem will be analyzed; namely, the stochastic analog of the linear-quadratic cost problem treated in Section (2.4.10). While this problem is not typical of all stochastic optimization problems, it can be solved rather easily and is frequently used as a model for stochastic problems occurring in flight control systems and trajectory analyses. Also, three different feedback loops or types of observability will be considered:

(1) Perfectly Observable: the state or output of the system can be determined exactly at each instant of time.

(2) Perfectly Inobservable: no knowledge of the state or output of the system is available once the system is started.

(3) Partially Observable: observations of the state or output of the system are made at each instant but the observations themselves are contaminated by noise.

Of the three, the partially observable case is the most representative of the type of situation which would occur in an actual system. The other two are limiting cases, with the perfectly observable or perfectly inobservable system resulting as the noise in the observations becomes zero or infinite, respectively.

2.5.2 Problem Statement

Let the system be described by the differential equation

$$\dot{x} = Ax + Gu \tag{2.5.1}$$

where x is an n vector denoting the state of the system, u is an r vector denoting the control, ξ is an n vector denoting noise or disturbing forces and A and G are $n \times n$ and $n \times r$ matrices, respectively. The state of the system is not known initially. Rather, the initial state, x_0, is a Gaussian random variable with mean $\hat{x}_0$ and covariance matrix V_0; that is,

$$\begin{aligned} E(x_0) &= \hat{x}_0 \\ E\left\{(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right\} &= V_0 \end{aligned} \tag{2.5.2}$$

where E denotes the expectation operator defined over the entire ensemble of states. Alternately, the Gaussian random variable can be represented by its density function

$$P(x_0) = P(x_{1_0}, x_{2_0}, \cdots x_{n_0}) = \frac{1}{(2\pi)^{n/2}\sqrt{|V_0|}}\, e^{-1/2\left\{(x_0 - \hat{x}_0)^T V_0^{-1}(x_0 - \hat{x}_0)\right\}} \tag{2.5.3}$$

with

$$\begin{aligned} E(x_0) &= \int_{-\infty}^{\infty} x_0 P(x_0)\, dx_0 \\ E\left\{(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right\} &= \int_{-\infty}^{\infty} (x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T P(x_0)\, dx_0 \end{aligned} \tag{2.5.4}$$

Note that the case in which x_0 is precisely specified can also be included in this formulation by requiring that

$$\begin{aligned} E(x_0) &= \hat{x}_0 \\ E\left\{(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right\} &= V_0 = 0 \end{aligned} \tag{2.5.5}$$

where now $\hat{x}_0$ denotes the specified value of x_0. In this case, the density function in Eq. (2.5.3) becomes a product of n Dirac delta functions with

$$P(x_0) = \delta(x_{1_0} - \hat{x}_{1_0})\, \delta(x_{2_0} - \hat{x}_{2_0}) \cdots \delta(x_{n_0} - \hat{x}_{n_0}) \tag{2.5.6}$$

The noise vector, ξ , which appears in the state equation is required to be a Guassian white noise with zero mean and covariance matrix $\Sigma(t)$. Thus

$$E(\xi) = 0$$

$$E(\xi(t)\,\xi^T(\tau)) = \Sigma(t)\,\delta\,(t-\tau) \qquad (2.5.7)$$

where $\delta(t-\tau)$ is again the Dirac delta function denoting the "white" or uncorrelated property of the noise. Note that $\Sigma(t)$ is a symmetric matrix and will be positive definite in the case in which ξ is truly an n vector. In the case in which ξ is not n dimensional, additional components of zero mean and zero variance can be added to make it n dimensional. In such cases, the $n \times n$ symmetrix matrix $\Sigma(t)$ is only positive semi-definite. An example of this will be given later.

The optimization problem is to determine the control action such that the expected value of a certain functional J is a minimum; that is,

$$E\{J\} = E\left\{\int_{t_0}^{t_f} (x^T Q_1 x + u^T Q_2 u)\,dt + x_f^T \Lambda\, x_f\right\} \qquad (2.5.8)$$

where Q_2 is a positive definite symmetric matrix and Q_1 and Λ are positive semi-definite symmetric matrices. The admissible control set U is the entire r dimensional control space. Thus, no restrictions are placed on the control vector u . Also, it is assumed that no constraints are placed on the terminal state.

This problem is quite similar to the linear quadratic cost problem treated in Section (2.4.10). The state equations are the same except for the disturbing force ξ , while the problem of minimizing a quadratic functional J has been replaced by the problem of minimizing the average or expected value of J .

To illustrate the type of physical situation that can be represented by Eqs. (2.5.1) to (2.5.8), consider a stochastic version of the simple attitude control problem treated in Section (2.4.8). Let the system equation be [see Eq. (2.4.97) and Sketch (2.4.6)].

$$I\ddot{\theta} = F\ell + \xi_2$$

where I is the moment of inertia, F is the applied force, ℓ is the lever arm and ξ_2 is a Gaussian white noise (one dimensional) with zero mean and variance σ_2 ; that is,

* · As stated previously, only quadratic cost will be considered at this time.

$$E(\xi_2) = 0$$

$$E\left\{\xi_2(t)\xi_2(\tau)\right\} = \sigma^2\delta(t-\tau)$$

Now, introducing the variables

$$x_1 = \frac{\theta I}{\ell}$$

$$x_2 = \frac{\dot{\theta} I}{\ell}$$

$$F = u$$

and letting ξ denote the 2 vector

$$\xi = \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}$$

where ξ_1 is identically zero; equivalently, ξ_1 is a Gaussian white noise with zero mean and zero variance. Under this change of variables, the system equation becomes

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u + \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix}$$

with

$$E\left(\xi_2\right) = 0$$

$$E\left(\xi(t)\xi(\tau)\right) = \Sigma(t)\delta(t-\tau) = \begin{pmatrix} 0 & 0 \\ 0 & \sigma^2 \end{pmatrix}\delta(t-\tau)$$

Now, the performance index is defined to be

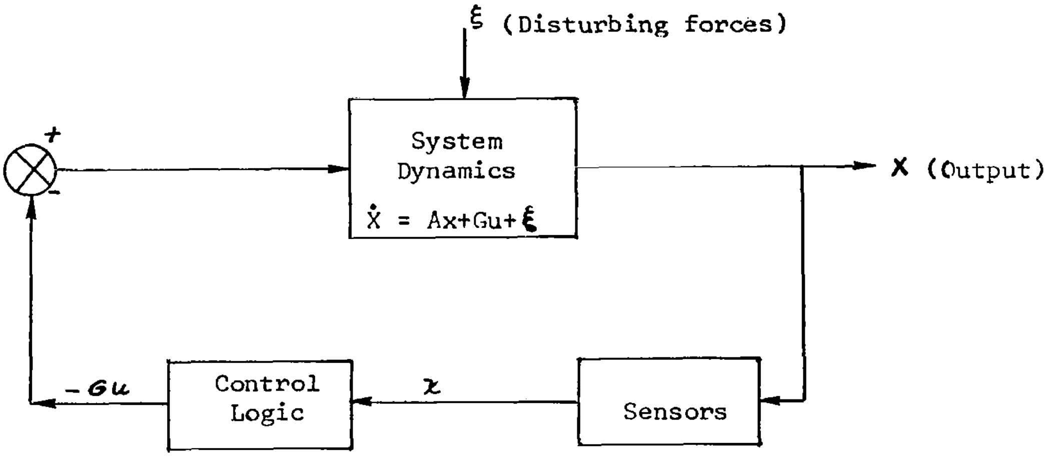
$$E(J) = E\left\{\int_{t_0}^{t_f}(x_1^2 + x_2^2 + u^2)dt\right\} = min$$

It is observed that this problem attempts to keep the expected value of a combined measure of the fuel and angular displacement and rate errors as small as possible.

In order to proceed with the solution for the general problem given in Eqs. (2.5.1) to (2.5.8), the feedback or observation loop must be specified. The reason for this is that the averaging process, that is, the expectation operator in Eq. (2.5.8), varies as the type and quantity of the observational data varies. As indicated in the introduction, three different types of observability will be considered and these are treated in the three succeeding sections.

2.5.2.1 Perfectly Observable Case

In the perfectly observable case, it is assumed that the entire output of the system (i.e., all the components of the vector x) can be determined exactly at each instant of time. This type of situation is represented in the sketch below.



Sketch (2.5.1)

The state equations are

$$\dot{X} = AX + Gu + \xi \qquad (2.5.9)$$

where ξ is a Gaussian white noise with

$$E(\xi) = 0$$

$$E\left\{\xi(t)\,\xi(\tau)^T\right\} = \Sigma(t)\,\delta(t-\tau) \qquad (2.5.10)$$

But, since the system is perfectly observable, it is assumed that the initial state of the system is known exactly with

$$X = \hat{X}_0 \text{ at } t = t_0 \qquad (2.5.11)$$

Thus, the performance index takes the form

$$\underset{\xi}{E}\left\{J\right\} = \underset{\xi}{E}\left\{\int_{t_0}^{t_f}(x^TQ_1x + u^TQ_2u)\,dt + x_f^T\Lambda x_f\right\} = \text{MIN.} \qquad (2.5.12)$$

where the variable ξ has been placed under the expectation operator to indicate that the "averaging" is to be conducted over this particular variable, the only random element appearing in the problem.

