DECOMPOSITION AND OPTIMAL CONTROL THEORY

by

MART MASAK

B.A.Sc., University of Toronto, 1963

M.A.Sc., University of Toronto, 1964

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in the Department of

Electrical Engineering

We accept this thesis as conforming to the required standard

Research Supervisor .....

Members of Committee .....

.....

Head of Department .....

Members of the Department

of Electrical Engineering

THE UNIVERSITY OF BRITISH COLUMBIA

July, 1968

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and Study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver 8, Canada

Date July 3, 1968

ABSTRACT

The objective of this thesis is to investigate decomposition and its applicability to the theory of optimal control. The work begins with a representation of the structure of the optimal control problem in terms of directed graphs. This representation exposes a strong connectedness property leading to fundamental difficulties which are central in limiting the class of control problems to which decomposition can successfully be applied.

Computational problems of optimal control are then considered, and decomposition is found to provide a framework within which to analyse numerical methods suitable for parallel processing. A number of such methods are shown and a numerical example is used to illustrate the viability of one of these.

In the second part of the thesis, the optimal control law synthesis problem is discussed together with an inverse problem. The latter concerns the requirement of a second-level co-ordinator in a hierarchical structure. A multi-level controller is then suggested for a class of systems. The effect of this controller structure is to provide a performance very close to the optimal while maintaining adequate sub-optimal control in case of a breakdown of the second-level co-ordinator. The structure

is justified on the basis of the second variation theory of the calculus of variations.

Finally, a new computational technique founded on the geometrical concepts of optimal control theory is introduced. This results in replacing the unstable co-state variables associated with Pontryagin's maximum principle with a set of bounded variables. The facility in the choice of initial iterates makes the method promising.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

ACKNOWLEDGEMENT

It is my pleasure to acknowledge the contributions of a number of people towards this research. In particular, I wish to express my gratitude to Professor E.V.Bohn for his assistance and continued encouragement, and to Dr. A.G. Longmuir for stimulating discussions and criticisms. The original motivation for this thesis is to a large extent due to Professors Ed.Gerecke and M.Mesarovic at the Swiss Federal Institute of Technology.

Special thanks are given to my wife, Katre-Ann, without whose confidence the preparation for this thesis would have been considerably more difficult.

This research was supported by a National Research Council Studentship, 1965-1968, for which I am deeply indebted.

1. INTRODUCTION

The objective of this thesis is to study the applicability of the decomposition concept to optimal control problems. The requirement for decomposition arises from the necessity to understand and control increasingly complex systems both in industry and society in general. These problems are so large that they cannot be handled by the classical scalar input-output analysis; neither do they belong in the category of "disorganized complexity" as do those studied in thermodynamics, for which statistical methods are adequate. This middle ground of so-called organized complexity requires the development of new and possibly special techniques for exploiting the system structure. These really large problems, because they do contain subsystems, are bound to have a structure that can be utilized to advantage.

Decomposing, or breaking down a problem into its constituent parts, has long been recognized as a natural approach to complexity, and numerous works, [1] to [6] , from many research areas have been devoted to different aspects of decomposition, with varying degrees of success. Although a few works such as [5] and [6], and others to be discussed subsequently, have attempted to apply this concept to the dynamic optimization problem, no unqualified breakthroughs

can be claimed.

This thesis begins by defining the structure of an optimization problem and representing this structure by means of directed graphs. Although no significant quantitative conclusions arise from this approach, it does offer a unified framework within which to consider decomposition problems. The thesis then branches into two related studies, one concerned with the computational problem of optimal control, and the other involving the optimal feedback control law synthesis problem. It is argued that the former benefits from decomposition because new, attractive algorithms, suitable for multi-processing computers, arise as practical possibilities. The synthesis problem benefits, since decomposition is the natural first step in obtaining a hierarchical controller structure, and this structure is desirable from many engineering considerations.

John Von Neumann has been quoted as saying that [7] "fundamental improvements in control are really improvements in communicating information within an organization or mechanism." When confronted with the problem of controlling a complex system, a central integrated controller can very well become a bottle neck for the information flow. Decomposition, because it offers the possibility of decentralized control, has the potential of overcoming this bot-

tle- neck, thereby improving the information flow significantly.

2. BACKGROUND TO DECOMPOSITION AND OPTIMIZATION PROBLEMS

Introduction

This part of the thesis provides some definitions as well as basic background material to be used in the subsequent sections. Terms not defined here, such as adjoint variables , strong connectedness and so forth,are assumed to have well defined meanings in the current literature. The Maximum Principle of Pontryagin is stated, but proofs are referenced, and the assumption is made that the reader has familiarity with the current state of optimal control theory. The problem of decomposition and system structure is most readily studied with the aid of the theory of directed graphs, and one section is therefore devoted to the relevance of this theory to the structure of optimal control problems. Finally, this theory is used to limit the class of problems for which decomposition is readily applicable.

