THE APPLICATION OF DECISION THEORY AND DYNAMIC PROGRAMMING TO ADAPTIVE CONTROL SYSTEMS

THE APPLICATION OF DECISION THEORY AND DYNAMIC PROGRAMMING TO ADAPTIVE CONTROL SYSTEMS

By

LOUIS K KING LEE, B.ENG. (ELEC.)

/

A thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Engineering

McMaster University
September 1967

MASTER OF ENGINEERING (Electrical Engineering) | McMASTER UNIVERSITY Hamilton, Ontario

TITLE: The Application of Decision Theory and Dynamic Programming to Adaptive Control Systems.

AUTHOR: Louis K King Lee, B.Eng. (McMaster University)

SUPERVISOR: Dr. N. K. Sinha

NUMBER OF PAGES: VI, 61

SCOPE AND CONTENTS:

It is generally assumed that the implementation of adaptive control requires a precise identification of plant parameters. In the case of a system with varying parameters, the identification problem gets very involved, as speed of identification and accuracy are contradictory requirements.

In this thesis it has been shown that using a feedback policy, the optimal controller is relatively insensitive to changes in plant parameters as long as these lie within some specified ranges. It is, therefore, concluded that, with such an arrangement, adaptive control can be implemented if one has only the knowledge of the ranges within which the parameters of the plant lie. Thus identification can be carried on more rapidly, as stringent accuracy is no longer necessary.

ACKNOWLEDGEMENTS

The author wishes to acknowledge with sincere gratitude to Dr. N. K. Sinha for his invaluable guidance and encouragement in preparation of this thesis. The author is also thankful to the Department of Electrical Engineering for the scholarship and teaching assistantship.

Thanks are due to my parents for their patience, general assistance and advice during my absence from home. Also acknowledgement is due to Miss Marie Kaneary for typing the manuscript.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Since 1956 control engineers have given considerable attention to the development of the concept of adaptive control -- or a feedback control in which a system is capable of changing its controller parameters in order to achieve optimal performance in spite of large changes in the plant parameters. Adaptation to unpredictable conditions is in fact one of the basic requirements for a control system. However, the conventional feedback control system is only capable of maintaining the performance with small changes in plant parameters. This is the reason why intensive research in the field of adaptive control has been carried out in the last few years [1], [2], [3], [4], [5], [6], [7].

Identification refers to the process of characterizing plant dynamics. By their very nature, adaptive systems demand frequent and rapid solutions of the identification problem, with as much precision as possible. Thus, the identification problem is of special importance in any approach to the design of adaptive control systems. Moreover, the requirements of short measuring time and accuracy are in direct conflict to each other, making the problem very difficult in the case of systems with rapidly varying parameters.

In this thesis, a new approach has been suggested in order to overcome the difficulty mentioned above. The plant parameters are considered to be random and lie within some specified ranges with known probability distributions. The analysis is designed to search for an optimal controller (or near-optimal controller) which automatically adapts the system to the plant parameter variations within the specified ranges. The parameters of the calculated optimal controller for each range of plant parameters will be stored in the memory of the computer and hence will save time in adaptation.

Of the techniques available in control theory, decision theory and dynamic programming appear to be more appropriate. Decision theory is one of the most promising mathematical tool to be applied to control systems. A growing interest is evident from the numerous papers published recenlty [8], [9], [10]. In practice, decision theory is difficult to apply to control systems unless certain simplifying assumptions have been made [8], [10]. One of the examples of Reference [10] has been discussed in Chapter III in detail to illustrate that fact. Hence, dynamic programming is considered as the preferable alternative approach. In Chapter 4, the basic concept of dynamic programming and the formulation of dynamic programming from decision theory have been introduced by considering a simple game. The new approach, utilizing dynamic programming, designed to ease the identification problem has been demonstrated through four examples. All calculations in this thesis have been carried out on an IBM-7040 computer.

CHAPTER II

ADAPTIVE CONTROL SYSTEMS

2.1 Introduction:

With the progress being made in space, nuclear, and other industrial technologies, there is a growing need for automatic control systems which are capable of changing their own parameters in order to remain efficient in spite of large changes in their environments. This has led to intensive research during the past few years on adaptive control systems [1], [2], [3], [4], [5], [6], [7].

A common example of adaptive control is a human being steering an automobile. The driver continually injects small variational signals onto the steering wheel in order to maintain "the feel of the road and the car"; i.e. the driver is continually measuring the dynamics of the process to be controlled in order that he may be prepared to effect near optimum control when input signals arise (e.g., when the eye detects a curve in the road or when the driver is suddenly called upon to swerve the car to avoid an object on the road).

In this Chapter, the general definition and the classification of adaptive control and the problem of identification are introduced; and then it is explained why the parameters should be considered within sepcified ranges.

2.2 Definition of Adaptive Control:

Control systems can be divided into two main classes: adaptive and non-adaptive. Adaptive control systems may be defined as those which are capable of modifying their own parameters with changes in environments in such a manner that their performance is optimized on the basis of a prescribed criterion. Non-adaptive control systems do not have this facility.

All adaptive control systems perform some of the following operations: measurement, identification, pattern recognition, determination of optimum control strategy and modification of the controller.

2.3 Classification of Adaptive Control Systems:

Adaptive control systems can be classified into three types: the basic adaptive system, the static adaptive system and the dynamic adaptive system. The basic adaptive system is the simplest type. It does not have any facility for pattern recognition. The static adaptive system involves the comparison of the present environmental situation with the past records of different sets of such situations, and recognizing it as belonging to a particular set. The system can be compared to the technician who has memorized the solution to the problems he is most likely to encounter but yet he is not prepared to learn anything new and has not the capability of solving a new problem. The dynamic adaptive system works like the static adaptive system, but when a new or unexpected situation arises, thus creating a pattern which does not match with any of the stored patterns, the system would temporarily adjust to the pattern closest to the actual pattern while determination for optimum strategy is being made.

2.4 Problem of Identification:

Most control systems consist of two sub-systems: plant and controller. The plant is considered to be the mechanism to be controlled and has little design freedom in most cases. The controller is that part of the system which is designed with a view toward making the entire system work properly. Evidently the success with which a given plant can be controlled in a described fashion depends on how accurately its varying parameters are known. One of the open problems challenging design engineers involves the accurate measurement of the parameters of the control system to be optimized and the computation of the change in the performance index in response to parameter variations.

Identification involves the use of the measured data for the determination of certain unknown parameters. Practically, identification should be made in the presence of normal operating signals and noise disturbances. Also, any test performed on the process must not unduly disturb the normal operation. A typical configuration of an adaptive control system is as shown in Figure 2-1.

It is noted in Figure 2-1 that the measurement of parameter variations is done in a finite time which should be chosen sufficiently small so that the effect of the variations of the parameters is insignificant. It is, however, impossible to make the measurement time short without decreasing its accuracy. Conversely, one must take a fairly long time for the measurement if it is to be done with significant accuracy. Therefore, the demand for short measurement time and the necessity for accurate identification are in direct conflict with each other. As a matter of fact, it is impossible to construct the error-free identification of the parameter because of all sorts of errors caused by the short measurement time interval and external noise, incompleteness of equipments etc.

Now let us consider the unknown parameter to be identified in the system shown in Figure 2-1 to be expressed by $\theta(t)$. In the process of performing the measurement or observation, random disturbances are added to the input of the measuring device. Therefore, it gives an imperfect measurement $\theta^*(t)$ of the value of the parameter so that

$$\theta^*(t) = \theta(t) + n(t)$$

where $n(t)$ may be considered as the presence of noise contaminating the incoming signal to the measuring device. Thus, the only known quantity is the measured value $\theta^*(t)$, and information concerning the true value of $\theta(t)$ must be inferred from this acquired data $\theta^*(t)$.

The central problem in adaptive control systems is the determination of the controller parameters on the basis of the above incomplete information. Due to the handicaps discussed previously, it is natural and practical to consider that the varying plant parameters lie in specified ranges. In general, the ranges of plant parameters and their probability distributions may be determined by statistical inference from the previous measurements or by the specifications of the designer. Using feedback policy, the analysis of the design is to search for an optimal controller which is relatively insensitive to variations in plant parameters as long as they lie within those ranges. This approach may overcome the difficulties that arise in the identification problem.

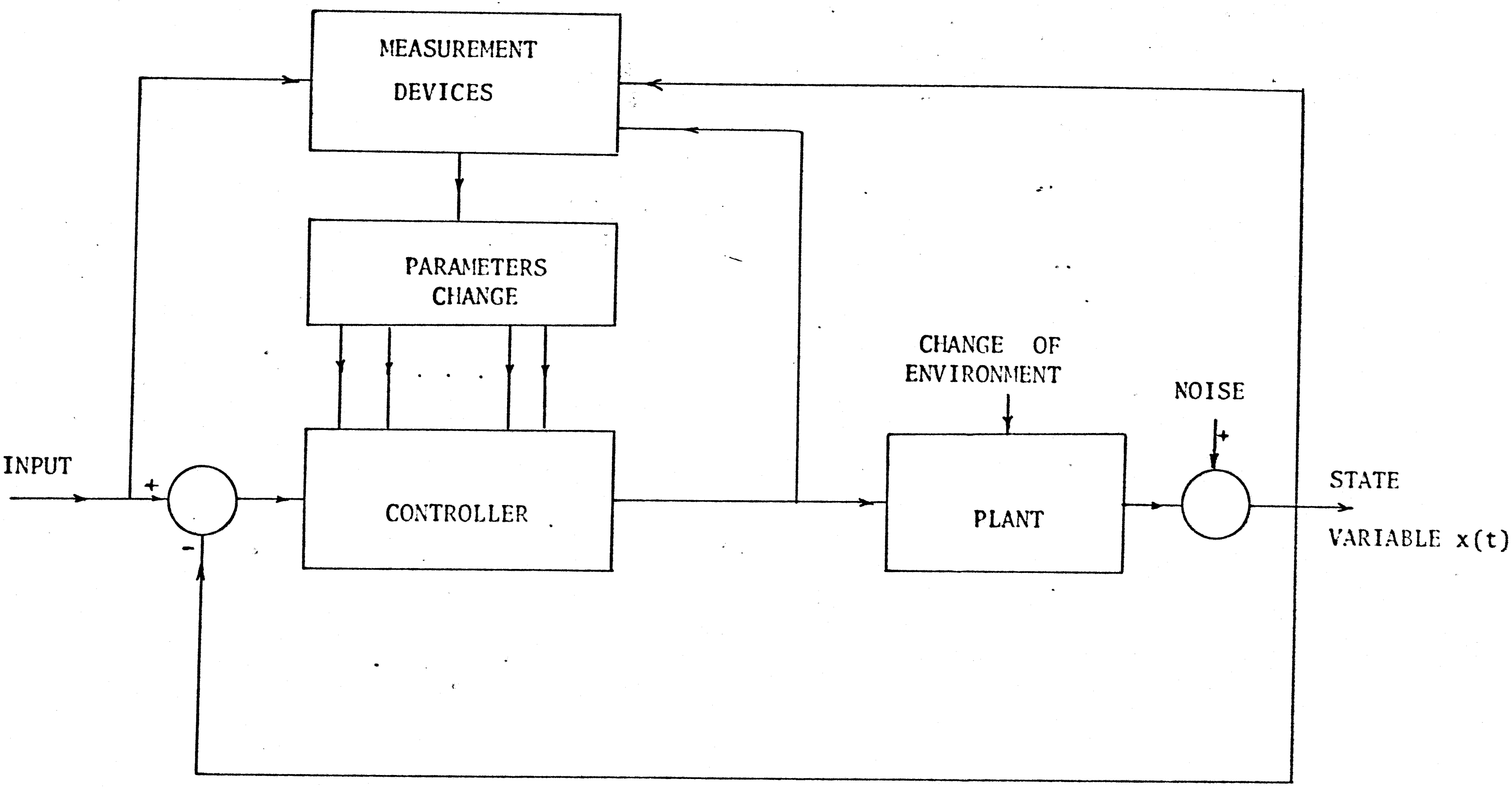


FIGURE 2-1: An Adaptive Control System

CHAPTER III

DECISION THEORY

3.1 Historical Perspective:

The foundations of decision theory were first laid by John von Neumann in the late 1920's. von Neumann and Oskar Mogenstern published their book [11] in 1953, and since then, the theory has been widely applied to economics, games, and military situations. In 1958, N. M. Abramson presented an excellent paper [12], which was the first application of Game Theory to Electrical systems. In the field of adaptive control, J. G. Truxal and J. J. Padalino have suggested an approach using decision theory [8]. However, their example is too simple and artificial. Recently Dorato and Kestenbaum have applied decision theory to the sensitivity problem of control systems [10]. Their approach appears to be quite natural but their examples are limited to second-order systems with only one varying plant parameter. One of their examples will be discussed later in this Chapter in order to illustrate the possibility of application of decision theory to control systems.