To determine the solution to this problem, the Principle of Optimality can be employed essentially as it was in the deterministic case. Let $R(x,t)$ denote the minimum value of the performance index for the system which starts in state x at time t ; that is,

$$R(x,t) = \underset{t\le\tau\le t_f}{\underset{u(\tau)}{\text{MIN.}}}\ \underset{\xi(\tau)}{E}\left\{\int_{t}^{t_f}(x^TQ_1x + u^TQ_2u)\,dt + x_f^T\Lambda x_f\right\} \qquad (2.5.13)$$

Now, this expression can be rewritten as

$$R(x,t) = \underset{t\le\tau\le t_f}{\underset{u(\tau)}{\text{MIN.}}}\ \underset{\xi(\tau)}{E}\left\{\int_{t}^{t+\Delta t}(x^TQ_1x + u^TQ_2u)\,dt + \int_{t+\Delta t}^{t_f}(x^TQ_1x + u^TQ_2u)\,dt + x_f^T\Lambda x_f\right\}$$

$$= \underset{t\le\tau_1\le t+\Delta t}{\underset{u(\tau_1)}{\text{MIN.}}}\ \underset{\xi(\tau_1)}{E}\left\{\underset{t+\Delta t\le\tau_2\le t_f}{\underset{u(\tau_2)}{\text{MIN.}}}\ \underset{\xi(\tau_2)}{E}\left[(x^TQ_1x + u^TQ_2u)\Delta t + o(\Delta t)\right.\right.$$

$$\left.\left.+\int_{t+\Delta t}^{t_f}(x^TQ_1x + u^TQ_2u)\,dt + x_1^T\Lambda x_f\right]\right\}$$

(2.5.14)

But, since the first term in the square bracket on the right of (2.5.14) does not depend on $u(\tau_2)$ or $\xi(\tau_2)$ for $t+\Delta t \le \tau_2 \le t_f$, Eq. (2.5.14) can be written in the form

$$R(X,t) = \underset{\substack{u(\tau_1)\,\xi(\tau_1)\\ t \leq \tau_1 \leq t+\Delta t}}{\text{MIN}\ E} \left\{ (x^T Q_1 x + u^T Q_2 u)\Delta t + o(\Delta t) + \underset{\substack{u(\tau_2)\ \xi(\tau_2)\\ t+\Delta t \leq \tau_2 \leq t_f}}{\text{MIN}\ E} \left[\int_{t+\Delta t}^{t_f} (x^T Q_1 x + u^T Q_2 u)dt + X_f^T \Lambda X_f \right] \right\}$$

$$= \underset{\substack{u(\tau_1)\ \ \xi(\tau_1)\\ t \leq \tau_1 \leq t+\Delta t}}{\text{MIN}\ E} \left\{ (x^T Q_1 x + u^T Q_2 u)\,\Delta t + o(\Delta t) + R(x+\Delta x,\ t+\Delta t) \right\} \qquad (2.5.15)$$

Finally, since the first term on the right of (2.5.15) does not depend on $\xi(\tau_1)$ for $t \leq \tau_1 \leq t+\Delta t$

$$R(x,t) = \underset{\substack{u(\tau_1)\\ t \leq \tau_1 \leq t+\Delta t}}{\text{MIN}} \left\{ (x^T Q_1 x + u^T Q_2 u)\Delta t + \underset{\substack{\xi(\tau_1)\\ t \leq \tau_1 \leq t+\Delta t}}{E} \left[R(x+\Delta x, t+\Delta t) + o(\Delta t) \right] \right\} \qquad (2.5.16)$$

Equation (2.5.16) is essentially a mathematical statement of the Principle of Optimality for the problem at hand. It indicates that the minimum average value for the functional is achieved by an optimum first control decision followed by an optimal sequence of control decisions which are averaged over all possible states resulting from the first decision. Note that $R(x+\Delta x,\ t+\Delta t)$ has the expansion

$$R(X+\Delta X,\ t+\Delta t) = R(X,t) + \left(\frac{\partial R}{\partial t} + \frac{\partial R^T}{\partial X} \dot{X} \right) \Delta t + \dot{X}^T \frac{\partial^2 R}{\partial X^2} \dot{X} \frac{\Delta t^2}{2} + \cdots$$

Using the expression for $\dot{x}$ and ξ in (2.5.9) and (2.5.10) and taking the expected value of $R(x+\Delta x, t+\Delta t)$ over $\xi(\tau_1)$ for $t \leq \tau_1 \leq t+\Delta t$ provides

$$\underset{\substack{\xi(\tau_i)\\ t\le\tau_i\le t+\Delta t}}{E}\left\{R(x+\Delta x,\, t+\Delta t)\right\} = R(x,t) + \left\{\left(\frac{\partial R}{\partial x}\right)^T (Ax+Gu) + \frac{tr}{2}\left(\Sigma(t)\,\frac{\partial^2 R}{\partial x^2}\right)\right\}\Delta t + O(\Delta t) \qquad (2.5.17)$$

where tr denotes the trace of the matrix $\Sigma(t)\,\frac{\partial^2 R}{\partial x^2}$. This last term is derived from the expected value of the quantity

$$\underset{\substack{\xi(\tau_i)\\ t\le\tau_i\le t+\Delta t}}{E}\left\{\dot{x}^T\,\frac{\partial^2 R}{\partial x^2}\,\dot{x}\,\Delta t^2\right\} = \underset{\substack{\xi(\tau_i)\\ t\le\tau_i\le t+\Delta t}}{E}\left\{tr\left[\dot{x}\,\dot{x}^T\,\frac{\partial^2 R}{\partial x^2}\right]\Delta t^2\right\} \qquad (2.5.18)$$

The Dirac delta appearing in the variance expression for ξ in Eq. (2.5.10) causes this term to reduce to first order in Δt. Substitution of Eq. (2.5.18) into (2.5.16) and taking the limit as Δt goes to zero provides the final result

$$0 = \underset{u(t)}{MIN}\left\{x^T Q_1 x + u^T Q_2 u + \frac{\partial R}{\partial t} + \left(\frac{\partial R}{\partial x}\right)^T (Ax+Gu) + \frac{tr}{2}\left[\Sigma\,\frac{\partial^2 R}{\partial x^2}\right]\right\} \qquad (2.5.19)$$

The boundary condition on $R(x,t)$ is easily developed from the definition of R given in (2.5.13). Thus,

$$R(x_f, t_f) = x_f^T \Lambda x_f$$

or alternately

$$R(x, t_f) = x^T \Lambda x \qquad (2.5.20)$$

Eq. (2.5.19) is similar to that developed in the deterministic case [see Equation (2.4.113)] , the only difference being the appearance of the term $\frac{tr}{2}\left(\Sigma\,\frac{\partial^2 R}{\partial x^2}\right)$. This, however, is a major difference. While the Bellman equation is a first order partial differential equation and can be solved in a straightforward manner using the method of characteristics, this equation is a second order equation of the diffusion type. As a general rule, diffusion processes are rather difficult to solve. Fortunately, Eq. (2.5.19) solves rather easily.

Performing the minimization indicated in (2.5.19) { i.e., setting the derivative with respect to u to zero } provides

$$2u^TQ_2 + \left(\frac{\partial R}{\partial x}\right)^T G = 0$$

which can be rewritten as

$$u = -\frac{1}{2} Q_2^{-1} G^T \frac{\partial R}{\partial x} \qquad (2.5.21)$$

Substituting this expression in (2.5.19) yields

$$x^TQ_1x - \frac{1}{4}\left(\frac{\partial R}{\partial x}\right)^T G Q_2^{-1} G^T \frac{\partial R}{\partial x} + \left(\frac{\partial R}{\partial x}\right)^T A x + \frac{\partial R}{\partial t} + \frac{tr}{2}\left(\Sigma \; \frac{\partial^2 R}{\partial x^2}\right) = 0 \qquad (2.5.22)$$

It can be shown that (2.5.22) has a solution of the form

$$R(x,t) = x^T S(t) x + \beta(t) \qquad (2.5.23)$$

where $S(t)$ is an nxn time dependent symmetric matrix and $\beta(t)$ is a time varying scalar. This expression will satisfy the boundary condition of Eq. (2.5.20) provided

$$\begin{aligned} S(t_f) &= \Lambda \\ \beta(t_f) &= 0 \end{aligned} \qquad (2.5.24)$$

Also, by substituting Eq. (2.5.23) into (2.5.22), it follows that the proposed R function will satisfy Eq. (2.5.22) if

$$\dot{S} + Q_1 - SGQ_2^{-1}G^TS + SA + A^TS = 0 \qquad (2.5.25)$$

$$\dot{\beta} + tr(\Sigma S) = 0 \qquad (2.5.26)$$

Collecting results, the solution is achieved by integrating Eqs. (2.5.25) and (2.5.26) backwards from t_f to t_0 and using the boundary conditions in (2.5.24). From (2.5.21) and (2.5.23), the optimal control action is then determined from

$$u = -Q_2^{-1} G^T S x \qquad (2.5.27)$$

The minimum value of the performance index is given by

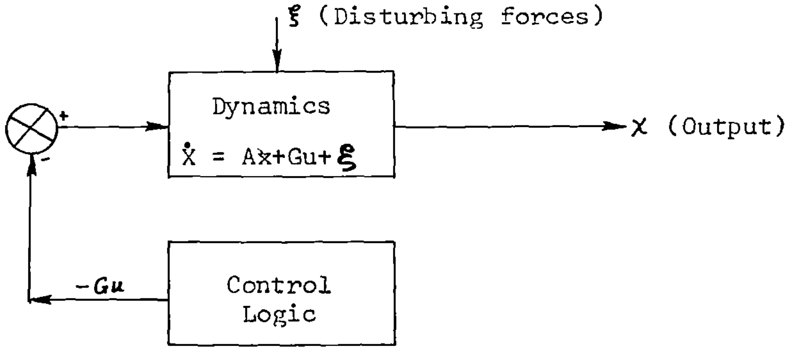
$$\mathit{Min}\ E\{J\} = R(\hat{x}_0, t_0) = \hat{x}_0^T S(t_0)\hat{x}_0 + \beta(t_0) \qquad (2.5.28)$$

Two observations concerning the control law of Eq. (2.5.27) can be made. First the control law in the stochastic case in identical to the control law for the deterministic case in which the random variable ξ in Eq. (2.5.9) is set to zero and the criterion of minimizing the expected value of J is replaced by minimizing J itself. Dreyfus in Reference (2.4.1) refers to this property as "certainty equivalence" and points out that it occurs infrequently in stochastic problems. However, a non-linear example of certainty equivalence is given in Reference (2.5.1). A second observation is that the control law is an explicit function of the state, the actual system output. To implement this law, the state must be observed at each instant of time, a requirement that can be met only in the perfectly observable case; that is, the control law could not be implemented if something less than perfect knowledge of the system output were available. This point clearly demonstrates that the optimal control law in a stochastic problem is very much a function of the type of observational data being collected.

For the treatment of additional stochastic problems in which perfect observability is assumed, the reader is referred to References (2.1.3), (2.4.1), (2.5.1) and (2.5.2).

2.5.2.2 Perfectly Inobservable Case

In this case, it is assumed that no knowledge of the output of the system is available for $t > t_0$. A diagram of this type of controller is given in Sketch (2.5.2) below.



Sketch (2.5.2)

Note that since there is no feedback loop, the optimal control can be computed only as a function of time and whatever knowledge is available concerning the initial state x_0 .