2.1 Systems Theory

The theory of systems is a mathematical theory, dealing with abstract entities or mathematical models. In this context, the term *object* , or synonomously *subsystem* is taken to mean a finite set of variables together with a set of relations between them. The term *system* refers to a collection of objects united by some form of mathematical-

ly well-defined interaction or interdependence. By applying cause-effect relationships, the set of variables characterizing an object i may be partitioned into two non-intersecting sets, v_i, known as the subsystem input, and y_i, the system output. In this thesis, it is assumed that the set of relations between v_i and y_i, the input-output description, can be expressed in the form of an ordinary, first-order, vector differential equation,

$$\dot{y}_i = f_i(y_i, v_i) \qquad 2.1$$

and at some time t_o, it is assumed that y_i is available for measurement, so that

$$y_i(0) = y_{o_i} \qquad 2.2$$

Furthermore, it is assumed that the interactions which unite the collection of objects to form the system be expressible by a vector algebraic equation of the form

$$g(y,v) = 0 \qquad 2.3$$

wher y and v are the composite vectors formed from the vectors y_i and v_i as i ranges over all the objects comprising the system.

Equations 2.1, 2.2, and 2.3 are sufficient to provide a complete mathematical description of the system dynamics. In fact, some redundancy is bound to exist, since many subsystems have outputs which form part of the input of some other subsystem. In this thesis as well as in almost all control system literature, these redundancies are eliminated

by explicitly solving for some of the variables in equation 2.3 . The result of this is a new smaller set of input-output variables, called x and u, which are interconnected by the dynamical equation

$$\dot{x} = f(x,u) \qquad 2.4$$

while equation 2.3 has been eliminated completely.

It is now assumed that the remaining system inputs, u, are all control inputs in the sense that, within certain practical limitations, these vectors may be chosen at will by the system designer. This precludes the possibility of noise inputs, but this is justified by the following argument. In the case of the computational algorithm, the objective is to calculate the ideal behaviour of the system under no imperfections such as noise or controller limitations. This ideal is then used either directly in controller synthesis, or indirectly, as a measure of the performance of some realistic, implementable control law. In the case of the synthesis procedure for on-line controllers, a control law is obtained whose sole purpose is to correct for noise inputs to the system. Unless these uncontrolled disturbances are highly predictable, there is no effective way to compensate for them in the generation of the feedback control law, and they are therefore ignored.

It is further noted that the foregoing discussion

makes no mention of the concept of the state of the system. In the first place, no purpose is served by entering into a long discussion of this concept here, when so many excellent references such as [8] and [9] treat this in detail. More significantly, the assumption is made that there exists a unique minimal state description of the system, which is identical to the afore-mentioned input-output description of the system. The reason for this somewhat restrictive assumption is that with decomposition, the emphasis is on the system structural properties as manifested at the input-output level. As with stability properties, these structural properties are not preserved with arbitrary transformations of state, and at present, no restrictions on these transformations are available which do imply these preservations. Henceforth in this thesis, the terms output and state shall be employed interchangeably, keeping in mind the above assumption.

2.2 The Optimal Dynamical System

The optimization problem to be studied in this thesis is now defined. Given a dynamical system whose motion is described by the equation

$$\dot{x} = f(x,u), \qquad 2.4$$

$$x(0) = x_o$$

determine the vector $u \in \Omega$ which will minimize the functional

$$J = \int_o^{t_f} h(x,u)\, dt \qquad 2.5$$

subject to

$$M(x(t_f)) = 0 \qquad 2.6$$

where Ω is some convex set,

$M(.)$ is a vector valued terminal condition,

$h(.,.)$ is a scalar valued, positive semi-definite function.

The Hamiltonian for the problem is defined as

$$H(x,p,u) = -h(x,u) + p'f(x,u) \qquad 2.7$$

The Maximum Principle of Pontryagin states that if the control $u= u^*$ is optimal, then corresponding to u^* and the generated optimal trajectory x^*, there exists an adjoint vector p^*, such that

$$\dot{x}^* = \nabla_p H(x^*,p^*,u^*) \qquad 2.8$$

$$\dot{p}^* = -\nabla_x H(x^*,p^*,u^*) \qquad 2.9$$

$$p^*(t_f)' M_x(x^*(t_f)) = 0 \qquad 2.10$$

$$H(x^*,p^*,u^*) \geqslant H(x^*,p^*,u)$$

for all $u \in \Omega$, and all t, $0 \leqslant t \leqslant t_f$. 2.11

This result is proved in detail in [10], while [11] and [12] serve as useful references and give examples of how this is used. Equations 2.8 to 2.11 define a two-point boundary value problem, henceforth referred to as the TPBVP. Equation 2.11 provides an algebraic relationship between the variables x, u and p, and with the use of the implicit function theorem, if the problem is non-singular [13], u can be expressed as a function of x and p. Therefore, these equations are sufficient for generating the optimal trajectory $x(t)$, and the optimal control $u(t)$, and these will presumably be unique if the problem is well defined.