3.2 The Basic Concept of Decision Theory: [13] [14]

In broad terms decision theory is concerned with the problem of making the optimum decision when one is faced with a choice of several possibilities. Since it is easier to start with games, let us consider the game of matching pennies. Players A and B each display simultaneously a single penny. A takes B's penny if A matches his penny with B's, i.e. either both pennies are heads or both are tails. Otherwise, B takes A's penny. The pay-off is represented in matrix form as following:

A \ B	H	T
H	+1	-1
T	-1	+1

where (i) +1 means A takes B's penny and

(ii) -1 means A loses his penny to B.

If either one of the players constantly uses the same pure strategy, the other can take advantage of this, e.g. if A shows head constantly, B can show tail each time and win. So a mixed form of strategy of heads and tails will be used. This is a very simple example of the application of decision theory.

Now let us consider a game with a pay-off table as below:

A \ B	Strategy B1	Strategy B2
Strategy A1	\$1.00	\$6.00
Strategy A2	\$6.00	\$4.00

In this game, A first pays B a definite amount which may be considered as the "value of the game" and depends on the pay-off table. And A then chooses one of his strategy A1 or A2. Without knowing A's choice, B chooses his strategy B1 or B2. They then compare their choices and B pays A according to pay-off table. For example, if B chooses his strategy B1 and A has his strategy A1, B then pays A one dollar as indicated on the pay-off table; if B chooses his strategy B2 and A chooses his strategy A1, then B pays A six dollars as indicated.

From the above table, if A chooses his pure strategy A1 and B uses his strategies B1 or B2 in the ratio x:(1-x). In other words, the probability of B choosing B1 is x and choosing B2 is 1-x, and his pay-offs for B1 and B2 are one dollar and six dollars respectively as indicated on the pay-off table. Then B's average pay-off(I) is denoted by

$$pyf = 1\ x + 6(1-x) = -5\ x + 6 \text{ dollars}$$

While if A chooses his pure strategy A2, then using the same ratio for his strategies, B's average pay-off (II) is denoted by

$$pyf = 6\ x + 4(1-x) = 2\ x + 4 \text{ dollars}$$

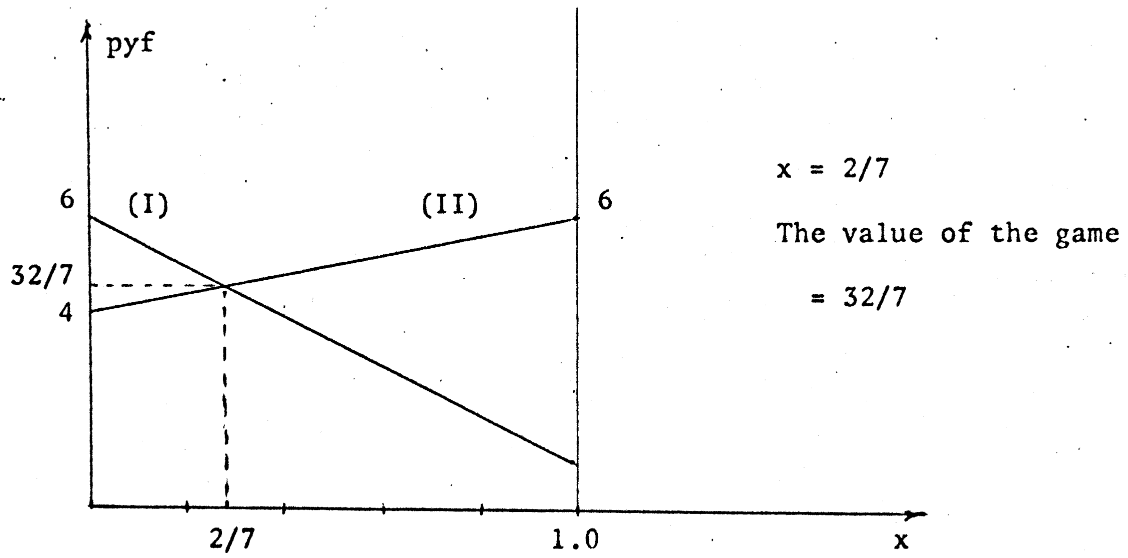


FIGURE 3-1: Graphical Solution of a 2 x 2 Matrix Game

The average pay-off (I) and (II) are plotted against x as shown in the above Figure 3-1. The intersection point is at $x = 2/7$. This means that the best strategy for B is to choose his strategy B1 two-sevenths of the time and strategy B2 five-sevenths of the time in a random manner. The same principle applies to player A. It is noticed that B intends to minimize his pay-off and A intends to maximize B's pay-off. This leads to the important minimax theory.

The minimax theory states that a necessary and sufficient condition for optimal strategies is as follows:

$$M = \max_{X} \min_{Y} M(X, Y)$$

$$= \min_{Y} \max_{X} M(X, Y)$$

Where X is the random strategy chosen from a set A to maximize the pay-off M, Y is the strategy chosen from another set B to minimize the pay-off M. The order of the operators $\min_{Y}$ and $\max_{X}$ makes no difference, in other words, the operators are commutative. The detailed statement of the theory is in Appendix I [15].

3.3 Application of Decision Theory to the Minimum Sensitivity Design of Optimum Systems:

Dorato and Kestenbaum [10] have suggested that decision theory may be applied to system design with unknown plant parameters. In their example, it has been assumed that a controller structure is known and furthermore that this controller is optimal when the controller parameter is equal to the plant parameter.

Consider a linear control system as shown in Figure 3-2.

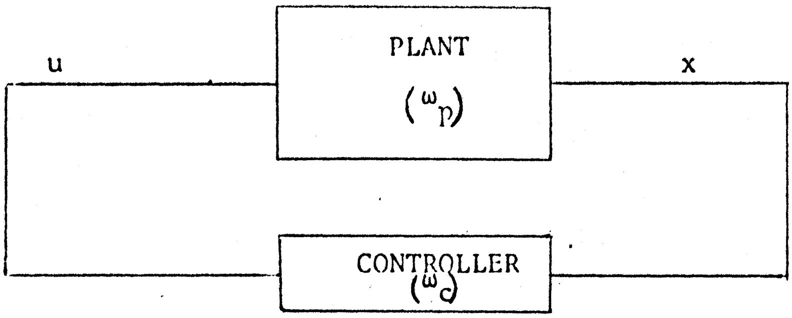


FIGURE 3-2: A Linear Control Problem

$$\dot{\underline{X}}(t) = \underline{f}(\underline{X}(t), \underline{u}(t), \omega_p)$$

$$\underline{u}(t) = \underline{\psi}(\underline{X}(t), \omega_c, t)$$

or $$\underline{u}(t) = D\, \underline{X}(t)$$

$$S = \int_{t_o}^{T} F(\underline{X}, \underline{u})\, dt$$

where

$\underline{X}(t)$ is the state vector

$\underline{u}(t)$ is the forcing function

S is the performance index

ω_p is the plant parameter

ω_c is the controller parameter

The plant parameter ω_p is known lying somewhere in the range $\omega_1 \leqslant \omega_p \leqslant \omega_2$. The problem is to choose a matrix D which is in terms of ω_c (ω_c is considered to have the same range as ω_p) will maintain as low as possible a value of S. On the other hand, ω_p intends to maximize the value of S. Obviously ω_c and ω_p are opposite to each other and the performance index S can be considered as the pay-off function in decision theory.

Technically the statement is:

$$S^o = \min_{\omega_c} \max_{\omega_p} S(\omega_c, \omega_p)$$

$$= \max_{\omega_p} \min_{\omega_c} S(\omega_c, \omega_p)$$

$$= S(\omega_c^o, \omega_p^o) \quad .$$

The pair (ω_c^o, ω_p^o) is referred to as the optimum strategy and

$$S(\omega_c^o, \omega_p) \leq S^o \qquad \omega_1 \leq \omega_p \leq \omega_2$$

Here, ω_c^o is so chosen that S is always less than S^o no matter what value ω_p may be.

The following numerical example has been considered in their paper [10]:

A plant with dynamics

$$\dot{X} + \omega_p X = u$$

where $0 \leq \omega_p \leq 2$ with initial value $X(0) = 1$ and $\dot{X}(0) = 0$, and a performance index

$$S = \int_o^\infty (X^2 + \dot{X}^2 + u^2)\, dt$$

Dorato and Kestenbaum have obtained the optimal solution by using the minimax theory mentioned previously,

$$S^o = \min_{D_2} \max_{\omega_p} S(D_2, \omega_p)$$

$$= S(D_2^o, \omega_\rho^o)$$

$$= S(-0.64, 2)$$

where

$$D_2 = -\omega_c - \sqrt{3 + \omega_c^2}$$

The procedure for obtaining the solution is well presented in their paper [10] and hence will not be repeated here.

The same problem has been considered in Rohrer and Sobrals' paper [22]. They have obtained the optimal u(i.e. u^o) by classical calculus and also by considering its relative sensitivity.

Now let us compare their results:

$$u = -\begin{bmatrix} D_1 & D_2 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \qquad \text{general form}$$

$$u = -\begin{bmatrix} 1.0 & 1.\bar{3} \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \qquad \text{by classical calculus}$$

$$u = -\begin{bmatrix} 1.0 & 0.64 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \qquad \text{by decision theory}$$

First of all, this example might be specially chosen by the authors so that the first controller parameter D_1 is independent of the plant parameter ω_p, i.e. D_1 is always equal to 1.0. Hence, the only variable controller parameter is D_2. Secondly the inaccuracy of Dorato and Kestenbaums' calculation of D_2 is most likely due to the assumption that the system is optimum when the controller parameter is equal to the plant parameter. It seems that the authors have made this assumption only for the sake of simplifying the calculation and have ignored the fact that the assumption is not practical.

On the whole, the method suggested in their paper is probably limited to lower order system and few variables, (i.e., only second order system with one varying plant parameter and one varying controller parameter has been considered in their paper). It is obvious that the joint probability of many variables and higher order systems will make the computation tedious, especially when the calculation has to be done manually.

To overcome the difficulties mentioned above, dynamic programming has been utilized as a powerful tool for the optimization of control systems.

CHAPTER 4

DYNAMIC PROGRAMMING

4.1 Introduction:

In the 1950's, R. Bellman developed dynamic programming; and since then, it has been applied to numerous fields, e.g. theory of inventory and production, purchasing and investmemt problems, design of chemical plants, statistical communication theory and control systems. Bellman has published some distinguished books on dynamic programming and adaptive control system [16], [17], [18], but extremely little of the theory has been converted into practice [19].

Dynamic programming is related to multi-stage decision processes in which a decision is required at discrete intervals in time (e.g., the series of decisions required in card games such as Contract Bridge and in the management of an industrial production process). Furthermore, all continuous decision processes can be approximated by multi-stage decision processes by utilizing a small discrete interval between decisions.

In this Chapter, firstly an example solved by decision theory has been described in great detail in order to emphasize the essential concept of dynamic programming and the basic idea of formulating the dynamic programming from decision theory. Then the characteristics and the general advantages of dynamic programming are discussed. Finally, four control systems (from first order to fourth order) have been solved by dynamic programming and their numerical results tabulated. The second order system discussed in Chapter III is one of the four examples; the prupose is to serve the comparison of the application of decision theory and dynamic programming to the control system.

4.2 Formulation of Dynamic Programming from Decision Theory - One Example:

In this game, we assume two players A and B; A first pays B a definite amount (value of the game). Each player initially possesses three chips; on the first move, each may play one or two chips. A and B play simultaneously, and after the play B pays A an amount as indicated by the pay-off table:

A \ B	1	2
1	$0	$30
2	$10	$0

(i.e., if A plays 1 and B plays 2, B pays A $30).