Again the state equations are

$$\dot{x} = Ax + Gu + \xi \tag{2.5.29}$$

with ξ a Gaussian white noise with

$$\begin{aligned} E(\xi) &= 0 \\ E\{\xi(t)\,\xi^T(\tau)\} &= \Sigma(t)\,\delta(t-\tau) \end{aligned} \tag{2.5.30}$$

The initial state x_0 is assumed to be a Gaussian random variable with mean $\hat{x}_0$ and covariance V_0 , that is

$$\begin{aligned} E(x_0) &= \hat{x}_0 \\ E\left\{(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right\} &= V_0 \end{aligned} \tag{2.5.31}$$

The performance index is again

$$E\{J\} = E\left\{\int_{t_0}^{t_f}(x^TQ_1x + u^TQ_2u)dt + x_f^T\Lambda x_f\right\} \tag{2.5.32}$$

There are two means available for evaluating the expected value of the functional J . First, the state equation can be solved to develop the function relationship between x and the random variables ξ and x_0 . Following this development, the expected value of J can be computed by using the appropriate density function for ξ and x_0 . A second approach is to develop the probability density function for x , $p(x,t)$, given the densities of x_0 and ξ . This approach is more direct and will be used here since it leads to the rather simple relationship

$$\begin{aligned} E(J) &= E\left\{\int_{t_0}^{t_f}(x^TQ_1x + u^TQ_2u)dt + x_f^T\Lambda x_f\right\} \\ &= \int_{t_0}^{t_f}\left\{\int_{-\infty}^{\infty} p(x,t)\left[x^TQ_1x + u^TQ_2u\right]dx\right\}dt + \int_{-\infty}^{\infty} p(x,t_f)\,x^T\Lambda x\,dx \end{aligned} \tag{2.5.33}$$

from which the optimal control can be readily determined.

Since the state equation is linear and since ξ and x_0 are Gaussian, it follows that the random process $x(t)$ is also Gaussian*. The mean and

* See Reference (2.5.3) for the demonstration that linear transformation on Gaussian random processes lead to Gaussian random processes.

variance characterizing $x(t)$ can be evaluated either from the Fokker*-Planck equation (also called the forward Kolmogorov equation) or by direct calculation as follows. Let $\hat{x}$ denote the mean of x and let V denote the covariance. Thus

$$\hat{x}(t) = E(x(t)) = \int_{-\infty}^{\infty} x\,p(x,t)\,dx \qquad (2.5.34)$$

$$V(t) = E\left\{\left(x(t)-\hat{x}(t)\right)\left(\dot{x}(t)-\hat{x}(t)\right)^T\right\} = \int_{-\infty}^{\infty}(x-\hat{x}(t))(x-\hat{x}(t))^T p(x,t)\,dx \qquad (2.5.35)$$

Differentiating these two equations, and using Eq. (2.5.29) yields

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Gu \\ \dot{V} &= AV + VA^T + \Sigma\end{aligned} \qquad (2.5.36)$$

while from Eq. (2.5.31), the boundary conditions

$$\begin{aligned}\hat{x}(t_0) &= \hat{x}_0 \\ V(t_0) &= V_0\end{aligned} \qquad (2.5.37)$$

must hold. Thus, the density for x is

$$p(x,t) = \frac{1}{(2\pi)^{n/2}\sqrt{|V(t)|}}\, e^{-\frac{1}{2}\left\{(x-\hat{x}(t))\,V^{-1}(t)(x-\hat{x}(t))\right\}} \qquad (2.5.38)$$

with $\dot{x}$ and $\dot{V}$ satisfying Eqs. (2.5.36) and (2.5.37).

Proceeding with the optimization problem, let

$$\bar{x} = x - \hat{x} \qquad (2.5.39)$$

Note that

$$E(\bar{x}) = E(\bar{x}\hat{x}^T) = 0 \qquad (2.5.40)$$

and from Eqs. (2.5.29) and (2.5.36) that

$$\dot{\bar{x}} = A\bar{x} + \xi \qquad (2.5.41)$$

* See Reference (2.5.4)

thus, substituting the expression

$$x = \bar{x} + \hat{x}$$

into Eq. (2.5.32) and making use of (2.5.40) reduces the optimizing criterion to

$$E\{J\} = \int_{t_0}^{t_f} (\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)\,dt + \hat{x}_f^T \Lambda \hat{x} + E\left\{\int_{t_0}^{t_f} (\bar{x}^T Q_1 \bar{x})\,dt + \bar{x}_f^T \Lambda \bar{x}_f\right\} \qquad (2.5.42)$$

Furthermore, since

$$\bar{x}^T Q_1 \bar{x} = tr\,\{\bar{x}\,\bar{x}^T Q_1\}$$

where tr denotes the trace operator, and since

$$E(\bar{x}\,\bar{x}^T) = E\,\{(x-\hat{x})(x-\hat{x})^T\} = V$$

Eq. (2.5.42) can be rewritten as

$$E(J) = \int_{t_0}^{t_f} (\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)\,dt + \hat{x}_f^T \Lambda \hat{x}_f + \int_{t_0}^{t_f} tr(VQ_1)\,dt + tr(V_f \Lambda) \qquad (2.5.43)$$

Now since the covariance V does not depend on the control u [see Eq. (2.5.36)] , it follows that minimizing the expected value of J is equivalent to minimizing the first two terms on the right hand side of (2.5.43). Thus, the optimal control is that control which minimizes the functional $\bar{J}$ where

$$\bar{J} = \int_{t_0}^{t_f} (\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)\,dt + \hat{x}_f^T \Lambda \hat{x}_f \qquad (2.5.44)$$

subject to the conditions

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Gu \\ \hat{x}(t_0) &= \hat{x}_0 \end{aligned} \qquad (2.5.45)$$

This reduced problem is deterministic and can be solved using the methods of Section (2.4.10).

Letting $R(\hat{x},t)$ denote the minimum value of $\bar{J}$ for the trajectory starting at the point $(\hat{x},t)$, it follows from Dynamic Programming that $R(\hat{x},t)$ satisfies the Bellman equation and boundary condition given by

$$0 = \underset{u(t)}{MIN} \left\{ \hat{x}^T Q_1 \hat{x} + u^T Q_2 u + \left(\frac{\partial R}{\partial \hat{x}}\right)^T (A\hat{x} + Gu) + \frac{\partial R}{\partial t} \right\} \tag{2.5.46}$$

$$R(\hat{x}, t_f) = \hat{x}^T \Lambda \hat{x} \tag{2.5.47}$$

The solution takes the form

$$R(\hat{x}, t) = \hat{x}^T S(t) \hat{x} \tag{2.5.48}$$

with S satisfying

$$\dot{S} + Q_1 - S'GQ_2^{-1}G^T S + SA + A^T S = 0 \tag{2.5.49}$$

$$S(t_f) = \Lambda \tag{2.5.50}$$

and the optimal control given by

$$u = -\frac{1}{2} Q_2^{-1} G^T \frac{\partial R}{\partial \hat{x}} = -Q_2^{-1} G^T S \hat{x} \tag{2.5.51}$$

Thus,

$$R(\hat{x}_0, t_0) = \hat{x}_0^T S(t_0) \hat{x}_0$$

Substitution of this expression into Eq. (2.5.43) now gives the minimum expected value of J as

$$E(J) = \hat{x}_0^T S(t_0) \hat{x}_0 + \int_{t_0}^{t_f} tr(VQ_1) dt + tr(V_f \Lambda) \tag{2.5.52}$$

Two comments on the form of the control law given in Eq. (2.5.51) are in order. First, it is the same form as that which would result for the deterministic problem, but with the state x replaced by the expected value of the state. This result, while interesting, is not surprising in view of the similar findings for the perfectly observable case of the preceding section. Secondly, the variable $\hat{x}$ on which the control depends is a function only of the expected initial state and time. No feedback information is used in the computation of $\hat{x}$, a result consistent with the perfectly inobservable quality of the system.

It is interesting to compare the value of the performance index for the perfectly observable and perfectly inobservable cases. Since more information is available and is used in the perfectly observable case, the performance index for the perfectly observable case is the smaller of the two. Let the covariance of the initial state, V_0 is Eq. (2.5.31) be taken as zero. Hence, the initial state is known exactly in both the perfectly observable and inobservable cases and is given by

$$x = \hat{x}_0 \quad \text{AT} \quad t = t_0$$

From Eq. (2.5.28) and the expression for β in (2.5.24) and (2.5.26), the performance index in the perfectly observable case is given by

$$E\{J\}_{\text{OBSERVABLE}} = \hat{x}_0^T S(t_0)\, \hat{x}_0 + \int_{t_0}^{t_f} tr(\Sigma S)\, dt \tag{2.5.53A}$$

while from Equation (2.5.52)

$$E\{J\}_{\text{IN OBSERVABLE}} = \hat{x}_0^T S(t_0)\, \hat{x}_0 + \int_{t_0}^{t_f} tr(VQ_1)\, dt + tr(V_f \Lambda) \tag{2.5.53B}$$

Since the matrix S satisfies the same equation and boundary condition in both cases, it follows that

$$E\{J\}_{\text{INOBSERVABLE}} - E\{J\}_{\text{OBSERVABLE}} = \int_{t_0}^{t_f} tr(VQ_1 - \Sigma S)\, dt + tr(V_f \Lambda) \tag{2.5.54}$$

This difference can be shown to be positive by noting that S and V, from Eqs. (2.5.49) and (2.5.36), satisfy

$$\frac{d}{dt} tr\{VS\} = tr\{\Sigma S - VQ_1 + VSGQ_2^{-1}G^T S\} \tag{2.5.55}$$

Integrating this expression with the condition that $V(t_0) = 0$ and combining with (2.5.54) yields

$$E\{J\}_{\text{INOBSERVABLE}} - E\{J\}_{\text{OBSERVABLE}} = \int_{t_0}^{t_f} tr\{VSGQ_2^{-1}G^T S\}\, dt \tag{2.5.56}$$

Since V is positive definite for $t > t_0$ and Q_2 is positive definite, it follows that the right hand side of (2.5.56) is positive and that the performance index in the perfectly inobservable case is always larger than that for the perfectly observable case.

2.5.2.3 Partially Observable Case

The partially observable case differs from the preceding cases in that some knowledge of the system output is available at each instant, but the knowledge is imperfect due to the presence of noise and the possibility that only a part of the output can be measured. The system is again given by

$$\dot{x} = Ax + Gu + \xi \qquad (2.5.57)$$

with x_0 a Gaussian variable with mean and covariance given by

$$E(x_0) = \hat{x}_0$$
$$E\{(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\} = V_0 \qquad (2.5.58)$$

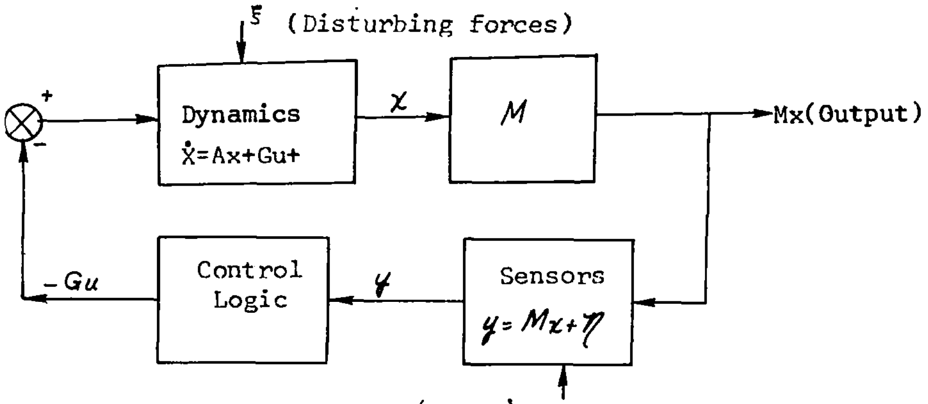
The fact that some data are being collected (i.e., some observations are being made) is represented by the equation

$$y = Mx + \eta$$

where y is the m dimensional observation vector $(m \leq n)$, M is an $m\text{x}n$ time varying matrix and η is a white Gaussian noise with zero mean and variance $\Gamma(t)$; that is,

$$E(\eta) = 0$$
$$E(\eta(t)\,\eta^T(\tau)) = \Gamma(t)\,\delta(t-\tau) \qquad (2.5.59)$$

The physical situation is pictured in the sketch below.