The difficulty arises because the mixed boundary conditions do not allow readily obtainable solutions. To quote Letov [14] : "... the famous two-point boundary value problem stands as a fortress that has been attacked time after time, but never conquered. ... (I am) not so bold as to claim that success is close at hand in many applications of variational techniques which are both rigorous and legitimate..."

This obstacle has sent many to search for other optimization techniques, the most notable of which has been dynamic programming. However, none has been found that is really as useful in general as the maximum principle, and, therefore, this thesis will consider optimal control problems only from this viewpoint.

Consequently, in the subsequent sections, the dynamical system defined by the necessary conditions of the maximum principle will be studied extensively, and this system, governed by equations 2.8 to 2.11 will be referred to as the Optimal Dynamical System, or the ODS for short. By the term generating system for the ODS is meant the original dynamical system defined by equation 2.4, where no stipulations are made regarding performance functionals. As will become evident shortly, it is important to differentiate between these two systems.

The structure of the system is now defined as the network of couplings between the objects which constitute the system. Clearly, the structure of the ODS is potentially

much more complex than the structure of the generating system. Therefore a successful decomposition technique must utilize all the information not only of the generating system structure, but also that of the ODS, in which the generating system is embedded. The logical tool for studying system structure is the branch of topology known as the theory of directed graphs [15] . The objective of applying this theory to the study of the ODS is to justify the use of a limited class of systems in the subsequent sections dealing with decomposition.

2.3 Digraph Theory and the Optimal Dynamical System

The subsequent discussion of the theory of directed graphs will employ terminology identical to that of reference [15], Harary et al.A directed graph, henceforth referred to as a digraph, consists of two sets,the set of points and the set of lines, each line having associated with it a direction.

To obtain the digraph of an ODS, all the variables x , p , and u * are considered as points. The lines for the digraph are obtained as follows. Since each of the components of x and p is generated by a differential equation, the lines converging to a point will be defined by the variables on the right hand side of the respective differential equation. For example, if the

* In the next section, some others are also included.

differential equation generating p_3 is given by

$$\dot{p}_3 = f(x_2, x_5, p_1)$$

then to the node associated with p_3 will come directed lines from the nodes associated with x_2, x_5, and p_1. If $f(.)$ were to include p_3, logically a self-loop would be required, but since these do not further the understanding of the relationships between the variables, they will be omitted.

For each component of u, there is a relationship requiring that H be a maximum, and consequently, the explicit solution of this relationship for the particular u component provides an algebraic equation which is used analogously to obtain the lines of the graph.

The digraph for the entire ODS then consists of the totality of such points and lines. It is noted that the digraph thus obtained is in many ways similar to a standard block diagram. However, block diagrams already have associated with them concepts of transfer functions and linearity, while the digraph is much more general. The generation of an ODS digraph is now illustrated with some examples.

Example 2.1

Find the functions u and v which will minimize the functional

$$J = \tfrac{1}{2} \int_0^T (x^2 + y^2 + u^2 + v^2) \, dt$$

subject to $\dot{x} = -x + 10xy + u \qquad x(0) = x_o$

$\dot{y} = x + v \qquad y(0) = y_o$

$|u| \leq U \qquad |v| \leq V$

The Hamiltonian is

$$H = -\tfrac{1}{2}(x^2 + y^2 + u^2 + v^2) + p(-x + 10xy + u) + q(x + v)$$

Thus, the ODS will be of the form

$\dot{x} = -x + 10xy + u \qquad x(0) = x_o$

$\dot{y} = x + v \qquad y(0) = y_o$

$u = U\ \mathrm{sat}(p/U)$

$v = V\ \mathrm{sat}(q/V)$

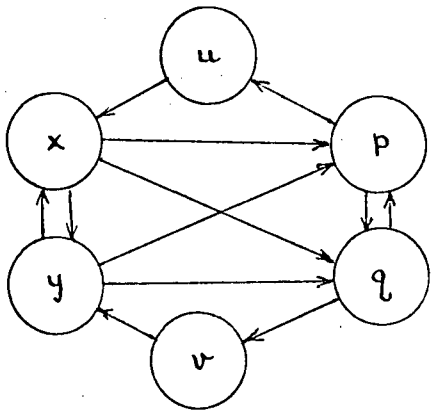
$\dot{p} = x + p - 10yp - q \qquad p(T) = 0$

$\dot{q} = y - 10xp \qquad q(T) = 0$

where

$$\mathrm{sat}(z) = \begin{cases} 1 & z > 1 \\ z & -1 \leq z \leq 1 \\ -1 & z < -1 \end{cases}$$

and the digraph will be as follows:



It is observed that this digraph is strongly connected.