After the first play and pay-off, those chips which were played are removed and the game continues. A and B each possess either one or two chips (depending on how many each played in the first play). After the second play, we again have a pay off described by the above pay-off table. The game continuous in this fashion until one player has no chips left. The problem is to determine how each player should play to maximize his total return from the overall game. This is not a single stage game but a multi-stage one. Solution of the problem requires that we determine a complete strategy for the succession of plays.A will make during a complete game. There are five possible strategies which A may adopt before the game starts:

a_1: plays 2, then plays 1

a_2: plays 1, then plays 1 if B first played 1, 2 if B first played 2

a_3: plays 1, then plays 2 if B first played 1, 1 if B first played 2

a_4: plays 1, then plays 1 regardless of B's first play

a_5: plays 1, then plays 2 regardless of B;s first play

Actually two of these strategies are obviously undesirable: if B first played 2, A knows that B's second play must be 1; if A has a choice for this second play, he certainly should choose 2 rather than 1 ($10 is better than zero). Hence strategies a_3 and a_4 can be eliminated, the possible strategies left are a_1, a_2, a_5. Similarly a dual line of reasoning indicates that B has the possible strategies b_1 b_3 b_4. Then an overall pay-off table has been formed:

A \ B	b_1	b_3	b_4
a_1	$ 0	$10	$10
a_2	$ 40	$30	$ 0
a_5	$ 40	$ 0	$10

Solution of this game with the standard techniques yields the optimum strategies for A and B:

A: - $a_1 : a_2 : a_3 : a_4 : a_5 = \frac{13}{17} : \frac{1}{17} : 0 : 0 : \frac{3}{17}$

B: - $b_1 : b_2 : b_3 : b_4 : b_5 = \frac{1}{17} : 0 : \frac{4}{17} : \frac{12}{17} : 0$

and the value of the game $= \$\frac{160}{17}$.

In the above solution, we essentially reduce the multi-stage process to a single three-by- three game.

In the dynamic programming formulation of the above problem, we first define a function $f(x, y)$ as the expected return to A from the overall game when A starts with x chips and B with y chips. Further, we define p_1 as the probability that A plays 1, p_2 that A plays 2, and q_1 and q_2 correspondingly for player B. In terms of these definitions and the pay off table which is specified, a simple enumeration of all possibilities leads to a functional equation for $f(x, y)$:

$$f(x, y) = \sum_{i=1}^{2} \sum_{j=1}^{2} p_i q_i \left[\mathcal{A}_{ij} + f(x - i, y - j)\right] \qquad \text{----(4.1)}$$

where $\mathcal{A}_{ij}$ is the entry in the pay off matrix for A playing i and B playing j. The above expression states that the expected return is the pay off from the first play $\sum_{i=1}^{2} \sum_{j=1}^{2} p_i q_j \mathcal{A}_{ij}$ plus the return which can be expected from all later plays in the same game (the later plays essentially comprising a new game with the starting resources (x-i) and (y-j) for players A and B, respectively). There is still one important element not included in the above formulation: The requirement that A and B should play in such a way as to maximize their individual returns. The Minimax Theory is then applied, the function equation for $f(x, y)$ is:

$$f(x, y) = \max_{p} \min_{q} \left\{ \sum_{i=1}^{2} \sum_{j=1}^{2} p_i q_j \left[\mathcal{A}_{ij} + f(x - i, y - j)\right] \right\} \text{----(4.2)}$$

(A must select the p_i in such a way as to maximize his return while B is selecting the q_i to minimize his pay to A).

Equation (4.2) then represents the complete dynamic programming formulation of the problem with the following constraints:

$$0 \leq p_i \leq 1 \qquad 0 \leq q_i \leq 1$$

$$p_1 + p_2 = 1 \qquad q_1 + q_2 = 1$$

$$f(x, y) = 0 \qquad \text{for } x \text{ or } y \leq 0$$

The example which we have described above is simple and its function equation is:

$$f(3, 3) = \max_{p} \min_{q} \{p_1q_1 [\mathcal{a}_{11} + f(2,2)] + p_1q_2 [\mathcal{a}_{12} + f(2, 1)]$$

$$+ p_2q_1 [\mathcal{a}_{21} + f(1, 2)] + p_2q_2 [\mathcal{a}_{22} + f(1, 1)]\} \quad \text{----(4.3)}$$

With $p_2 = 1 - p_1$ $q_2 = 1 - q_1$.

The value of $\mathcal{a}_{ij}$ are given in the payoff table at the beginning of this problem as

$$\mathcal{a}_{11} = 0 \qquad \mathcal{a}_{12} = \$30 \qquad \mathcal{a}_{21} = \$10 \qquad \mathcal{a}_{22} = 0$$

The solution can be determined iteratively by starting with the final decisions (or the simplest decision) and working up to more complex cases. In this example, the game stops whenever either player has no more chips; hence $f(0, 0) = 0$, $f(0, 1) = 0$, $f(1, 0) = 0$, $f(2, 0)=0$ and $f(0, 2) = 0$. Furthermore we can write $f(1, 1) = 0$ since if each player possesses only one chip, he must play 1 and the $\mathcal{a}_{11}$ payoff is zero. The value $f(1, 2)$ can likewise be determined directly: If A has only one chip, he must play 1 and B knows that A will play 1; hence, B will always play 1 also and there will be zero payoff: $f(1, 2) = 0$. Similarly $f(2, 1) = \$10$.

Now we have to formulate $f(2, 2)$; an equation similar to equation (4.3) has been formed:

$$f(2, 2) = \max_{p} \min_{q} \{p_1q_1 [\mathcal{q}_{11} + f(1, 1)] + p_1q_2 [\mathcal{q}_{12} + f(1, 0)] + p_2q_1 [\mathcal{q}_{21} + f(0, 1)] + p_2q_2 [\mathcal{q}_{22} + f(0, 0)]\} \quad \text{----(4.4)}$$

By standard techniques, a two-by-two game has been formed and the value of f(2, 2) is found as $\$\frac{30}{4}$.

The above values of $\mathcal{q}_{ij}$ and f(x, y) are substituted into equation (4.3):

$$f(3, 3) = \max_{p} \min_{q} \{(\tfrac{30}{4})\, p_1q_1 + (40)\, p_1q_2 + (10)\, p_2q_1 + (0)\, p_2q_2\} \quad \text{----(4.5)}$$

Now the problem has been reduced to a single stage game with the payoff table:

A \ B	b_1	b_2
$\mathcal{q}_1$	$\$\frac{30}{4}$	\$40
$\mathcal{q}_2$	\$10	\$ 0

and the solution is the strategies:

A: $\left[\frac{4}{17} \quad \frac{13}{17} \right]$

B: $\left[\frac{16}{17} \quad \frac{1}{17} \right]$

f(3, 3) = $\$\frac{160}{17}$

The two different approaches discussed above give the same result; in the initial decision A should play 1 and 2 in the ratio $\frac{4}{17}:\frac{13}{17}$, while B should play 1 and 2 in the ratio $\frac{16}{17} : \frac{1}{17}$, and the value of the game is $\$ \frac{160}{17}$.

From the above example, one may have the idea how dynamic programming has been developed and formulated. The general mathematical formulation of dynamic programming and the formulation of dynamic programming applied to linear multivariable digital control systems are in Appendex II and III respectively.

4.3 Characterisitcs of Dynamic Programming:

(i) The Principle of Optimality underlies the method of dynamic programming. This principle was formulated by R. Bellman. It states that an optimal policy has the property that, whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. In other words, "do the best you can in terms of where you are". The Principle of Optimality reduces the N-stages decision problem into a sequency of N-single-stage decision problems

(ii) The solution of the dynamic programming proceeds in the reverse direciton: We first solve the final or last decision, since this solution requires no knowledge of the effects of last decisions. From this solution for the last decision, we turn to the next-to-last decision, which can now be solved because we know the payoffs to be expected from the final play. In this way, we iteratively continue the solution and finally we reach the initial decision.

(iii) Dynamic Programming is specially designed for use with a digital computer.

4.4 Advantages of Dynamic Programming:

Dynamic programming has the following advantages:

(a) It is a potentially powerful tool.

(b) It provides an elegant treatment of linear multivariable control systems.

(c) The addition of constraints on the control action or decision does not increase the difficulty of the problem,

(d) The solution of the control problem does not depend upon the linearity of the system or the nature of the performance index, and

(e) The computers do the tedious numerical calculations

4.5 The First Order System:

(i) The problem:·

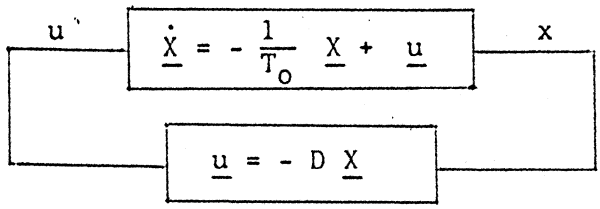


Figure 4-1 A Simple First Order Control System

the plant dynamics $\dot{\underline{X}} = -\frac{1}{T_o} \underline{X} + \underline{u}$

and the controller $\underline{u} = - D \underline{X}$ where T_o is the plant parameter varying in a specified range.

The problem is to determine the forcing function u such that the performance criterion

$$S = \int_o^T (X^2 + \lambda u^2) \, dt$$

is minimized with the initial condition $X_o = 1.0$.

Where T is some specified interval of time and usually should be the solution time (the time required for the system to damp out all transient is commonly called the solution time).

λ is a weighting factor which compares the relative importance of minimizing X^2 and u^2. In the example T and λ are randomly taken as 0.3 and 0.01 respectively.

(ii) Plant parameters and their probability of occurrence:

The value of the plant parameter T_o is difficult to measure accurately. Hence T_o is considered varying within the range 0.6 to 1.4. It is also assumed to have a uniform probability distribution or Gaussian distribution as shown in Figure 4-2.

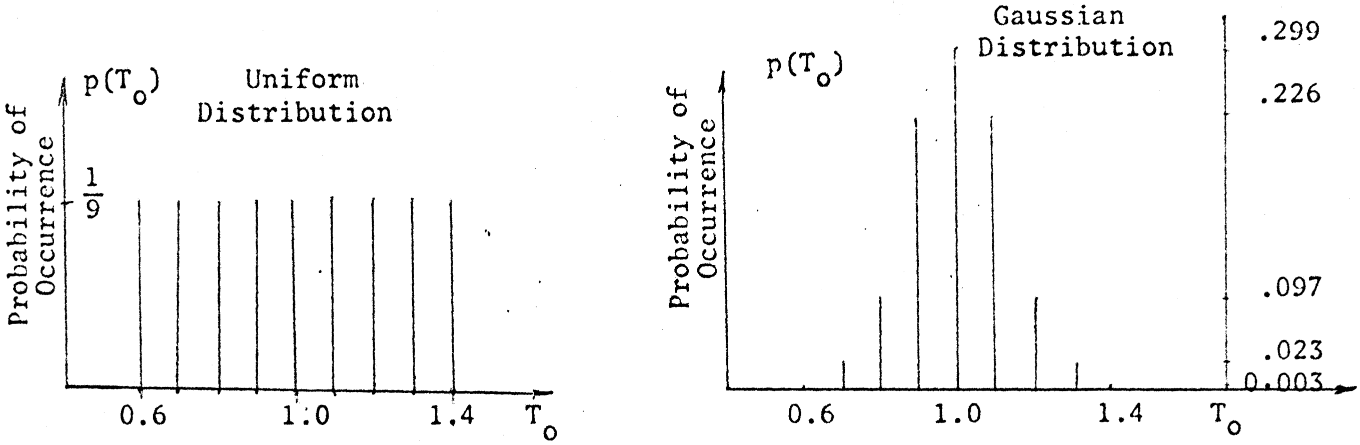


Figure 4-2: Probibility of Occurrence of the Plant Parameter T_o
(NOT TO SCALE)

Each of these values has probability of occurrence 1/9 when it is uniformly distributed. As for gaussian distribution, the location of the peak will be $T_o = 1.0$ and the standard deviation $\sigma = \frac{0.4}{3}$ (i.e. $3\sigma = 0.4$). It means that the probability of the parameter T_o existing outside the range 0.6 to 1.4 is only 0.003. The normalized probability of occurrence has been calculated and tabulated as follows:

T_o	$p(T_o)$ probability of occurrence
0.6	.00320
0.7	.02379
0.8	.09722
0.9	.22605
1.0	.29941
1.1	.22605
1.2	.09722
1.3	.02377
1.4	.00320

Table 4-1: The Probability of Occurrence of the Plant Parameter T_o with Gaussian Distribution

(iii) The optimized controller:

Computer program for calculating the optimized controller D by dynamic programming has been written. Results are as tabulated in the following Table 4-2:

T_o	D The Optimal Controller
0.6	8.3243
0.7	8.5181
0.8	8.6739
0.9	8.7921
1.0	8.8915
1.1	8.9683
1.2	9.0398
1.3	9.0999
1.4	9.1453

Table 4-2: The Optimal Controller D of the Second Order System

With uniform probability of occurrence of T_o, the expected value of the controller D gives

$$\sum_{i=1}^{9} P(T_i)\, D(T_i) = \frac{1}{9}\left[\sum_{i=1}^{9} D(T_i)\right] = 8.8377$$

While if the probability of occurrence of T_o is Gaussian, the expected value of the controller D will give

$$\sum_{i=1}^{9} P(T_i)\, D(T_i) = 8.8941$$

A comparison can now be made:

optimized D = 8.8915 $T_o = 1.0$

expected D = 8.8377 uniformly distributed T_o

expected D = 8.8941 Gaussian distributed T_o

As we know, in practice the designer cannot afford the luxury of time in making accurate measurement or doing tedious calculations. The above values are very close to each other. Hence it is better to choose the value of the optimized D at the middle of the rangeof the varying parameter as the functional D. The functional D is the one which will be automatically modified in the controller while the measurement indicates that the plant parameter lying in the given range. In the example, the functional D is 8.8915 while T_o varies between 0.6 to 1.4.