Sketch (2.5.3) η (sensor noise)

Let $\mathcal{Y}(t)$ denote the observations that are made on the interval (t_o, t), that is

$$\mathcal{Y}(t) = \{ y(t_i) : \quad t_o < t_i < t \} \tag{2.5.60}$$

The expected value of the functional J can be written as

$$\begin{aligned} E\{J\} &= E\left\{ \int_{t_o}^{t_f} (x^T Q_1 x + u^T Q_2 u) dt + x_f^T \Lambda x_f \right. \\ &= \int_{t_o}^{t_f} \left\{ \iint_{-\infty}^{\infty} p(x, \mathcal{Y}, t)(x^T Q_1 x + u^T Q_2 u) dx\, d\mathcal{Y} \right\} dt \\ &\quad + \iint_{-\infty}^{\infty} p(x, \mathcal{Y}, t_f) x^T \Lambda x \, dx\, d\mathcal{Y} \end{aligned} \tag{2.5.61}$$

where $p(x, \mathcal{Y}, t)$ is the joint density function for x and $\mathcal{Y}$ as developed from the density functions for ξ, η and x_o. The variable $\mathcal{Y}$ must be included since the control $u(t)$ will depend on, and vary with, the observations. Now the density $p(x, \mathcal{Y}, t)$ can be expressed as

$$p(x, \mathcal{Y}, t) = p(x, t/\mathcal{Y}) p(\mathcal{Y})$$

where $p(x, t/\mathcal{Y})$ is the probability density of x conditioned on $\mathcal{Y}$. Also, the expected value of some function $K(x, \mathcal{Y})$ can be written as

$$\begin{aligned} E\{K(x, \mathcal{Y})\} &= \iint_{-\infty}^{\infty} K(x, \mathcal{Y}) p(x, \mathcal{Y}, t) dx\, d\mathcal{Y} \\ &= \int_{-\infty}^{\infty} p(\mathcal{Y}) \left\{ \int_{-\infty}^{\infty} K(x, \mathcal{Y})\, p(x, t/\mathcal{Y}) dx \right\} d\mathcal{Y} \\ &= \underset{\mathcal{Y}}{E} \left\{ E\left[K(x, \mathcal{Y})/\mathcal{Y} \right] \right\} \end{aligned}$$

Using this result, the performance index in Eq. (2.5.61) can be written

$$E\{J\} = \underset{\mathcal{Y}}{E}\left\{E(J/\mathcal{Y})\right\} = \underset{\mathcal{Y}}{E}\left\{E\left[\int_{t_0}^{t_f}(x^TQ_1x+u^TQ_2u)dt+x_f^T\Lambda x_f/\mathcal{Y}\right]\right\} \qquad (2.5.62)$$

where the first expectation on the right is taken with respect to $\mathcal{Y}$ and the second with respect to x conditioned on $\mathcal{Y}$.

It is well known that the conditional density $p(x, t/\mathcal{Y})$ for the problem under consideration is Gaussian with mean $\hat{x}$ and covariance V satisfying the differential equations and boundary conditions

$$\dot{\hat{x}} = A\hat{x} + Gu + VM^T\Gamma^{-1}(y - M\hat{x}) \qquad (2.5.63A)$$

$$\dot{V} = AV + VA^T + \Sigma - VM^T\Gamma^{-1}MV \qquad (2.5.63B)$$

$$\begin{aligned} x(t_0) &= \hat{x}_0 \\ V(t_0) &= V_0 \end{aligned} \qquad (2.5.64)$$

These results can be derived either directly by differentiating and reworking the defining expression

$$\hat{x} = E\left\{x(t)/\mathcal{Y}(t)\right\} \qquad (2.5.65A)$$

$$V = E\left\{(x-\hat{x}(t))(x-\hat{x}(t))^T/\mathcal{Y}(t)\right\} \qquad (2.5.65B)$$

as in Ref. (2.5.5) or through the modified Fokker-Planck equation as developed in Ref. (2.5.6). As in the perfectly inobservable case, let

$$x = \hat{x} + \bar{x} \qquad (2.5.66)$$

with $\bar{x}$ satisfying the differential equation

$$\dot{\bar{x}} = A\bar{x} + \xi - VM^T\Gamma^{-1}\{M\bar{x} + \eta\} \qquad (2.5.67)$$

and with

$$E\{\bar{x}/\mathcal{Y}\} = E\{\bar{x}\,\hat{x}^T/\mathcal{Y}\} = 0 \qquad (2.5.68)$$

Then, substituting (2.5.66) into (2.5.62) and making use of (2.5.68) reduces the performance index to the form

$$E(J) = \underset{y}{E}\left\{\int_{t_0}^{t_f}(\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)dt + \hat{x}_f^T \Lambda \hat{x}_f\right\} + \underset{y}{E}\left\{E\left[\int_{t_0}^{t_f}\bar{x}^T Q_1 \bar{x}\,dt + \bar{x}_f^T \Lambda \bar{x}_f / y\right]\right\} \tag{2.5.69}$$

Note that the terms in the second bracket do not depend on the control u or on the observations y since the variable $\bar{x}$ is not a function of u or y. In fact, these terms can be immediately evaluated using x as defined in Eqs. (2.5.65B) and (2.5.68). The final result for the performance index is

$$E(J) = \underset{y}{E}\left\{\int_{t_0}^{t_f}(\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)dt + \hat{x}_f^T \Lambda \hat{x}_f\right\} + tr(V_f \Lambda) + \int_{t_0}^{t_f} tr(V Q_1)dt \tag{2.5.70}$$

From this equation, the optimal control is that control which minimizes the expression

$$\underset{y}{E}\left\{\int_{t_0}^{t_f}(\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)dt + \hat{x}_f^T \Lambda \hat{x}_f\right\} = \underset{\substack{y(\tau) \\ t_0 \le \tau \le t_f}}{E}\left\{\int_{t_0}^{t_f}(\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)dt + \hat{x}_f^T \Lambda \hat{x}_f\right\}$$

Let

$$R(\hat{x}, t) = \underset{\substack{u(\tau) \\ t \le \tau \le t_f}}{Min}\ \underset{y(\tau)}{E}\left\{\int_{t_0}^{t_f}(\hat{x}^T Q_1 \hat{x}, u^T Q_2 u)dt + \hat{x}_f^T \Lambda \hat{x}_f\right\} \tag{2.5.71}$$

for the optimal solution starting at the point $(\hat{x}, t)$. Then $R(\hat{x}, t)$ satisfies the boundary condition $R(\hat{x}, t_f) = \hat{x}^T \Lambda \hat{x}$. Using the Dynamic Programming approach

$$R(\hat{x}, t) = \underset{\substack{u(\tau_1) \\ t \le \tau_1 \le t+\Delta t}}{Min}\ \underset{y(\tau_1)}{E}\left\{\underset{\substack{u(\tau_2) \\ t+\Delta t \le \tau_2 \le t_f}}{Min}\ \underset{y(\tau_2)}{E}\left[\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)\Delta t + O(\Delta t) + \int_{t+\Delta t}^{t_f}(\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)dt + \hat{x}_f^T \Lambda \hat{x}_f\right]\right\}$$

$$= \underset{\substack{u(\tau_1) \\ t \le \tau_1 \le t+\Delta t}}{Min}\left\{(\hat{x}^T Q_1 \hat{x} + u^T Q_2 u)\Delta t + O(\Delta t) + \underset{\substack{y(\tau_1) \\ t \le \tau_1 \le t+\Delta t}}{E}\left\{R(\hat{x} + \Delta\hat{x}, t + \Delta t)\right\}\right\}$$

Taking the limit and using the expression for $\dot{\hat{x}}$ in Eq. (2.5.63A) provides

$$0 = \underset{u(t)}{Min}\left\{\hat{x}^T Q_1 \hat{x} + \frac{\partial R}{\partial t} + \left(\frac{\partial R}{\partial \hat{x}}\right)^T (A\hat{x}+Gu) + \frac{1}{2}\, tr\left\{\frac{\partial^2 R}{\partial \hat{x}^2} VM^T\Gamma^{-1}MV\right\}\right\} \qquad (2.5.72)$$

where the second order term arises in exactly the same manner as in Eqs. (2.5.18) and (2.5.19) for the perfectly observable case. Performing the minimization as indicated in (2.5.72) provides

$$u = -\frac{1}{2} Q_2^{-1} G^T \frac{\partial R}{\partial \hat{x}} \qquad (2.5.73)$$

Thus, substitution of this expression for u back into (2.5.72) yields

$$\hat{x}^T Q_1 \hat{x} - \frac{1}{4}\left(\frac{\partial R}{\partial \hat{x}}\right)^T G Q_2^{-1} G^T \frac{\partial R}{\partial \hat{x}} + \left(\frac{\partial R}{\partial \hat{x}}\right)^T A\hat{x} + \frac{\partial R}{\partial t} + \frac{tr}{2}\left(\frac{\partial^2 R}{\partial \hat{x}^2} VM^T\Gamma^{-1}MV\right) = 0 \qquad (2.5.74A)$$

Equation (2.5.74A) is essentially the same as the diffusion equation which resulted in the perfectly observable case and has a similar solution. Letting

$$R(\hat{x},t) = \hat{x}^T S(t)\hat{x} + \beta(t) \qquad (2.5.74B)$$

and substituting this expression into (2.5.74A) provides

$$\begin{aligned} &\dot{S} + Q_1 - SGQ_2^{-1}G^T S + SA + A^T S = 0 \\ &\dot{\beta} + tr\{SVM^T\Gamma^{-1}MV\} = 0 \end{aligned} \qquad (2.5.75)$$