(v) Performance criterion & relative sensitivity:

The performance of a control system is generally a function of stability, sensitivity, accuracy and transient response. The exact specifications are directed by the required system performance. Certain characteristics are more important in some systems than in others. Transient response is by far the most important consideration for all the physical systems and designers of such systems are often faced with the problem of optimizing the transient behaviour [20].

From the modern optimal control and adaptive control theory view point [21], the performance criterion S can be considered as the loss function or cost function, representing a measure of the instantaneous change from ideal per formance.

The performance criterion of the first order system as stated previously is:

$$S = \int_0^{0.3} (X^2 + 0.01\, u^2)\, dt$$

By considering the plant dynamics

$$\underline{\dot{X}} = -\frac{1}{T_o}\,\underline{X} + \underline{u}$$

and the controller

$$\underline{u} = - D\underline{X}$$

With the initial condition $X_o = 1.0$ we obtain:

$$S = \left[\frac{1 + 0.01\, D^2}{2(D + 1/T_o)} \right] \left[1 - e^{-0.6(D + \frac{1}{10})} \right] \quad ----(4.6)$$

For each value of T_o, the performance criterion S has been calculated by substituting the value of the optimized D corresponding to each T_o (from Table 4-3) into equation (4.6). The results are denoted by the optimal S^* as shown in the second column of the following table.(4-3). The calculations have been repeated with the functional D (8.8915) for each T_o; the results are as in the third column of the table and denoted by actual S.

T_o	The Optimal S^*	The Actual S	Relative Sensitivity $S^R=(S - S^*)/S^*$
0.6	0.084518	0.084659	0.00167
0.7	0.086523	0.086575	0.00058
0.8	0.088062	0.088068	0.000068
0.9	0.089280	0.089285	0.000056
1.0	0.090272	0.090272	0
1.1	0.091089	0.091096	0.000077
1.2	0.091783	0.091794	0.00012
1.3	0.092371	0.092392	0.00023
1.4	0.092875	0.092918	0.00046

Table 4-3: Relative Sensitivity of the First Order System

The last column of the above table is the relative sensitivity which was introduced recently by Rohrer and Sobral [22] for optimal control systems. For a set of plant parameters T_o the relative sensitivity of the controller D is defined to be the difference between the actual value of the performance index and that which would be obtained if the control were the optimal for plant parameters T_o (divided by the optimal performance index for normalization).

$$S^R [T_o, D] = \frac{S[T_o, D] - S^*[T_o, D^*]}{S^*[T_o, D^*]}$$

One of the advantages of the relative sensitivity is that it is always a positive number. Moreover the relative sensitivity reduces to zero at the value of plant parameters T_o for which the controller D is the optimal one.

The Table 4-3 shows that the relative sensitivity for controller D to plant parameter is only up to 0.00167. We may conclude that in a first order system the controller parameter is not sensitive to the change of a plant parameter. Similar conclusions have been drawn by Barnett and Storey [23] recently. From this example we also can see the application of dynamic programming to control systems.

The logic of considering the higher order systems is exactly the same as for the first order system. Hence, in the following examples most of the explanations mentioned above will not be repeated. For the same reason, only the second order system flow charts and computer programmes are given in the appendix.

4.6 The Second Order System:

(i) The problem:

In order to compare the results obtained by Dorato and by Rohrer, we consider the same problem:

The plant dynamic

$$\underline{\dot{X}} = B\,\underline{X} + C\,\underline{u}$$

where

$$B = \begin{bmatrix} 0 & 1 \\ 0 & B_{22} \end{bmatrix} \qquad C = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$0 \leq B_{22} \leq 2$$

With the specified performance criterion:

$$S = \int_0^{12} (X_1^2 + X_2^2 + u^2)\ dt$$

and the given initial condition

$$X_1(0) = 1.0$$

$$X_2(0) = 0.0$$

(ii) Results:

B_{22}	The Optimal Controller $\bar{D}$: D_1	D_2
0	0.98283	1.71219
0.2	0.98468	1.52675
0.4	0.98632	1.36322
0.6	0.98774	1.22048
0.8	0.98898	1.09678
1.0	0.99005	0.99003
1.2	0.99097	0.89804
1.4	0.99176	0.81870
1.6	0.99244	0.75011
1.8	0.99302	0.69057
2.0	0.99351	0.63865

Table 4-4: The Optimal Controller D of the Second Order System

B_{22}	The Optimal S^*	The Actual S	$S^R = \frac{S - S^*}{S^*}$
0	1.74209	2.02024	0.15966
0.2	1.75358	1.88125	0.07280
0.4	1.78767	1.84044	0.02952
0.6	1.84306	1.86065	0.00954
0.8	1.91783	1.92119	0.00175
1.00	2.00999	2.00999	0.00000
1.2	2.11705	2.11910	0.00097
1.4	2.23714	2.24365	0.00241
1.6	2.36800	2.37978	0.00497
1.8	2.50823	2.52527	0.00679
2.0	2.65397	2.67574	0.00820

Table 4-5: Relative Senstivity of the Second Order System

As Table 4-4 shows, D_1 is approximately equal to 1.0. The optimized D_2 varies from 0.63865 to 1.712 in Table 4-4 while it varies from 0.65 to 1.73 in Figure 4 of [22]. Now let us make a comparison:

(from Table 4-4)

B_{22}	The Optimized D_2
0	1.71219
0.4	1.36332
0.6	1.22048
1.0	0.99003
2.0	0.63865

(from Figure 4 of [22])

B_{22}	D_2
0.0	1.73
0.5	1.3
1.0	1.0
2.0	0.65

The results above agree as expected. The optimal S^* and the actual S are plotted against the plant parameter B_{22} as shown in Figure 4-3 which shows the difference. The relativity sensitivity vs. B_{22} is in figure 4-4, which is exactly the same as the curve of $D_2 = -1.0$ in figure 4 of [22]. From the example, it is obvious that dynamic programming is powerful. It saves time in doing the tedious calculations by computer and gives the accurate results. It also does not need the assumption (the system is

FIGURE 4-3: The Optimal S^* and the Actual S Against B_{22} of the Second Order System

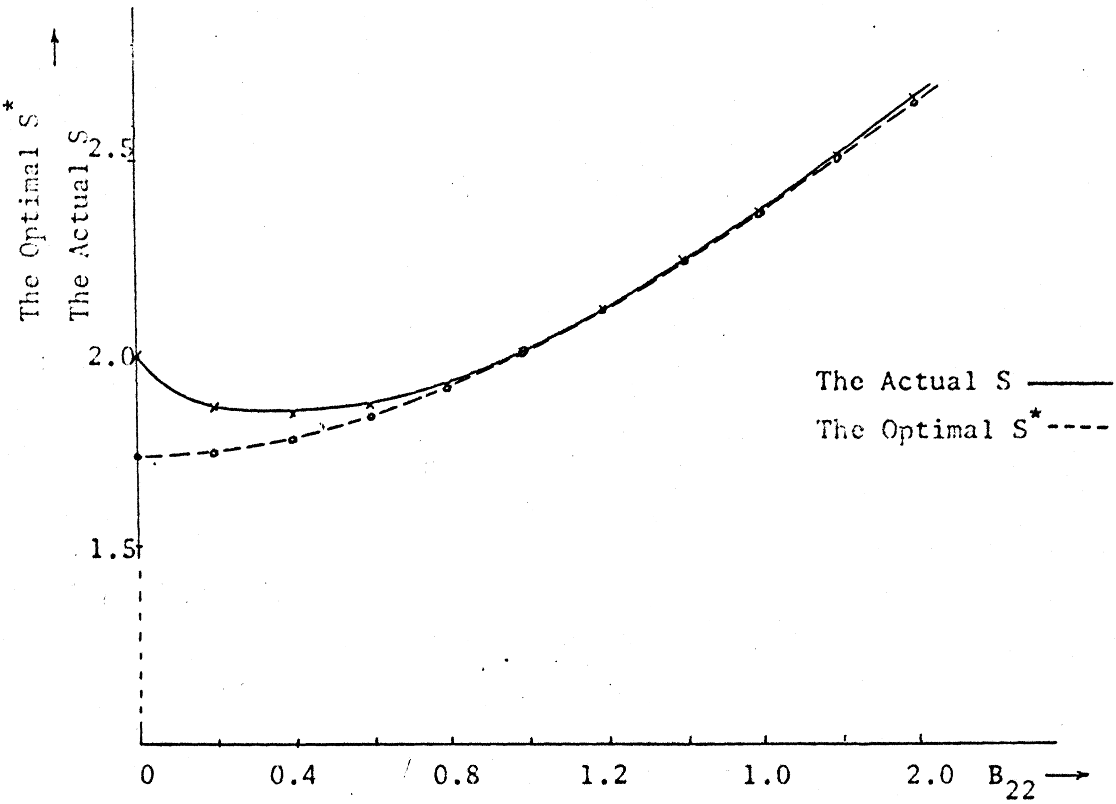
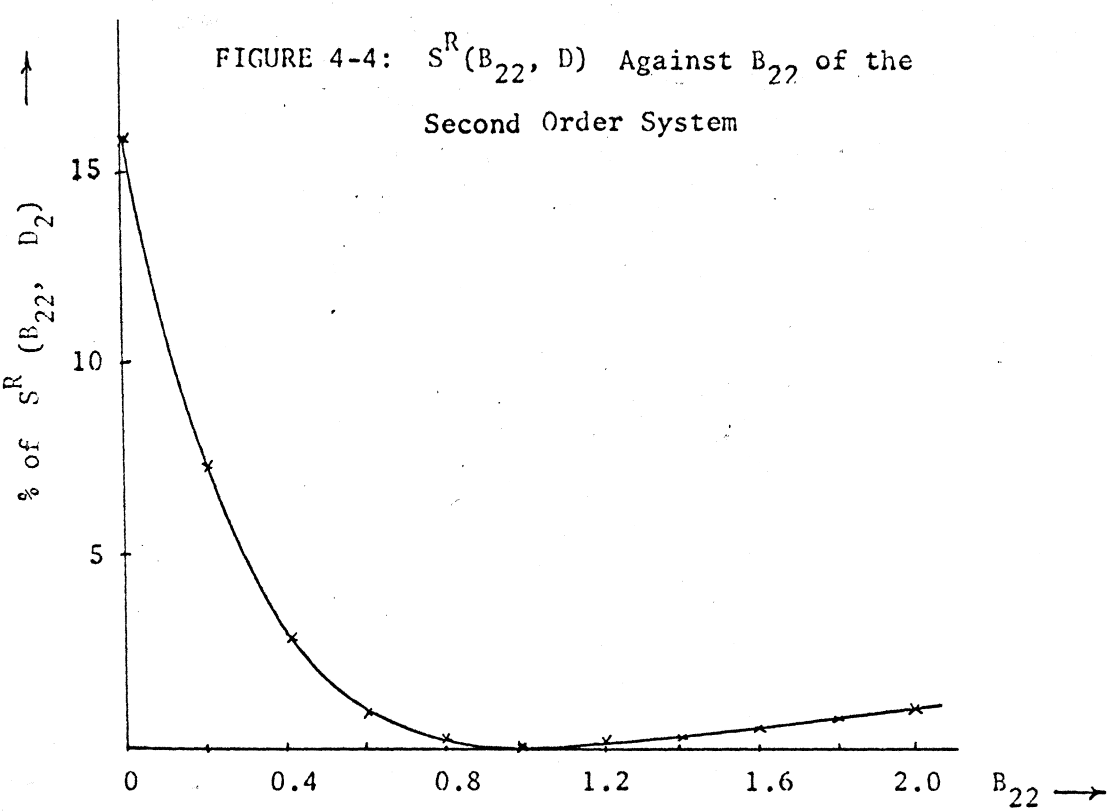


FIGURE 4-4: $S^R(B_{22}, D)$ Against B_{22} of the Second Order System



optimized when the controller parameter is equal to the plant parameter) which is very important in cases of using decision theory.

4.7 The Third Order System:

(i) The problem:

The plant dynamic,

$$\dot{\underline{X}} = B\,\underline{X} + C\,\underline{u}$$

where u is the forcing function

$$B = \begin{bmatrix} 0 & B_{12} & B_{13} \\ -1 & -3 & -2 \\ -1 & -2 & -2 \end{bmatrix} , \qquad C = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$B_{12} = B_{13} = B_V$ and $0.4 \leqslant B_V \leqslant 1.6$. If we consider the minimizing of u^2 to be as important as the minimizing X_1^2, X_2^2 and X_3^2; and the time T is randomly chosen as 4. Then the performance criterion is:

$$S = \int_0^4 \left[X_1^2 + X_2^2 + X_3^2 + u^2\right] dt$$

The given initial condition is:

$$X_1(0) = 1.0$$

$$X_2(0) = 1.0$$

$$X_3(0) = 0.0$$

(ii) Results:

B_V	The Optimal Controller D $[D_1$	D_2	$D_3]$
0.4	0.31978	0.04342	0.24265
0.6	0.37836	0.06483	0.26167
0.8	0.39712	0.08492	0.27997
1.0	0.40523	0.10396	0.29698
1.2	0.40815	0.12237	0.31309
1.4	0.40918	0.14039	0.32871
1.6	0.40951	0.15811	0.34413

Table 4-6: The Optimal Controller D of the Third Order System

B_v	The Optimal S^*	The Actual S	$S^R = \frac{S - S^*}{S^*}$
0.4	3.48933	3.51239	.00661
0.6	2.92574	2.92732	.00054
0.8	2.54893	2.54903	.00004
1.0	2.29459	2.29459	0
1.2	2.11782	2.11794	.00006
1.4	1.99049	1.99082	.00017
1.6	1.89618	1.89686	.00036

Table 4-7: Relative Sensitivity of the Third Order System

4.8 The Fourth Order System:

(i) The problem:

Two forcing functions u_1 and u_2 as well as two varying plant parameters B_{21} and B_{22} are considered in the fourth order system with the plant dynamics:

$$\dot{\underline{X}} = B\,\underline{X} + C\,\underline{u}$$

where

$$B = \begin{bmatrix} -1 & 0 & 0 & 0 \\ B_{21} & B_{22} & 0 & 2 \\ 0 & +\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$1.4 \leq B_{21} \leq 2.6$ $\qquad$ $-2.6 \leq B_{22} \leq -1.4$

and

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & \frac{1}{2} \\ 0 & 0 \end{bmatrix}$$

The performance criterion is:

$$S = \int_0^4 (X_3^{\,2} + 0.05\,u_1^{\,2} + 0.05\,u^2)\,dt$$

and the given initial condition:

$X_1(0) = 1.0 \qquad X_2(0) = 1.0$

$X_3(0) = 0.0 \qquad X_4(0) = 0.0$

(ii) Results:

The plant parameters B_{21} and B_{22} are assumed to have uniform probability distribution or Gaussian distribution, and the values of the expected controller with the plant parameters having different probability of occurrence were calculated and tabulated as follows:

The Probability Distribution of B_{21} and B_{22}	The Controller D $\begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \end{bmatrix}$			
In the Middle of the Range	0.22722 0.23339	0.13944 0.40916	0.45983 3.51793	0.50933 0.30824
B_{21} Uniform B_{22} Uniform	0.24117 0.23247	0.14635 0.41174	0.45795 3.51729	0.52307 0.30710
B_{21} Uniform B_{22} Gaussian	0.22941 0.23188	0.14006 0.40929	0.45683 3.51398	0.50500 0.29363
B_{21} Gaussian B_{22} Uniform	0.23826 0.23330	0.14722 0.41142	0.46058 3.51411	0.52990 0.30658
B_{21} Gaussian B_{22} Gaussian	0.22899 0.23270	0.14084 0.40898	0.45846 3.51071	0.51168 0.30744

TABLE 4-8: The Optimal Controller D of the Fourth Order System

The above matrices are very nearly equal to each other. Hence, the optimal controller in the middle of the specified range is considered as the functional D ignoring the probability of occurrence of the plant parameter as in the first order system.

B_{21}	B_{22}	The Optimal S^*	The Actual S	$S^R = \frac{S - S^*}{S^*}$
1.4	-2.6	0.16643	0.17902	0.07565
	-2.4	0.18824	0.19847	0.05435
	-2.2	0.21384	0.22157	0.03615
	-2.0	0.24594	0.25099	0.02053
	-1.8	0.28466	0.28725	0.00910
	-1.6	0.33438	0.33515	0.00230
	-1.4	0.39615	0.39701	0.00217
1.6	-2.6	0.18689	0.19626	0.05014
	-2.4	0.21121	0.21819	0.03305
	-2.2	0.23950	0.24412	0.01929
	-2.0	0.27481	0.27713	0.00844
	-1.8	0.31712	0.31774	0.00196
	-1.6	0.37102	0.37134	0.00086
	-1.4	0.43749	0.44063	0.07177
1.8	-2.6	0.20746	0.21404	0.03172
	-2.4	0.23412	0.23845	0.01849
	-2.2	0.26487	0.26715	0.08608
	-2.0	0.30335	0.30397	0.00204
	-1.8	0.34900	0.34906	0.00017
	-1.6	0.40688	0.40868	0.00442
	-1.4	0.47735	0.48536	0.01678
2.0	-2.6	0.22916	0.23328	0.01798
	-2.4	0.25810	0.26029	0.00848
	-2.2	0.29147	0.29216	0.00237
	-2.0	0.33285	0.33285	0
	-1.8	0.38161	0.38261	0.00262
	-1.6	0.44294	0.44830	0.01210
	-1.4	0.51690	0.53259	0.03035
2.2	-2.6	0.25068	0.25295	0.00905
	-2.4	0.28182	0.28262	0.00284
	-2.2	0.31756	0.31762	0.00019
	-2.0	0.36165	0.36225	0.00166
	-1.8	0.41327	0.41678	0.00844
	-1.6	0.47791	0.48883	0.02284
	-1.4	0.54498	0.58086	0.06583
2.4	-2.6	0.27311	0.27406	0.00348
	-2.4	0.30649	0.30664	0.00049
	-2.2	0.34449	0.34489	0.00116
	-2.0	0.39112	0.39364	0.00380
	-1.8	0.44553	0.45326	0.01735
	-1.6	0.51283	0.53152	0.03640
	-1.4	0.59261	0.63145	0.06554

B_{21}	B_{22}	The Optimal S^*	The Actual S	$S^R = \frac{S - S^*}{S^*}$
2.6	-2.6	0.29515	0.29547	0.00108
	-2.4	0.33043	0.33077	0.00103
	-2.2	0.37068	0.37246	0.00480
	-2.0	0.41967	0.42537	0.01358
	-1.8	0.47651	0.48999	0.02828
	-1.6	0.54637	0.57468	0.05181
	-1.4	0.62874	0.68284	0.08604

TABLE 4-9: Relative Sensitivity of the Fourth Order System

4.9 Discussion:

The following table was drawn from the results of the above examples.

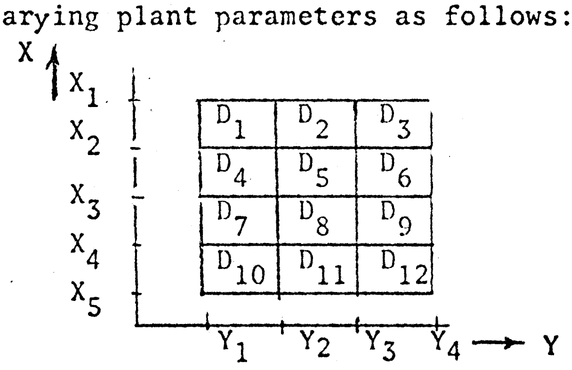
	Percentage Variation of the plant parameter (refer to its middle value)		The highest percentage Relative Sensitivity
1st Order System (one varying plant parameter)	40%		0.167%
2nd Order System (one varying plant parameter)	100%		15.97 %
	80%		7.25 %
	60%		2.95 %
	40%		.95 %
3rd Order System (Two varying plant parameters)	both 60%		0.66 %
4th Order System (Two varying plant parameters)	B_{21}	B_{22}	
	30%	30%	8.6 %
	30%	20%	5.4 %
	30%	10%	3.6 %
	30%	0%	2.0 %
	0%	10%	.26 %
	0%	20%	1.2 %
	0%	30%	3.03 %

From the above, the following points are noticed:

(i) In general the controller is not sensitive to the plant parameter variations when the percentage variation of the plant parameters (refer to its middle value) is below a certain limit. This limit is a function of the plant dynamics of the particular system in question (e.g., 40% of the plant parameter variation for the first order and second order systems, 60% in the third order system and 20% of one parameter variation in the fourth order system if the requirement for the percentage relative sensitivity is below 1.2%).

(ii) In case of high relative sensitivity of the controller to the plant parameters, the relative sensitivity reduces sharply in response to the decrease of the variation of the plant parameters.

(iii) In the second order system the percentage relative sensitivity is 15.97% for one hundred percent variation of the plant parameter; also in the fourth order system the percentage relative sensitivity is 8.6% while both plant parameters have thrity percent variation. The high sensitivity is not desirable for the system. In order to maintain the optimal performance due to the large variation of the plant parameters, we can, therefore, divide the range of the plant parameter into smaller ranges as required. More values of the optimal controller corresponding to the smaller ranges should be calculated beforehand. This is illustrated by considering two varying plant parameters as follows:



The figures show that the range of one parameter (X) has been divided into four smaller ranges and the other (Y) into three smaller ranges. The twelve different values of the optimal controller have been calculated and stored in the memory system of the computer. According to the measurement of the plant parameters, the corresponding optimal controller will be modified automatically (e.g., D_8 will be the optimal controller when the plant parameter X lies between its values X_3 and X_4 and Y lies between its values Y_2 and Y_3).

CHAPTER V

CONCLUSION

In this thesis a new approach has been suggested to circumvent the difficulty of the identification problem in adaptive control systems. Four examples of different orders have shown the fact that having a proper choice of feedback policy, the optimal controller is relatively insensitive to variations in the plant parameters as long as they lie within some specified ranges. In the examples of the first order and the fourth order systems, the expected optimal controller have been calculated by considering the plant parameters having uniform or Gaussian distribution over the given ranges. It is also found that the expected optimal controller is nearly equal to the optimal controller in the middle of the given ranges. Hence, the probability distribution of the plant parameters can be ignored and the optimal controller in the middle of the range will be the one automatically modified in the system. The limit of the ranges usually depends on the design specifications and the particular system in question. If the ranges were so large that the optimal controller can not maintain the optimal performance of the system, then the range of variations can be divided into smaller ranges as necessary. With this approach, the identification can be carried on more rapidly since it would not be required to compute the optimal controller during normal operation, at the sametime a bigger tolerance may be permitted in making the measurements.