The boundary condition on R is satisfied for

$$\begin{aligned} S(t_f) &= \Lambda \\ \beta(t_f) &= 0 \end{aligned} \qquad (2.5.76)$$

while the optimal control takes the form

$$u = -Q_2^{-1} G^T S \hat{x}$$

Note that the optimal control is a function of the estimate of the state which in turn depends on the observations as indicated in Eq. (2.5.62). Using Eqs. (2.5.74B) and (2.5.70), the minimum value for the performance index is

$$E\{J\} = \hat{x}_0^T \mathcal{S}(t_0)\hat{x}_0 + \beta_0 + tr(V_f \Lambda) + \int_{t_0}^{t_f} tr(VQ_1)dt \qquad (2.5.77)$$

The minimum performance index for the partially observable case falls somewhere between that for the perfectly observable and that for the perfectly inobservable system; that is

$$E(J)_{OBSERVABLE} < E(J)_{\substack{PARTIALLY\\OBSERVABLE}} < E(J)_{INOBSERVABLE} \qquad (2.5.78)$$

This statement can be shown by considering the case where the initial state is known (i.e., the variance for the initial state is zero, $V_0 = 0$). Since the matrix $\mathcal{S}$ is the same in all three cases, it follows from Eq. (2.5.53A), and the definition of β in (2.5.75) and (2.5.76) that

$$E(J)_{\substack{PARTIALLY\\OBSERVABLE}} - E(J)_{OBSERVABLE} = \int_{t_0}^{t_f} tr\left\{SVM^T\Gamma^{-1}MV - \mathcal{S}\Sigma + VQ_1\right\}dt + tr(V_f\Lambda) \quad (2.5.79)$$

But from the definitions of V and $\mathcal{S}$ in (2.5.63B) and (2.5.75)

$$\frac{d}{dt}\; tr(SV)\} = tr\left\{(SGQ_2^{-1}G^T S - Q_1)V + S(\Sigma - VM^T\Gamma^{-1}MV)\right\} \qquad (2.5.80)$$

Thus, integrating with $V_0 = 0$ and substituting into (2.5.79) yields

$$E(J)_{\substack{PARTIALLY\\OBSERVABLE}} - E(J)_{OBSERVABLE} = \int_{t_0}^{t_f} tr(SGQ_2^{-1}G^T S\, V)dt$$

Since V is positive definite for $t > t_0$ and Q_2^{-1} is positive, one half of the inequality in (2.5.78) is established. To establish the other half, note that from Eqs. (2.5.33B) and (2.5.77)

$$E(J)_{\substack{INOBSERVABLE}} - E(J)_{\substack{PARTIALLY\\OBSERVABLE}} = \left\{\int_{t_0}^{t_f} tr(VQ_1)dt + tr(V_f\Lambda)\right\}_{INOBSERVABLE} - \left\{\int_{t_0}^{t_f} tr(VQ_1)dt + tr(V_f\Lambda)\right\}_{\substack{PARTIALLY\\OBSERVABLE}} \qquad (2.5.81)$$

Note, also, that the variance functions, V , are different in the two cases. However, making use of Eq. (2.5.55) for the inobservable case and Eq. (2.5.80) for the partially observable case reduces (2.5.81) to

$$E(J)_{INOBSERVABLE} - E(J)_{\substack{PARTIALLY\\OBSERVABLE}} = \left\{\int_{t_0}^{t_f} tr(VSGQ_2^{-1}G^TS)dt\right\}_{INOBSERVABLE} - \left\{\int_{t_0}^{t_f} tr(VSGQ_2^{-1}G^TS)\,dt\right\}_{\substack{PARTIALLY\\OBSERVABLE}}$$

Now, since V in the partially observable case is less than V in the perfectly inobservable case (i.e., the observations y reduce the variance in the estimate of x) the inequality

$$E(J)_{INOBSERVABLE} - E(J)_{\substack{PARTIALLY\\OBSERVABLE}} > 0 \qquad (2.5.82)$$

is established.

2.5.2.4 Discussion

In all three cases, perfectly observable, perfectly inobservable and partially observable, the form of the optimal control action is the same. Specifically, the optimal control is a linear function of either the state, or the expected value of the state, with the proportionally factor being the same for each case. This is a rather striking similarity, but one which appears to hold only for the linear - quadratic cost problem.

Note that the performance index, which is to be minimized, decreases as a quality of the observational data increases. The two limiting cases, the perfectly observable and perfectly inobservable systems, provide lower and upper bounds, respectively, for the performance index value which can be achieved by a partially observable system.

The analysis through which the optimal control action is determined consists of a rather straightforward application of Dynamic Programming. While it is not difficult to formulate the perfectly inobservable problem using other methods, there appears to be no way of treating the perfectly observable or partially observable case using the Variational Calculus. Hence, the stochastic optimization problem is one area where Dynamic Programming is not an alternate procedure, but frequently the only procedure available for conducting the analysis.

2.5.3 The Treatment of Terminal Constraints

In the preceding section, the optimal control action was developed under the condition that no constraints were placed in the terminal state X_f . In this section, a slightly modified version of the linear-quadratic cost problem will be analyzed in which the expected value of the terminal state is required to satisfy one or more conditions. Specifically, the system is again governed by the state equation

$$\dot{x} = Ax + Gu + \xi \tag{2.5.83}$$

with ξ Gaussian white noise satisfying

$$E\left\{\xi(t)\,\xi(\tau)\right\} = \Sigma(t)\,\delta(t-\tau) \tag{2.5.84}$$

This time, however, the performance index takes the form

$$E\left\{J_1\right\} = \left\{\int_{t_0}^{t_f}(x^TQ_1x + u^TQ_2u)\,dt\right\} \tag{2.5.85}$$

Note that no measure of the terminal error is included in $E(J_1)$; that is, the performance index is a sub-case of the previous performance index in which the matrix Λ has been set equal to zero. The reason for this change will become apparent shortly.

Let $z_f = z(t_f)$ denote a p vector which is linearly related to the terminal state through

$$z(t_f) = z_f = Hx_f \tag{2.5.86}$$

where H is a constant $p \times n$ matrix and where $p \le n$. Three different types of terminal constraints will be considered.

(1) $\mathrm{tr}\,E\{z_f z_f^T\} \le c$; c is a scalar (2.5.87A)
(2) $(E\{z_f z_f^T\})_{ii} \le c_i$; $i = 1,p$ (2.5.87B)
(3) $(E\{z_f z_f^T\})_{ii} \le c_i$; $i = 1,p_1;\ p_1 < p$, (2.5.87C)

In the first case, the symbol tr denotes the trace of the matrix $E\{z_f z_f^T\}$. Hence, the sur of the diagonal elements of $E\{z_f z_f^T\}$ is required to be less than or equal to some number C . In the second case, the individual diagonal elements of $E\{z_f z_f^T\}$ that is, $(E\{z_f z_f^T\})_{ii}$, $i = 1,p$) are required to satisfy an inequality condition while in Case (3), only the first p_1 diagonal elements of $E(z_f z_f^T)$ are constrained. To clarify the physical meaning of these constraints, some examples will be considered.

Suppose the state, x, is six dimensional and the matrix H in Equation (2.5.86) is the row vector $H = (1, 0, 0, 0, 0, 0)$ $\{i.e. \; P=1\}$. In this case, $z_f = x_1(t_f)$ and all three constraint conditions takes the form

$$E(x_1(t_f)^2) \leq C$$

For a matrix H possessing the 2 rows

$$H = \begin{pmatrix} 1, & 0, & 0, & 0, & 0, & 0 \\ 0, & 1, & 0, & 0, & 0, & 0 \end{pmatrix}$$

it follows that

$$z_f = \begin{pmatrix} z_{1f} \\ z_{2f} \end{pmatrix} = \begin{pmatrix} x_1(t_f) \\ x_2(t_f) \end{pmatrix}$$

with

$$E\{z_f z_f^T\} = \begin{pmatrix} E(x_{1f}^2) & E(x_{1f} x_{2f}) \\ E(x_{1f} x_{2f}) & E(x_{2f}^2) \end{pmatrix}$$

In this case, the terminal constraint of Equation (2.5.87A) takes the form

$$E(x_{1f}^2) + E(x_{2f}^2) \leq C$$

while (2.5.87B) reduces to the two conditions

$$E(x_{1f}^2) \leq C_1$$

$$E(x_{2f}^2) \leq C_2$$

As a third example, suppose H is the identity matrix. Then Equation (2.5.87A) states that

$$E\{x_{1f}^2 + x_{2f}^2 + \cdots + x_{6f}^2\} \leq C$$

while Equation (2.5.87B) requires

$$E(x_{1_f}^2) \leq c_1$$
$$E(x_{2_f}^2) \leq c_2$$
$$\vdots$$
$$E(x_{6_f}^2) \leq c_6$$

Alternately, if the constraint of Equation (2.5.87C) is imposed, then only p_1 of the above conditions must hold where p_1 is some integer less than six.

These three possibilities by no means exhaust the types of H matrices that may be used. Rather, they are introduced simply to indicate the types of physical situations that can be represented by constraints of the form of Equations (2.5.87A) to (2.5.87C). In the following development, it is only required that H be some constant matrix with dimensions less than or equal to n , where n is the number of components in the state vector X .

As a further simplification, it will be assumed that the symmetric matrix Q_1 in the performance index of Equation (2.5.85) can be expressed as

$$Q_1 = \phi^T(t_f, t)\, H^T Q H\, \phi(t_f, t) \tag{2.5.88}$$

where $\phi(t_f, t)$ is the fundamental $n \times n$ matrix solution

$$\frac{d}{dt}\phi(t_f, t) = -\phi(t_f, t)A \tag{2.5.89}$$

$$\phi(t_f, t_f) = I \tag{2.5.90}$$

(In what follows, the symbol ϕ will frequently be used to denote $\phi(t_f, t)$.) Since H is a $p \times n$ matrix, it is necessary that Q be a $p \times p$ symmetric matrix. Also, since Q_1 is positive semi-definite, it follows that Q is also positive semi-definite. The reason for this assumption as to the form of Eq. (2.5.88) will be made clear subsequently and it will be shown that Eq. (2.5.88) is physically consistent with the terminal constraints of Eq. (2.5.87).

Following the usual procedure of the Calculus of Variations, the problem of minimizing the functional $E\{J_1\}$ subject to a terminal constraint on the quantity

$$E\{z_f z_f^T\} \; ; \; z_f = Hx_f$$

is equivalent to minimizing the modified functional $E\{J\}$ where

$$E\{J\} = E\{J_1 + \mathrm{tr}(\Lambda z_f z_f^T)\} \qquad (2.5.91)$$

where Λ is a $p \times p$ constant diagonal matrix of Lagrange multipliers (recall that H is a $p \times n$ matrix), and selected so that the specified terminal variance condition is satisfied. The particular form of the matrix Λ will depend on the particular terminal constraint which is imposed [i.e., Equation (2.5.87A) or (2.5.87B) or (2.5.87C)]. For example, if X is a six dimensional vector and H is the matrix

$$H = \begin{pmatrix} 1, 0, 0, 0, 0, 0 \\ 0, 1, 0, 0, 0, 0 \end{pmatrix}$$

the terminal constraint of Eq. (2.5.87A) becomes

$$E\{x_{1_f}^2 + x_{2_f}^2\} = E\{z_{1_f}^2 + z_{2_f}^2\} \leq c$$

and the quantity to be adjoined to Eq. (2.5.85) to form (2.5.91) is

$$\lambda E\{z_{1_f}^2 + z_{2_f}^2\} = E\{\lambda(z_{1_f}^2 + z_{2_f}^2)\}$$

This form is equivalent to $E\{\mathrm{tr}(\Lambda z_f z_f^T)\}$ provided

$$\Lambda \doteq \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

Alternately, if the terminal constraint of Eq. (2.5.87B) is imposed, the quantity to be adjoined to Eq. (2.5.91) is

$$\lambda_1 E(z_{1_f}^2) + \lambda_2 E(z_{2_f}^2); \; \lambda_1, \lambda_2 \text{ are scalars}$$

which will equal to $E\{tr(\Lambda z_f z_f^T)\}$

$$\Lambda = \begin{pmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{pmatrix}$$

In any event, whatever the form of the matrix H, if the terminal constraint is to satisfy one of the conditions in Eq. (2.5.87), then the problem can be handled as is indicated in Eq. (2.5.91). Using the definition in Eq. (2.5.88) and noting that

$$tr(\Lambda z_f z_f^T) = z_f^T \Lambda z_f$$

the performance index can be written as

$$E(J) - E\left\{\int_{t_0}^{t_f}\left[x^T(\phi^T H^T Q H \phi)x + u^T Q_2 u\right] dt + z_f^T \Lambda z_f\right\} \quad (2.5.92)$$

One further simplification is necessary before proceeding with the optimization problem. Let

$$z(t) = z = H\phi(t_f, t)x = H\phi x \quad (2.5.93)$$

Thus, differentiation of this expression with respect to time and using Eqs. (2.5.83) and (2.5.89) provides

$$\dot{z} = H\phi G u + H\phi\xi \quad (2.5.94)$$

with the boundary condition

$$z_0 = z(t_0) = H\phi(t_f, t_0)x_0 \quad (2.5.95)$$

Now, since x_0 is a Gaussian random variable with mean $\hat{x}_0$ and covariance V_0 it follows from (2.5.95) that z_0 is Gaussian with mean and covariance given by

$$\begin{aligned} E(z_0) &= \hat{z}_0 = H\phi\hat{x}_0 \\ E\{(z_0 - \hat{z}_0)(z - z_0)^T\} &= H\phi V_0 \phi^T H^T \end{aligned} \quad (2.5.96)$$

The developments in the preceding paragraphs, while algebraically complex, considerably simplify the terminal constraint problem. Substituting the definition of Eq. (2.5.93) into the performance index of (2.5.92) provides

$$E\left\{J\right\} = E\left\{\int_{t_o}^{t_f} (z^T Q z + u^T Q_2 u)\, dt + z_f^T \Lambda z_f \right\} = \mathrm{MIN} \qquad (2.5.97)$$

The problem is now one of selecting the control u to minimize $E\left\{J\right\}$ subject to the new state equation

$$\dot{z} = H\phi G u + H\phi \xi \qquad (2.5.98)$$

and where z_o is a Gaussian random variable given by Eq. (2.5.96). The elements of the diagonal matrix Λ are to be selected so that the particular terminal constraint specified by one of the equations in (2.5.87) is satisfied. The number of independent or free diagonal elements in Λ is equal to the number of constraints contained in Eq. (2.5.87). For example, if Eq. (2.5.87A) is imposed, (i.e., one constraint) then all the diagonal elements of Λ are equal with their particular value chosen so that (2.5.87A) is satisfied. If Eq. (2.5.87B) is imposed, then the first p_1 diagonal element are independent and the remaining $p - p_1$ are zero.

Since the form of the expectation operator in the performance index depends on the type of observations taken, the perfectly observable, perfectly inobservable and partially observable case must be treated separately. This treatment follows in the next three sections.

2.5.3.1 Perfectly Observable Case

In the perfectly observable case, perfect knowledge of the state x is available at each instant of time. Since z and x are related by the deterministic transformation of Eq. (2.5.93), the vector z is also known at each instant. Hence, the problem is one of minimizing $E\,(J)$ where

$$E\,(J) = E_{\xi}\left\{\int_{t_o}^{t_f} (z^T Q z + u^T Q_2 u)\, dt + z_f^T \Lambda z_f \right\} \qquad (2.5.99)$$

subject to the differential equation

$$\dot{z} = H\phi G u + H\phi \xi \qquad (2.5.100)$$

It is assumed that z_o is known initially, or alternately, that z_o is a Gaussian variable with mean $\hat{z}_o$ and variance zero.

This problem is the same as that treated in Section (2.5.2.1) except that Λ is not known; rather, this matrix must be selected to satisfy a terminal condition. However, the analysis is essentially the same once Λ

is determined.

Let $R(z,t)$ be given by

$$R(z,t) = \underset{\substack{u(\tau)\ \xi(\tau) \\ t \le \tau \le t_f}}{\mathrm{MIN}\ E} \left\{ \int_t^{t_f} (z^T Q z + u^T Q_2 u)\, dt + z_f^T \Lambda z_f \right\} \qquad (2.5.101)$$

then proceeding exactly as in Section (2.5.2.1)

$$R(z,t) = \underset{\substack{u(\tau) \\ t \le \tau \le t+\Delta t}}{\mathrm{MIN}} \left\{ (z^T Q z + u^T Q_2 u)\Delta t + o(\Delta t) + \underset{\substack{\xi \\ t \le \tau \le t+\Delta t}}{E} \left[R(z+\Delta z, t+\Delta t) \right] \right\}$$

Taking the limit as $\Delta t \to 0$ and using Eq. (2.5.100) provides

$$0 = \underset{u(t)}{\mathrm{MIN}} \left\{ z^T Q z + u^T Q_2 u + \frac{\partial R}{\partial t} + \left(\frac{\partial R}{\partial z}\right)^T (H \phi G u) + \frac{\mathrm{tr}}{2} \left(\frac{\partial^2 R}{\partial z^2} H \phi \Sigma \phi^T H^T \right) \right\}$$

with $R(z,t)$ satisfying the terminal condition

$$R(z, t_f) = z^T \Lambda z \qquad (2.5.103)$$

Thus, differentiating Eq. (2.5.102) with respect to u yields the optimal control

$$u = -\frac{1}{2} Q_2^{-1} G^T \phi^T H^T \frac{\partial R}{\partial z} \qquad (2.5.104)$$

and substitution of this expression into (2.5.102) provides

$$0 = z^T Q z - \frac{1}{4} \left(\frac{\partial R}{\partial z}\right)^T H \phi G Q_2^{-1} G^T \phi^T H^T \frac{\partial R}{\partial z} + \frac{\partial R}{\partial t} + \frac{\mathrm{tr}}{2} \left[\frac{\partial^2 R}{\partial z^2} H \phi \Sigma \phi^T H^T \right]$$

This equation has a solution of the form

$$R = z^T S(t) z + \beta(t) \qquad (2.5.105)$$

where S is a $p \times p$ symmetric matrix and β is a scalar satisfying the differential equations

$$\dot{S} + Q - S H \phi G Q_2^{-1} G^T \phi^T H^T S = 0 \qquad (2.5.106A)$$

$$\dot{\beta} + \mathrm{tr}\, (S H \phi \Sigma \phi^T H^T) = 0 \qquad (2.5.106B)$$

and the boundary conditions

$$S(t_f) = \Lambda \qquad \beta(t_f) = 0 \tag{2.5.106C}$$

The optimal control takes the form

$$u = -Q_2^{-1} G^T \phi^T H^T S z \tag{2.5.107}$$

The one remaining consideration is the selection of the matrix Λ so that the terminal constraint of Eq. (2.5.87) is satisfied. This point will be treated next.

Let $\hat{z}$ denote the expected value of z conditioned only on the initial information $z_0 = \hat{z}_0$, but using the optimal control of Eq. (2.5.107); that is, $\hat{z}(t)$ would be the value which would be predicted for $z(t)$ if the prediction were being made at time t_0. Similarly, let P denote the variance of z conditioned on the same information. Thus,

$$\hat{z} = E(z) \tag{2.5.108}$$

and

$$P = E(z z^T) \tag{2.5.109}$$

Differentiating these expressions and making use of Eqs. (2.5.100) and (2.5.107) provides

$$\dot{\hat{z}} = -H\phi G Q_2^{-1} G^T \phi^T H^T S \hat{z} \tag{2.5.110}$$

$$\dot{P} = -H\phi G Q_2^{-1} G^T \phi^T H^T S P - P S H \phi G Q_2^{-1} G^T \phi^T H^T + H \phi \Sigma \phi^T H^T \tag{2.5.111}$$

while the boundary conditions are

$$\hat{z}(t_0) = \hat{z}_0 \tag{2.5.112A}$$

$$P(t_0) = \hat{z}_0 \hat{z}_0^T \tag{2.5.112B}$$

Thus, a terminal constraint on $E(z_f z_f^T)$ has been reduced to a constraint on $P(t_f)$ since

$$E\left\{z_f z_f^T\right\} = P(t_f) \tag{2.5.113}$$

The correct value of Λ , that is, the value of Λ which will satisfy the terminal constraint, can now be determined by the simultaneous solution of the P and S equations (i.e., Eqs. (2.5.106A) and (2.5.111) with the initial condition of Eq. (2.5.112B), the terminal condition of Eq. (2.5.106C) and with Λ selected so that $P(t_f)$ satisfies the terminal variance constraint which is imposed.

In most cases, the solution will have to be achieved iteratively. Thus, the process might proceed as follows:

(1) Guess the diagonal matrix Λ . As has been noted, the number of independent diagonal elements (i.e., the number of different quantities that can be guessed) is equal to the number of terminal constraints imposed. For example, if Eq. (2.5.87A) is used, then only one constraint is imposed and all the diagonal elements of Λ are equal to some number, say λ . This number would be guessed to start the iteration.

(2) Integrate the equation for S backwards in time with $S(t_f) = \Lambda$ [i.e., integrate Eq. (2.5.106A)] .