An examination in the example of some recently published work indicates that dynamic programming is preferrable to decision theory. The most significant advantage is that no simplifying assumptions have to be made. Whereas, such assumptions are necessary for the application of decision theory.

The application of dynamic programming can be easily extended to higher order and multivariables control systems. It is also possible to be applied to non-linear and linear time-variant systems. The only limitation appeared to be the requirement of the long computation time and the large storage of the computer.

REFERENCES

1. Aseltine, J. A., Mancini, A. R., and Sature, C. W., "A Survey of Adaptive Control Systems", IRE Trans. on Automatic Control, December, 1958.

2. Li, Y. T., "The Philosophy of Adaptive Control", First International Congress of International Federation on Automatic Control, Moscow, U.S.S.R., June 1960.

3. Gibson, J. E., "Making Sense Out of the Adaptive Principle", Control Engineering, August, 1960.

4. Mishkin, E., and Brown,/ L., "The Adaptive Control Systems", McGraw-Hill Book Co., New York, 1961.

5. Gibson, J. E., "Self-Optimizing or Adaptive Control Systems", Proceedings of the First International Congress of the IFAC, Moscow, 1960.

6. Caruthers, F. P., and Levenstein, H., "Adaptive Control Systems: Proceedings of a Symposium Held in Long Island, New York, October-1963", The MacMillan Co., 1963.

7. Sinha, N. K., "Adaptive Control Systems; A General Classification Scheme", Tech. Report No. 2 Control Theory Group, The University of Tennessee, October, 1964.

8. Padalino, J. J., "Decision Theory Applied to Control System Design", Doctoral Dissertation in Preparation at the Polytechnic Institute of Brooklin, 1963.

9. Hsu, J. C., and Meserve, W. E., "Decision-Making in Adaptive Control Systems", IRE. Trans. on Automatic Control, Vol. AC-7, January 1962.

10. Dorato, P., and Kestenbaum, A., "Application of Game Theory to the Sensitivity Design of Optimal Systems", IEEE Trans. on Automatic Control, February, 1967.

11. Neumann, J. V., and Morgenstern, O., "Theory of Games and Economic Behaviour", Princeton University Press, 1953.

12. Abramson, N. M., "Seminar on Decision Theory and its Application to Communication Theory", IBM Research Paper RJ-MR-8, January, 1958.

13. Weiss, L., "Statistical Decision Theory", McGraw-Hill Book Company, 1961.

14. Karlin, S., "Mathematical Methods and Theory in Games, Programming, and Economics", Vol. 2, Addison-Wesley Publishing Co., 1959.

15. Luce, R. D., and Raiffa, H., "Games and Decisions", John Wiley & Sons, Inc., New York, 1957.

16. Bellman, R., "Dynamic Programming", Princeton University Press, Princeton, New Jersey, 1957.

17. Bellman, R.,"Adaptive Control Process", Princeton University Press, Princeton, New Jersey, 1961.

18. Bellman, R. and Kalaba, R., "Dynamic Programming and Modern Control Theory", Academic Press, New York, 1965.

19. Norton, A. M. M., "Introduction to Variational Methods in Control Engineering", Pergamon Press, 1965.

20. Graham, D., and Lathrop, L. C., "The Synthesis of Optimum Transient Response: Criteria and Standard Forms", IEEE Trans. on Applications and Industry, November, 1953.

21. Shinners, S. M., "Control System Design", Chapter 10 and 11, John Wiley & Sons, New York, 1964.

22. Rohrer, R. A., and Sobral, M., "Sensitivity Considerations in Optimal System Design", IEEE Trans. on Automatic Control, January, 1955.

23. Barnett, S., and Storey, C., "Insensitivity of Optimal Linear Control System to Persistent Changes in Parameters", Int. J. Control, 1966, Vol. 4, No. 2.

APPENDIX I

General Statement of the Minimax Theorem

(i) There are two players, A and B.

(ii) A has a set $(\alpha_1, \alpha_2, ---, \alpha_m)$ of m pure strategies.

(iii) B has a set $(\beta_1, \beta_2, ---, \beta_n)$ of n pure strategies.

(iv) Associated to each pair of strategies (α_i, β_j) is a pay off a_{ij} unit from player B to A. Hence, the values to A and B of the strategy pair (α_i, β_j) are a_{ij} and $-a_{ij}$.

(v) These values sum to zero for every (α_i, β_j) pair, hence the game is called zero - sum.

(vi) Player A may adopt a mixed strategy by employing α_1 with probability X_1, α_2 with probability $x_2 \ldots$, α_m with X_m,

where $\sum_{i=1}^{m} X_m = 1$ and $X_i \geq 0$

Such a strategy is symbolically represented by

$XX = (X_1\alpha_1, X_2\alpha_2, \ldots, X_m\alpha_m)$ The set of all mixed strategies for player A is denoted by X_m.

(vii) Similarly the mixed strategies for B is denoted by

$yy = (y_1\alpha_1, \ldots., y_n\alpha_n)$ where $\sum_{i=1}^{n} y_n = 1$ $y_i \geq 0$

and the set of all mixed strategies for player B is represented by Y_n.

(viii) For each mixed strategy pair (XX, YY), the pay off M(XX, YY) to A is defined to be:

$$M(XX, YY) = \sum_{i=1}^{m} \sum_{j=1}^{n} X_i \, a_{ij} \, y_j$$

$$= \sum_{j=1}^{n} y_i \, (\sum_{i=1}^{m} X_i \, a_{ij})$$

$$= \sum_{i=1}^{m} X_i \, (\sum_{j=1}^{n} a_{ij} \, y_j)$$

The symbol

$$M(\alpha_i, YY) = \sum_{j=1}^{n} a_{ij} \, y_j$$

means the pay off to A when A uses pure strategy α_i and B uses YY. Similary when A uses XX and B uses β_j, the pay off is

$$M(XX, \beta_j) = \sum_{i=1}^{m} a_{ij} \, X_i$$

(ix) The player A aims to select a mixed strategy XX from X_m so as to maximize his return M(XX, YY) and B's aim is to minimize the return to A M(XX, YY) by choosing a mixed strategy YY.

(x) The rules of the game require that each player chooses his strategy in complete ignorance of his opponent's selection.

(xi) The minimax theorem states mathematically:

$$\min_{YY} M(XX^{(0)}, YY)$$

$$= \max_{XX} \min_{YY} M(XX, YY)$$

$$= \min_{YY} \max_{XX} M(XX, YY)$$

$$= \max_{XX} M(XX, YY^{(0)})$$

$$= M(XX^{(0)}, YY^{(0)})$$

where $XX^{(0)}$ is the maximum strategy chosen by A

$YY^{(0)}$ is the minimax strategy chosen by B

M is the payoff function.

APPENDIX II

GENERAL FORMULATION OF DYNAMIC PROGRAMMING

An m^{th} order system can be described by m state variables, let

$$\underline{X} = [X_1, X_2, \ldots, X_m] \qquad \text{----(1)}$$

The system is subject to S forcing functions, let,

$$\underline{u} = [u_1, u_2, \ldots, u_s] \qquad \text{----(2)}$$

where $S \leq m$, and the m equations of the system are, in vector notation,

$$\dot{\underline{X}} = \underline{f}\ (\underline{X}, \underline{u}, t) \qquad \text{----(3)}$$

Such equations are, in general, nonlinear and time variant. The equivalent discrete system is described by a vector transition equation which can be calcualted from equation (3) in sepcific cases; let it be,

$$\underline{X}_{n+1} = \underline{X}_n + h\ \underline{\phi}_n\ (\underline{X}_n, \underline{u}_n) \qquad \text{----(4)}$$

where

$$\lim_{n \to 0} \underline{\phi}_n\ (\underline{X}_n, u_n) = \underline{f}\ (\underline{X}, \underline{u}, nh)$$

The performance criterion or cost funciton is a series of N terms as following:

$$\sum_{n=0}^{N-1} g_n\ (\underline{X}_n, \underline{u}_n)$$

When the $\underline{u}_n$ have been chosen, each X_n is calculable from X_{n-1} by equation (4); then the series is only a function of initial state $\underline{X}_o$. The performance criterion is (a series N terms is used):

$$V_N(\underline{X}_o) = \min_{\underline{u}_o \ldots \underline{u}_{N-1}} \sum_{n=0}^{N-1} g_n(\underline{X}_n, \underline{u}_n)$$

For an n-stage process it can be written:

$$V_n(\underline{X}_o) = \min_{\underline{u}_o \ldots \underline{u}_{n-1}} \left[g_o(\underline{X}_o \ \underline{u}_o) + \sum_{i=1}^{n-1} g_i(\underline{X}_i, \underline{u}_i) \right]$$

Since the first stage is only affected by u_o, therefore;

$$V_n(\underline{X}_o) = \min_{\underline{u}_o} \left[g_o(\underline{X}_o, \underline{u}_o) + \min_{\underline{u}_1 \ldots \underline{u}_{n-1}} \sum_{i=1}^{n-1} g_i(\underline{X}_i \ \underline{n}_i) \right]$$

However, the second term in brackets is $V_{n-1}(x_1)$, the performance criterion for a (n-1) stage process starting at X_1. Hence we may write:

$$V_n(\underline{X}_o) = \min_{\underline{u}_o(n)} \left[g_o(\underline{X}_o, \underline{u}_o) + V_{n-1}(\underline{X}_1) \right] \qquad \text{----(5)}$$

The $\underline{u}_o(n)$ has been so written as a reminder that the calculated $\underline{u}_o$ depends on the number of stages in the process. The $\underline{X}_1$ of the equation can be determined by equation (4) with n = 0. Thus,

$$\underline{X}_1 = \underline{X}_o + h \ \underline{\phi_o} \ (X_o, u_o) \qquad \text{----(6)}$$

Equation (5), with equation (6), is the basic equation of dynamic programming. The desired $\underline{u}_o(N)$ is computed by using (5) and (6) in an iterative process.

APPENDIX III

FORMULATION OF DYNAMIC PROGRAMMING APPLIED TO LINEAR MULTIVARIABLE DIGITAL CONTROL SYSTEMS

The linear multivariable control system is described by a set of m first-order linear differential equations:

$$\dot{\underline{X}} = B\underline{X} + C\underline{u} \qquad \text{----(1)}$$

$\underline{X}$ is a column vector of the m state variables and $\underline{u}$ of the s forcing functions. B is an (m x m) and C an (m x s) matrix. The forcing functions are to be held constant throughout each periodic intervals of time h and changed in a step manner at the sampling instants. X(n) is the state vector at the beginning of the $(n+1)^{th}$ interval and $\underline{u}(n)$ is the vector of forcing function during the same interval. Since it is a linear system, the state vector at the end of the $(n+1)^{th}$ interval is linearly dependent on $\underline{X}(n)$ and $\underline{u}(n)$, thus,

$$\underline{X}(n+1) = F\,\underline{X}(n) + E\,\underline{u}(n) \qquad \text{-----(2)}$$

where F is an (m x m) matrix and E an (m x s) matrix. F and E can be obtained by using the solution of equations (1) in matrix form. Starting at X(0), X(t=h) is given by

$$X(h) = \exp(hB)\,X(0) + \int_0^h \exp\left[(h-t)B\right] Cu(0)\,dt$$

where

$$\exp(hB) = \sum_{n=0}^{\infty} \frac{1}{n!} (hB)^n$$

The performance criterion is

$$\sum_{n=0}^{N-1} \left[\underline{X}'(n)\ A\ \underline{X}(n) + \underline{u}'(n)\ H\ \underline{u}(n) \right] h$$

where the prime symbols denote transposed matrices; A is a symmetric and H a diagonal matrix. Similar to the logic in Appendix II, the performance criterion for an n-stage process can be written:

$$V_n \left[\underline{X}(0)\right] = \min_{\underline{u}(0)} \left\{ \left[\underline{X}'(0)\ A\ \underline{X}(0) + \underline{u}'(0)\ H\ \underline{u}(0)\right] h + V_{n-1}\left[\underline{X}(1)\right] \right\}$$

where

$$V_{n-1}\left[\underline{X}(1)\right] = \min_{u(1)..u(n-1)} \sum_{n=1}^{n-1} \left[\underline{X}'(n)\ A\ \underline{X}(n) + \underline{u}'(n)\ H\ \underline{u}(n)\right] h \quad \text{----(3)}$$

V_n is expressible as a quadratic form in $\underline{X}(0)$

$$V_n \left[\underline{X}(0)\right] = \underline{X}'(0)\ G_n\ \underline{X}(0) \quad \text{----(4)}$$

where G_n is a symmetric (mxm) matrix. Substitute (4) and (2) into (3)

$$\underline{X}'(0)\ G_n\ \underline{X}(0) = \min_{\underline{u}(0)} \left\{ \left[\underline{X}'(0)\ A\ \underline{X}(0) + \underline{u}'(0)\ H\ \underline{u}(0)\right] h + \left[F\ \underline{X}(0) + E\ \underline{u}(0)\right]'\ G_{n-1} \left[F\ \underline{X}(0) + E\ \underline{u}(0)\right] \right\} \quad \text{---(5)}$$

u(0) is chosen to minimize the right-hand side of equation (5). In order to minimize it, differentiate with respect to the s forcing functions.

$$h\ \underline{u}'(0)\ H = - \left[F\ \underline{X}(0) + E\ \underline{u}(0)\right]'\ G_{n-1}\ E$$

Take the transpose of the matrix equation, and solve for $\underline{u}(0)$

$$\underline{u}(0) = - \left[h\ H + E'\ G_{n-1}\ E\right]^{-1} E'\ G_{n-1}\ F\ \underline{X}(0)$$

Put

$$\underline{u}((\quad - D_n \underline{X}(0) \qquad ----(6)$$

where

$$D_n = [h H \cdot \quad {}_{n-1} E]^{-1} E' \quad F \qquad ----(7)$$

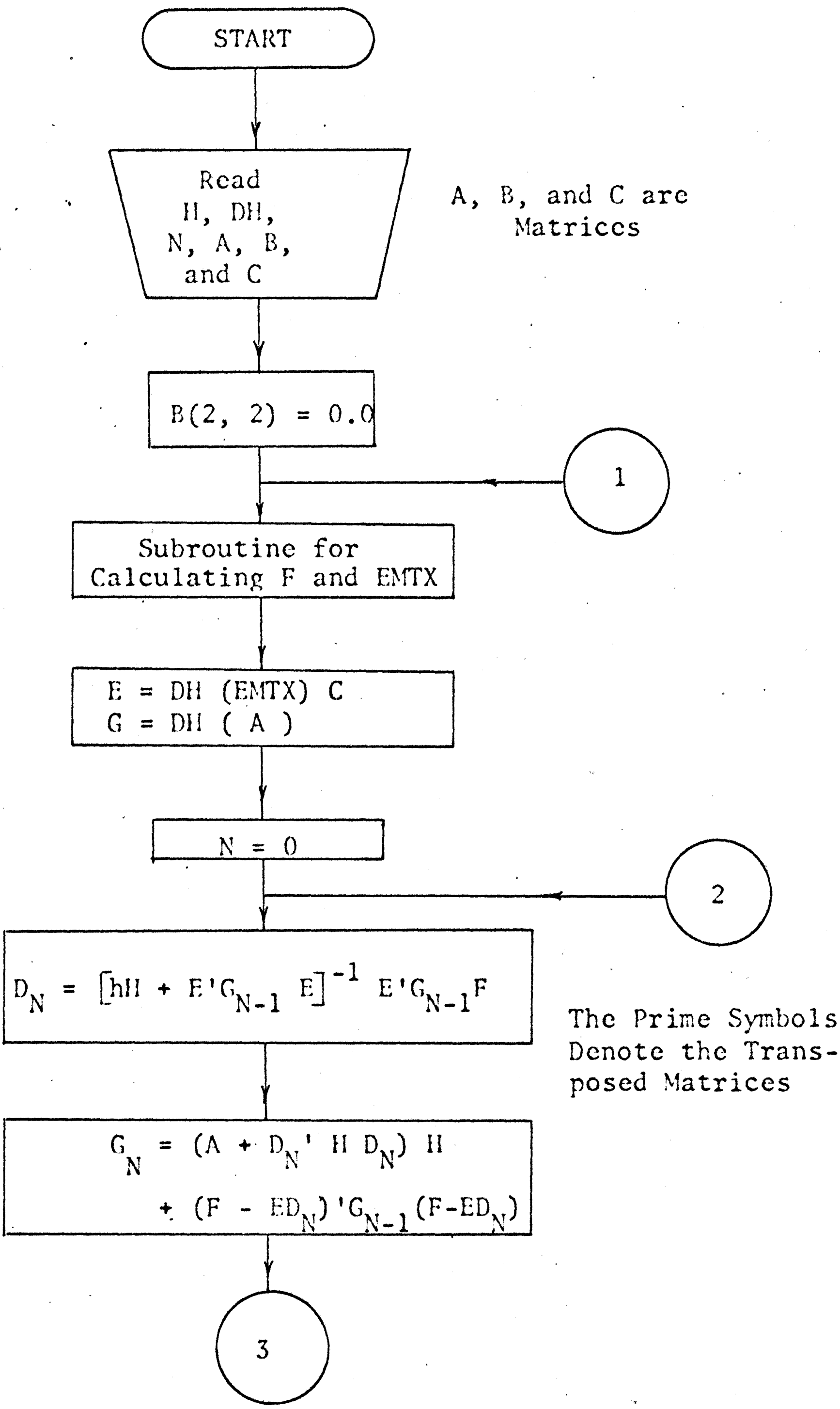
by substituting (6) into (

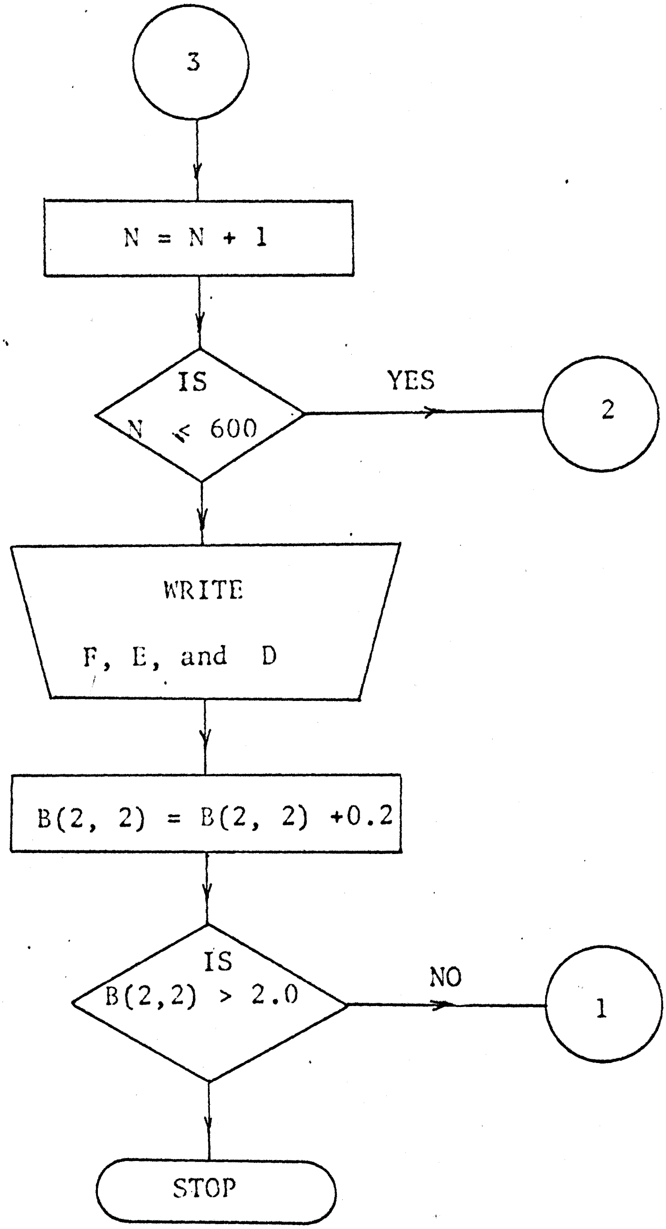
$$+ D_n' H \quad G_{n-1}(F-ED_n) \qquad ----(8)$$

Equations (6), (7), d with t tion define an iterative process t arried out The final result of $\underline{u}(0) = - D_n \underline{X}(0)$ s es the co e it gives the optimum set of s forcing fu s to be u 0.