(3) Set $P(t_0) = \hat{z}_0 \hat{z}_0^T$ and integrate the P equation forward from t_0 to t_f [i.e., Eq. (2.5.111)].

(4) Test $P(t_f)$ to see if the specified terminal constraints are satisfied.

(5) If the constraints are not satisfied, adjust Λ and go back to Step (2).

Since the terminal constraints are inequality constraints [see Eq. (2.5.87)] , this iteration scheme will not lead to a unique solution. However, it can be shown, using standard methods from the Calculus of Variations, that Λ must be a negative semi-definite matrix, with the diagonal elements all less than or equal to zero. This condition suggests that the iteration loop above should start with the condition $\Lambda = 0$; furthermore, it generally allows for a unique solution to the iteration problem.

Summarizing the results for the perfectly observable case, the optimal feedback control is given by Eq. (2.5.107) where the matrix S is determined from Eq. (2.5.106A). The Lagrange multiplier matrix Λ is selected so that the simultaneous solution of Eq. (2.5.106A) and (2.5.111) lead to a control which satisfies the specified terminal constraints.

2.5.3.2 Perfectly Inobservable Case

The treatment of the perfectly inobservable case parallels that given in Section (2.5.2.2) where no terminal conditions were imposed. Again, the problem is to minimize the performance index

$$E(J) = E\left\{\int_{t_0}^{t_f} (z^T Q z + u^T Q_2 u)\, dt + z_f^T \Lambda z_f\right\} \tag{2.5.114}$$

subject to the state equations

$$\dot{z} = H\phi G u + H\phi \xi \tag{2.5.115}$$

and a terminal constraint on the quantity $E\ (z_f\, z_f^T)$. The initial state z_0 is a Gaussian variable with mean and covariance given in Eq. (2.5.96).

Let $\hat{z}$ denote the expected value of z and P its covariance; that is,

$$\begin{aligned} \hat{z} &= E\ (z) = H\phi E(X) \\ P &= E\left\{(z - \hat{z})\,(z - \hat{z})^T\right\} = H\phi E\left\{(X-\hat{X})\,(X-\hat{X})^T\right\}\phi^T H^T \end{aligned} \tag{2.5.116}$$

Now, the expected value and covariance of X were calculated for the perfectly inobservable treatment given in Section (2.5.22) [see Eq. (2.5.36)] . Substituting these expressions into (2.5.116) provides

$$\begin{aligned} \dot{\hat{z}} &= H\phi G u \\ \dot{P} &= H\phi\, \Sigma\, \phi^T H \end{aligned} \tag{2.5.117}$$

with the boundary conditions

$$\begin{aligned} \hat{z}_0 &= H\phi \hat{X}_0 \\ P_0 &= H\phi V_0\, \phi^T H^T \end{aligned} \tag{2.5.118}$$

Also, letting

$$z = \hat{z} + \bar{z} \tag{2.5.119}$$

it follows that

$$E\ (\bar{z}) = E\ (\bar{z}\, \hat{z}^{-1\,T}) = 0 \tag{2.5.120}$$

$$\dot{\bar{z}} = H\phi \xi \tag{2.5.120A}$$

Thus, substituting the value for $\bar{z}$ given in (2.5.119) into (2.5.114) reduces the performance index to

$$\begin{aligned} E\ (J) &= \int_{t_0}^{t_f} (\hat{z}^T Q \hat{z} + u^T Q_2 u)\, dt + \hat{z}_f^T \Lambda \hat{z}_f + E\left\{\int_{t_0}^{t_f} (\bar{z}^T Q \bar{z})\, dt + \bar{z}_f^T \Lambda \bar{z}_f\right\} \\ &= \left\{\int_{t_0}^{t_f} (z^T Q \hat{z} + u^T Q_2 u)\, dt + \hat{z}_f^T \Lambda \hat{z}_f + \left\{\mathrm{tr}\ (\Lambda P_f) + \int_{t_0}^{t_f} \mathrm{tr}\,(QP)\, dt\right\}\right. \end{aligned}$$

(2.5.121)

Thus, the control is to be selected to minimize the quantity inside the first set of brackets in Eq. (2.5.121) (the quantity in the second bracket does not depend on u), and the stochastic problem has been reduced to deterministic form.

Let

$$R(\hat{z}, t) = \min_{\substack{u(\tau) \\ t \le \tau \le t_f}} \left\{ \int_{t_0}^{t_f} (\hat{z}^T Q \hat{z} + u^T Q_2 u)\, dt + \hat{z}_f^T \Lambda \hat{z}_f \right\}$$

Then, using the Dynamic Programming approach, it follows that

$$0 = \min_{u(t)} \left\{ \hat{z}^T Q \hat{z} + u^T Q_2 u + \frac{\partial R}{\partial t} + \left(\frac{\partial R}{\partial \hat{z}}\right)^T H \phi G u \right\}$$

with the solution

$$R(\hat{z}, t) = \hat{z}^T S(t) \hat{z} \tag{2.5.122}$$

$$\dot{S} + Q - S H \phi G Q_2^{-1} G^T \phi^T H^T S = 0 \tag{2.5.123}$$

$$S(t_f) = \Lambda \tag{2.5.124}$$

The optimal control is given by

$$u = - Q_2^{-1} G^T Q^T H^T S \hat{z} \tag{2.5.125}$$

To determine the value of Λ for which the terminal constraint is satisfied, note that

$$\begin{aligned} E(z_f z_f^T) &= E\left\{(z - \hat{z}_f)(z - \hat{z}_f)^T\right\} + \hat{z}_f \hat{z}_f^T \\ &= P(t_f) + \hat{z}_f \hat{z}_f^T \end{aligned}$$

Since the quantity $P(t_f)$ is independent of the control action (see (2.5.117)), a constraint on $E(z_f z_f)$ is equivalent to a constraint on the quantity $\hat{z}_f \hat{z}_f^T$. Let

$$W(t) = \hat{z}\hat{z}^T$$

then from (2.5.117) and (2.5.125)

$$\dot{W} = -H\phi G Q_2^{-1} G^T \phi^T H^T S W - W S H \phi G Q_2^{-1} G^T \phi^T H^T \tag{2.5.126}$$

with

$$W_0 = \hat{z}_0 \hat{z}_0^T \tag{2.5.127}$$

Thus Λ is to be selected so that the simultaneous solution of the S and W equations, which satisfies the boundary conditions of Eq. (2.5.124) and (2.5.127), provides a value of $W(t_f)$ which satisfies the terminal constraint. As in the previous case, the solution will usually require iteration. However, the matrix Λ is again negative semi-definite and this condition will aid in the iteration process.

2.5.3.3 Partially Observable Case

The problem is to select the control u to minimize the functional

$$E(J) = E\left\{\int_{t_0}^{t_f} (z^T Q z + u^T Q_2 u)\,dt + z_f^T \Lambda z_f\right\} \tag{2.5.128}$$

subject to the state equation

$$\dot{z} = H\phi G u + H\phi\xi$$

and a terminal constraint on $E(z_f z_f^T)$ In this case, however, observations of the state variable x are made continuously as represented by the observation equation

$$Y = Mx + \eta \tag{2.5.129}$$

where η is a Gaussian white noise with zero mean and variance $\Gamma(t)$; that is,

$$E(\eta) = 0$$
$$E\left\{\eta(t)\,\eta(\tau)^T\right\} = \Gamma(t)\,\delta(t-\tau) \tag{2.5.130}$$

Note, as in Section (2.5.2.3), the performance index can be written as

$$E\,(J) = \underset{y}{E}\,\left\{E\,[J/y]\right\}$$

$$= \underset{\substack{Y(\tau)\\ t\le\tau\le t_f}}{E}\left\{E\left[\int_{t_o}^{t_f}(z^TQz + u^TQ_2u)\,dt + z_f^T\Lambda z_f \,/\, y\right]\right\} \qquad (2.5.131)$$

Thus, letting

$$\hat{z}\,(t) = E\,\left(z\,(t)/y\,(t)\right);\quad \hat{x}\,(t) = E\,\left(x\,(t)/y\,(t)\right) \qquad (2.5.132)$$

and

$$\begin{aligned} P\,(t) &= E\left\{(z-\hat{z})\,(z-\hat{z})^T/y\,(t)\right\}\\ V\,(t) &= E\left\{(x-\hat{x})\,(x-\hat{x})^T/y\,(t)\right\} \end{aligned} \qquad (2.5.133)$$

it follows that

$$\begin{aligned} \hat{z} &= H\phi\hat{x}\\ P &= H\phi V\phi^TH^T \end{aligned} \qquad (2.5.134)$$

But the quantities $\hat{x}$ and V are given in Eqs. (2.5.63A) and (2.5.63B). Thus, using these expressions provides

$$\begin{aligned} \dot{\hat{z}} &= H\phi Gu + H\phi VM^T\Gamma^{-1}(Y - M\hat{x})\\ \dot{P} &= -H\phi VM^T\Gamma^{-1}MV\phi^TH^T + H\phi\Sigma\phi^TH^T \end{aligned} \qquad (2.5.135)$$

Note that these two equations contain the mean and covariance of the vector x . This fact will not effect the analysis since the matrix does not depend on the control. Thus, if $\hat{x}$ is evaluated at any point, the corresponding value of $\hat{z}$ can be readily determined. Finally, let

$$z = \hat{z} + \bar{z} \qquad (2.5.136)$$

and observe that

$$E\,(\bar{z}/y) = E\,(\bar{z}\,\hat{z}^T/y) = 0$$

Substituting this value for z into (2.5.131) yields

$$E\,(J) = \underset{\substack{y(\tau)\\ t_o \le \tau \le t_f}}{E} \left\{ \int_{t_o}^{t_f} (\hat{z}^T Q \hat{z} + u^T Q_2 u)\, dt + \hat{z}_f^T \Lambda \hat{z}_f \right\} \tag{2.5.137}$$

$$+ \int_{t_o}^{t_f} tr\,(Q H \phi V \phi^T H^T)\, dt + tr\,(\Lambda H V_f H^T)$$

Since only the first term in this expression depends on the control, the problem of minimizing $E\,(J)$ has been reduced to minimizing

$$\underset{\substack{y(\tau)\\ t_o \le \tau \le t_f}}{E} \left\{ \int_{t_o}^{t_f} (\hat{z}^T Q \hat{z} + u^T Q_2 u)\, dt + \hat{z}_f \Lambda \hat{z}_f \right\} = MIN \tag{2.5.138}$$