APPENDIX IV

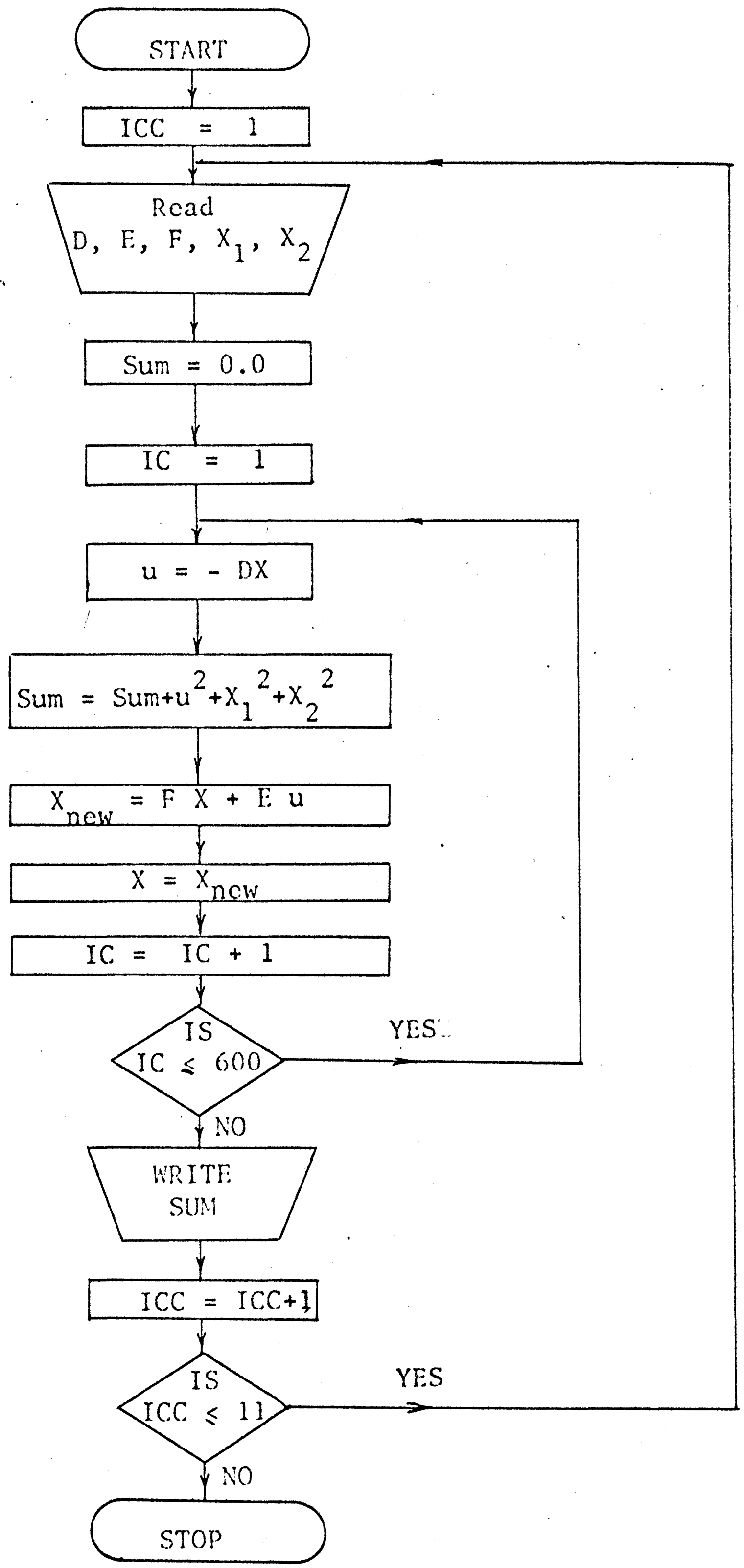
FLOW CHART FOR CALCULATING THE OPTIMAL CONTROLLER OF THE SECOND ORDER SYSTEM




3
N = N + 1
IS
N < 600
YES
2
WRITE
F, E, and D
B(2, 2) = B(2, 2) +0.2
IS
B(2,2) > 2.0
NO
1
STOP

APPENDIX V

FLOW CHART FOR CALCULATING THE PERFORMANCE INDEX OF THE SECOND ORDER SYSTEM



APPENDIX VI

```
$JOB           003512 LOUIS LEE        100    010    030
$IBJOB         NODECK
$IBFTC LEUNG
C CALCULATION OF AN OPTIMAL CONTROLLER BY DYNAMIC PROGRAMMING
C FOR A SECOND ORDER SYSTEM
      DIMENSION F(2,2), B(2,2),E(2,1),C(2,1),A(2,2),D(1,2),
     1 G(2,2),EP(1,2),EPG(1,2),EPGE(1,1),HM(1,1),EPGF(1,2),DP(2,1),
     2VHM(1,1),DPH(2,1),DPHD(2,2),FMED(2,2),FMEDP(2,2),FMG(2,2),
     3FMGX(2,2),EMTX(2,2),AMX(2,2),ED(2,2)
      H=1.0
      DH=0.02
      N=600
      READ (5,10) ((A(I,J),J=1,2),I=1,2)
   10 FORMAT (4F5.1)
      READ (5,11) (C(I,1), I=1,2)
  11  FORMAT(2F5.1)
      WRITE (6,12) ((A(I,J),J=1,2),I=1,2)
      WRITE (6,12) (C(I,1),I=1,2)
  12  FORMAT( 1H ,2F10.5)
      B(1,1)=0.0
      B(1,2)=1.0
      B(2,1)=0.0
      DO1 I1=0,20,2
      B(2,2)=I1
      B(2,2) = -B(2,2)/10.0
      CALL PHI(F,B,2,DH,0,50)
      CALLPHI(EMTX,B,2,DH,1,50)
      DO21 L=1,2
      E(L,1)=0.0
      DO 21 J=1,2
  21  E(L,1) =DH*EMTX(L,J)*C(J,1)+E(L,1)
      DO 30 J=1,2
 30   D(1,J)=0.0
      DO 31 I=1,2
      DO 31 J=1,2
  31  G(I,J)=DH*A(I,J)
      DO 40 I=1,2
  40  EP(1,I)=E(I,1)
      ITCT=0
  200 DO 41 I=1,2
      EPG(1,I)=0.0
      DO 41 J=1,2
  41  EPG (1,I)=EPG(1,I)+EP(1,J)*G(J,I)
      EPGE(1,1)=0.0
      DO 42 J=1,2
  42  EPGE(1,1) = EPGE(1,1)+EPG(1,J)*E(J,1)
  43  HM(1,1) = DH*H+EPGE(1,1)
  25  VHM(1,1) = 1.0/HM(1,1)
```

```
      DO 44 I=1,2
      EPGF(1,I)=0.0
      DO 44 J=1,2
 44   EPGF(1,I)=EPGF(1,I) + EPG(1,J)*F(J,I)
      DO 45 I=1,2
      D(1,I) =0.0
 45   D(1,I)=D(1,I)+VHM(1,1)*EPGF(1,I)
      DO 46 J=1,2
 46   DP(J,1)=D(1,J)
      DO 47 L=1,2
      DPH(L,1)=0.0
 47   DPH(L,1)=DPH(L,1)+DP(L,1) *H
      DO 48 L=1,2
      DO 48 I=1,2
      DPHD (L,I)=0.0
 48   DPHD(L,I)=DPHD(L,I) + DPH(L,1)*D(1,I)
      DO 49 I=1,2
      DO 49 J=1,2
 49   AMX(I,J) = (A(I,J)+DPHD(I,J))*DH
      DO 50 L=1,2
      DO 50 I=1,2
      ED(L,I) =0.0
 50   ED(L,I) = ED(L,I) + E(L,1)*D(1,I)
      DO 51 I=1,2
      DO 51 J=1,2
      FMED(I,J) = F(I,J) -ED(I,J)
 51   FMEDP(J,I)=FMED(I,J)
      DO 52 L=1,2
      DO 52 I=1,2
      FMG(L,I) =0.0
      DO 52J=1,2
 52   FMG(L,I) = FMG(L,I) + FMEDP(L,J)*G(J,I)
      DO 53 L=1,2
      DO53 I=1,2
      FMGX(L,I) =0.0
      DO53 J=1,2
 53   FMGX(L,I) =FMGX(L,I) + FMG(L,J)*FMED(J,I)
      DO 54 I=1,2
      DO54 J=1,2
 54   G(I,J) = AMX(I,J) +FMGX(I,J)
      ITCT =ITCT+ 1
      IF (ITCT .LT.N) GO TO 200
      WRITE(7,100) ((F(I,J),J=1,2),I=1,2)
      WRITE(7,101) (E(I,1),I=1,2)
      WRITE(7,101)  (D(1,J),J=1,2)
 1    CONTINUE
100   FORMAT(4F10.5)
101   FORMAT (2F15.5)
      STOP
      END
```

```
$IBFTC LEE
C CALCULATION OF MATRICES E AND F
      SUBROUTINE PHI(TMTX,AMTX,MSIZE,DT,MON,LMT)
      DIMENSION TMTX(MSIZE,MSIZE),AMTX(MSIZE,MSIZE),
     1PMTX(4,4),FMTX(4,4),UMTX(4,4)
      K=1
      FCTR=1.0
      KK=K+MON
      AK=KK
      FCTR=FCTR*DT/AK
      DO 501 L=1,MSIZE
      DO 501 I=1,MSIZE
      IF(L.EQ.I)GO TO 100
      UMTX(L,I)=0.0
      GO TO 501
  100 UMTX(L,I)=1.0
  501 CONTINUE
      DO 502 L=1,MSIZE
      DO 502 I=1,MSIZE
  502 TMTX(L,I)=UMTX(L,I)+AMTX(L,I)*FCTR
      DO 503 L=1,MSIZE
      DO 503 I=1,MSIZE
  503 FMTX(L,I)=AMTX(L,I)
  101 K=K+1
      KK=K+MON
      AK=KK
      FCTR=FCTR*DT/AK
  102 DO 504 L=1,MSIZE
      DO 504 I=1,MSIZE
      PMTX(L,I)=0.0
      DO 504 J=1,MSIZE
  504 PMTX(L,I)=FMTX(L,J)*AMTX(J,I)+PMTX(L,I)
      DO 505 L=1,MSIZE
      DO 505 I=1,MSIZE
  505 FMTX(L,I)=PMTX(L,I)
  103 DO 507 L=1,MSIZE
      DO 507 I=1,MSIZE
      PMTX(L,I)=PMTX(L,I)*FCTR
  507 TMTX(L,I)=TMTX(L,I)+PMTX(L,I)
      IF(K.EQ.LMT)GO TO 104
      GO TO 101
  104 RETURN
      END
$ENTRY
1.0  0.0  0.0  1.0
0.0  1.0
$IBSYS
```

CD TOT   0165

APPENDIX VII

```
$JOB              003512 LOUIS LEE         100    010    030
$IBJOB            NODECK
$IBFTC LEUNG
C CALCULATION OF THE OPTIMAL PERFORMANCE CRITERION S*
C OF A SECOND ORDER SYSTEM
      DIMENSION X(2,1),XX(2,1),D(1,2),E(2,1),F(2,2)
      ICC = 1
  1   READ(5,10) ((F(I,J),J=1,2),I=1,2)
  10  FORMAT(4F10.5)
      READ (5,11) (E(I,1),I=1,2)
      READ(5,11) (D(1,J),J=1,2)
  11  FORMAT (2F15.5)
      X(1,1)=1.0
      X(2,1)=0.0
      SUM=0.0
      IC=1
  100 U=-(D(1,1)*X(1,1) + D(1,2)*X(2,1))
      SUM=SUM+U**2 +X(1,1)**2+X(2,1)**2
      XX(1,1)=F(1,1)*X(1,1)+F(1,2)*X(2,1)+E(1,1)*U
      XX(2,1)=F(2,1)*X(1,1) +F(2,2)*X(2,1) +E(2,1)*U
      X(1,1)=XX(1,1)
      X(2,1)=XX(2,1)
      IC=IC+1
      IF(IC.LE.600)GO TO100
      SUM=SUM/50.0
      WRITE(6,4) SUM
  4   FORMAT(1H ,10X,F15.5)
      ICC=ICC+1
      IF(ICC.LE.11) GO TO 1
      STOP
      END
$ENTRY
   1.00000    0.02000   -0.00000    1.00000
        0.00020         0.02000
        0.98283         1.71219
   1.00000    0.01996   -0.00000    0.99601
        0.00020         0.01996
        0.98468         1.52675
   1.00000    0.01992   -0.00000    0.99203
        0.00020         0.01992
        0.98632         1.36322
   1.00000    0.01988   -0.00000    0.98807
        0.00020         0.01988
        0.98774         1.22048
   1.00000    0.01984   -0.00000    0.98413
        0.00020         0.01984
        0.98898         1.09678
```

1.00000	0.01980	-0.00000	0.98020
	0.00020	0.01980	
	0.99005	0.99003	
1.00000	0.01976	-0.00000	0.97629
	0.00020	0.01976	
	0.99097	0.89804	
1.00000	0.01972	-0.00000	0.97239
	0.00020	0.01972	
	0.99176	0.81870	
1.00000	0.01968	-0.00000	0.96851
	0.00020	0.01968	
	0.99244	0.75011	
1.00000	0.01964	-0.00000	0.96464
	0.00020	0.01964	
	0.99302	0.69057	
1.00000	0.01961	-0.00000	0.96079
	0.00020	0.01961	
	0.99351	0.63865	

$IBSYS

CD TOT 0080

```
$JOB              003512 LOUIS LEE         100    010    030
$IBJOB            NODECK
$IBFTC LEUNG
C CALCULATION OF THE ACTUAL PERFORMANCE CRITERION S
C OF A SECOND ORDER SYSTEM
      DIMENSION X(2,1),XX(2,1),D(1,2),E(2,1),F(2,2)
      ICC = 1
      READ(5,11) (D(1,J),J=1,2)
  1   READ(5,10) ((F(I,J),J=1,2),I=1,2)
      READ (5,11) (E(I,1),I=1,2)
  10  FORMAT(4F10.5)
  11  FORMAT (2F15.5)
      SUM=0.0
      X(1,1)=1.0
      X(2,1)=0.0
      IC=1
  100 U=-(D(1,1)*X(1,1) + D(1,2)*X(2,1))
      SUM=SUM+U**2 +X(1,1)**2+X(2,1)**2
      XX(1,1)=F(1,1)*X(1,1)+F(1,2)*X(2,1)+E(1,1)*U
      XX(2,1)=F(2,1)*X(1,1) +F(2,2)*X(2,1) +E(2,1)*U
      X(1,1)=XX(1,1)
      X(2,1)=XX(2,1)
      IC=IC+1
      IF(IC.LE.600)GO TO100
      SUM=SUM/50.0
      WRITE(6,4) SUM
  4   FORMAT(1H ,10X,F15.5)
      ICC=ICC+1
      IF(ICC.LE.11) GO TO 1
      STOP
      END
$ENTRY
        0.99005           0.99003
   1.00000    0.02000   -0.00000    1.00000
        0.00020           0.02000
   1.00000    0.01996   -0.00000    0.99601
        0.00020           0.01996
   1.00000    0.01992   -0.00000    0.99203
        0.00020           0.01992
   1.00000    0.01988   -0.00000    0.98807
        0.00020           0.01988
   1.00000    0.01984   -0.00000    0.98413
        0.00020           0.01984
   1.00000    0.01980   -0.00000    0.98020
        0.00020           0.01980
   1.00000    0.01976   -0.00000    0.97629
        0.00020           0.01976
```

```
1.00000   0.01972  -0.00000   0.97239
     0.00020        0.01972
1.00000   0.01968  -0.00000   0.96851
     0.00020        0.01968
1.00000   0.01964  -0.00000   0.96464
     0.00020        0.01964
1.00000   0.01961  -0.00000   0.96079
     0.00020        0.01961
$IBSYS

                              CD TOT   0068
```