Following a procedure identical to that used in Section (2.5.2.3), let $R\,(\hat{z}, t)$ be the minimum value of the performance index in (2.5.138) for the solution which starts the point $(\hat{z}, t)$. Using the Dynamic Programming approach, it follows that

$$R\,(\hat{z}, t) = \underset{\substack{u(\tau)\\ t \le \tau \le t+\Delta t}}{MIN} \left\{ (\hat{z}^T Q \hat{z} + u^T Q_2 u)\,\Delta t + o(\Delta t) + \underset{\substack{y(\tau)\\ t \le \tau\, t + \Delta t}}{E} \left[R\,(\hat{z} + \Delta \hat{z}, t + \Delta t \right] \right\}$$

Thus, taking the limit and using Equations (2.5.134) and (2.5.135) provides

$$0 = \underset{u\,(t)}{MIN} \left\{ \hat{z}^T Q \hat{z} + u^T + Q_2 u + \frac{\partial R}{\partial t} + \left(\frac{\partial R}{\partial \hat{z}}\right) H \phi G u \right.$$

$$\left. + \frac{tr}{2} \left[\frac{\partial^2 R}{\partial \hat{z}^2}\, H \phi V M^T \Gamma^{-1} M V \phi^T H^T \right] \right\}$$

This equation has the solution

$$R = \hat{z}^T S(t)\, \hat{z} + \beta(t) \tag{2.5.139}$$

where

$$\dot{S} + Q - S H \phi G Q_2^{-1} G^T \phi^T H^T S = 0 \tag{2.5.140}$$

$$\dot{\beta} + \mathrm{tr}\,(SH\phi VM^T\Gamma^{-1}MV\phi^TH^T) = 0$$

$$S(t_f) = \Lambda$$

$$\beta(t_f) = 0 \qquad (2.5.141)$$

The optimal control takes the form

$$u = -Q_2^{-1}G^T\phi^TH^TS\hat{z} \qquad (2.5.142)$$

The selection of Λ to satisfy terminal constraints on the quantity $E(z_f z_f^T)$ is accomplished as follows. Note that

$$E(z_f z_f^T) = \underset{y}{E}\left\{E(z_f z_f^T/y)\right\} = \underset{y}{E}\left\{E\left[(z_f - \hat{z}_f)(z_f - \hat{z}_f)^T/y\right] + \hat{z}_f\hat{z}_f^T\right\}$$

But, since the covariance $P = H\phi V\phi^TH^T$ does not depend on Y, this expression becomes

$$E(z_f z_f^T) = HV_fH^T + \underset{y}{E}(\hat{z}_f + \hat{z}_f^T) \qquad (2.5.143)$$

Further, since the quantity HV_fH is deterministic and independent of the control, a constraint on $E(z_f z_f^T)$ is equivalent to a constraint on $E(\hat{z}_f\hat{z}_f^T)$. Thus, if $W(t)$ is given by

$$W(t) = \underset{y(t)}{E}\left[\hat{z}(t)\hat{z}(t)^T\right]$$

Then, using Eqs. (2.5.135) and (2.5.142), it follows that

$$\dot{W} = -H\phi GQ_2^{-1}G^T\phi^TH^TSW - WSH\phi GQ_2^{-1}G^T\phi^TH^T + H\phi VM^T\Gamma^{-1}MV\phi^TH^T \qquad (2.5.144)$$

with

$$W_0 = \hat{z}_0\hat{z}_0^T \qquad (2.5.145)$$

The matrix Λ is to be selected so that the simultaneous solution of the $\boldsymbol{w}$ and $\mathcal{S}$ equations, together with the boundary conditions in (2.5.141) and (2.5.145), yields a value of $w(t_f)$ which satisfies the terminal constraints. As in the preceding two cases, iteration will usually be required to accomplish the solution. Ref. (2.5.7) contains an interesting application of this partially observable case to the interplanetary guidance problem.

2.5.3.4 Discussion

The inclusion of the terminal constraints does not appreciably alter the problem, except that the solution must be accomplished iteratively, rather than directly. However, the iteration loop appears to be no more difficult than that normally encountered in optimal control problems. In some cases, when the number of terminal constraints is small, closed form solutions may be possible [see Ref. (2.5.7)] .

As mentioned at the beginning of this section, the linear-quadratic cost problem is not typical of stochastic optimization problems. The reason for this is that the analysis is concerned with the solution of partial differential equations. The linear-quadratic cost problem is one of the few cases in which the variables separate, and the partial differential equations reduce to ordinary differential equations.

For additional treatments of stochastic control problems, the interested reader should consult Refs. (2.5.1) to (2.5.7) as well as Chapter (7) of Ref. (2.4.1). Refs. (2.5.8) to (2.5.10) also contain an elegant application of stochastic control theory to the mid-course correction problem.

3.0 RECOMMENDED PROCEDURES

The preceding sections of this report have illustrated the dual nature of Dynamic Programming as both a theoretical and computational tool. It is the general consensus of opinion (see Ref. (2.4.1)) that on the theoretical level, Dynamic Programming is not as strong or as generally applicable as either the Calculus of Variations or the Maximum Principle. However, the relative strengths and weaknesses of Dynamic Programming when compared with the variational methods are of little importance. What is important is the fact that Dynamic Programming is a completely different approach to optimization problems and its use can provide perspective and insight into the solution structure of a multistage decision processes. Furthermore, there are some problems that are rather difficult to attack using the classical methods, but which readily yield to solution by means of Dynamic Programming. One such example is the stochastic decision problem treated in Section (2.5).

On the computational side, Dynamic Programming has no equal as far as versatility and general applicability are concerned. Almost all optimization problems can be cast in the form of a multistage decision processes and solved by means of Dynamic Programming. However, it frequently happens that certain problems, or certain types of problems, are more efficiently handled by some other numerical method. Such is the case, for example, in regard to the trajectory and control problems normally encountered in the aerospace industry.

It has been amply demonstrated in the last few years that optimal trajectory and control problems can be solved using a variational formulation procedure coupled with a relatively simple iterative technique such as quasilinearization (Ref. (3.1)), steepest ascent (Ref. 3.2)) or the neighboring extermal method (Ref. (3.3)). The voluminous number of papers and reports dealing with problem solution by this method attest to its effectiveness. On the other hand, there are relatively few reports which treat trajectory or control problems using Dynamic Programming. The reason for this can be partially attributed to the "newness" of Dynamic Programming and the fact that other numerical procedures were available and were used before Dynamic Programming "caught on." More important, however, is the fact that solution generation by means of Dynamic Programming usually requires more computation, more storage, and more computer time than do the other numerical methods.

The role of Dynamic Programming in the flight trajectory and control area should increase in the not too distant future. Presently used techniques have been pushed almost to their theoretical limits and leave something to be desired as more complex problems are considered and more constraint conditions included. Dynamic Programming, on the other hand, is limited only by the computer, a limitation which is continuously on the decrease as more rapid and flexible computing equipment is developed.

4.0 REFERENCES

Section 2.1

(2.1.1) Bellman, R., "On the Theory of Dynamic Programming," Proceedings of the National Academy of Science, Vol. 38, pgs. 716-719 (1952).

(2.1.2) Bellman, R., Dynamic Programming, Princeton University Press, 1957.

(2.1.3) Bellman, R., Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.

(2.1.4) Bellman, R. and Dreyfus, S., Applied Dynamic Programming, Princeton University Press, 1962.

(2.1.5) Bellman, R., "Is Science Big Enough to Cope with Society," Saturday Review, June 5, 1965.

Section 2.2

(2.2.1) Mangad, M. and Schwartz, M., The Calculus of Variations and Modern Applications, NAA, S&ID, SID 65-1200-4, 1966.

(2.2.2) Elsgolc, L. E., Calculus of Variations, Pergamon Press Ltd., 1962.

(2.2.3) McMurray, L. R., Introduction to Dynamic Programming, NAA/Autonetics, TM 3241-2-2, 1962.

Section 2.4

(2.4.1) Dreyfus, S. E., Dynamic Programming and the Calculus of Variations, Academic Press, 1965.

(2.4.2) Sneddon, Ian, Elements of Partial Differential Equations, McGraw Hill, 1957.

(2.4.3) Pontryagin, L. S., et al., The Mathematical Theory of Optimal Processes, Interscience Publishers, 1962.

(2.4.4) Leitman, G., "Variational Problems with Bounded Control Variables," Chapter 5 of Optimization Techniques, Academic Press, 1962.

(2.4.5) Isaev, V. K., "L. S. Pontryagin's Maximum Principle and Optimal Programming of Rocket Thrust," Automation and Remote Control, August 1961.

REFERENCES

Section 2.5

(2.5.1) Kushner, H. J., "Sufficient Conditions for the Optimality of a Stochastic Control", J.S.I.A.M. on Control, Volume 3, No. 3 (1966)

(2.5.2) Florentin, J. J., "Optimal Control of Continuous Time, Marker, Stochastic Systems", J. Electronics Control, June 1961

(2.5.3) Papoulis, A., Probability, Random Variables, and Stochastic Processes, McGraw - Hill, 1965

(2.5.4) Barret, J. F., "Application of Kolmogorov's Equation to Randomly Disturbed Automatic Control Systems", I.F.A.C. Proceedings (1961), Volume II, Butterworths, 1962

(2.5.5) Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction Theory", A.S.M.E.J. Basic Engineering, March 1961

(2.5.6) Kushner, H. J., "On the Differential Equation Satisfied by Conditional Probability Densities of Mark on Processes, with Applications", J.S.I.A.M. on Control, Volume 2, No. 1, (1962)

(2.5.7) Tung, F., "Linear Control Theory Applied to Interplanetary Guidance", I.E.E.E. Transactions on Automatic Control, January 1964

(2.5.8) Striebel, C. T. and Breakwell, J. V., "Minimum Effort Control in Interplanetary Guidance", IAS Paper No. 63-80, presented at the IAS 31st Annual Meeting. New York. (January 1963)

(2.5.9) Tung, F., "An Optimal Discrete Control Strategy for Interplanetary Guidance", I.E.E.E. Trans. Automatic Control, AC-10. (July 1965)

(2.5.10) Breakwell, J. V., Rauch, H. E., and Tung, F. F., "Theory of Minimum Effort Control", NASA CR-378. (January 1966)

REFERENCES

Section 3.0

(3.1) McGill, R., and Kenneth, P., "Solution to Variational Problems by Means of a Generalized Newton-Raphson Operator," AIAA, 1964.

(3.2) Kelly, H., "Method of Gradients," Chapter 6 of Optimization Techniques, Edited by G. Leitmann, Academic Press, 1962.

(3.3) Breakwell, J. V., Speyer, J. L. and Bryson, A. E., "Optimization and Control of Nonlinear Systems Using the Second Variation," J.S.I.A.M. Control, Vol. 1, No. 2 (1963).

NASA-Langley, 1968 —— 30