Example 2.2

Choose u and v to minimize the functional

$$J = \tfrac{1}{2} \int_0^T (y^2 + u^2 + v^2) \, dt$$

subject to

$$\dot{x} = -x^3 + u \qquad x(0)=x_o \qquad x(T) = 0$$

$$\dot{y} = -y + v \qquad y(0)=y_o \qquad y(T) = 0$$

Thus $$H = -\tfrac{1}{2}(y^2 + u^2 + v^2) + p(-x^3 + u) + q(-y + v)$$

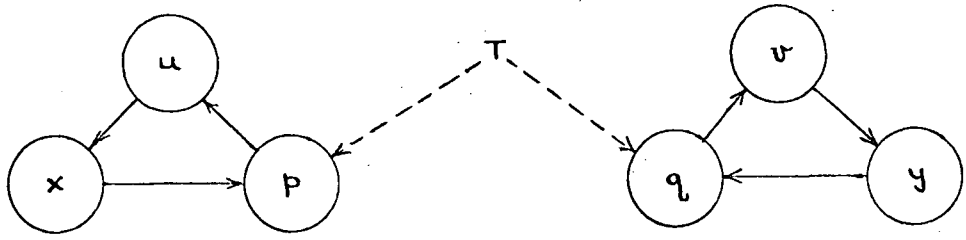
$$u = p$$

$$v = q$$

$$\dot{p} = 3x^2 p$$

$$\dot{q} = y + q$$

The digraph for this system is



If the digraph for this ODS is drawn according to the preceding rules, it is found to be totally disconnected, and this of course implies that two independent optimization problems are being considered. However, these problems are not entirely independent, since they are connected through the parameter T and the requirement that both subsystems reach the origin at the same time. This interconnection is indicated in the above graph by means of the dotted lines.

Thus, the overall graph remains connected, but not strongly connected as in the previous example.

Example 2.3

Again, find u and v to minimize

$$J = \tfrac{1}{2}\int_0^T (x^2 + y^2 + z^2 + u^2 + v^2)dt$$

subject to

$$\dot{x} = -xz + u$$

$$\dot{y} = -yz + v$$

$$\dot{z} = -z$$

Application of the maximum principle yields

$$u = p$$

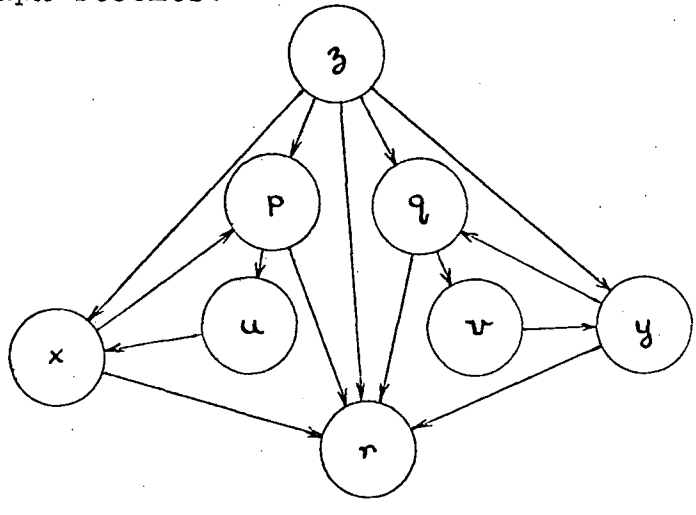
$$v = q$$

$$\dot{p} = x + pz$$

$$\dot{q} = y + qz$$

$$\dot{r} = px + qy + r + z$$

and the digraph becomes:



The basic digraph for this ODS has a source and a sink,

and is, therefore , again not strongly connected. A look at the generating system reveals it to be uncontrollable. In fact, it is easy to show that a necessary condition for the controllability of the generating system is that the basic digraph of the ODS, composed of x , p and u , have no sources or sinks. It is further noted , however, that the optimization problem for this example is well-defined and has a unique solution.

The foregoing examples illustrate how the structure of an ODS can be represented in terms of a digraph. It is evident that some transformations of state variables can profoundly affect the internal structure and hence the digraph of the system. However, because these transformations leave the input-output description of the system invariant, and because the synthesis is concerned mainly with this description and its associated structure, the original restriction to this unique description of the system is justified.

The basic shortcoming of digraphs is the lack of capacity or values on the lines, and this prohibits quantitative statements regarding the system structure. This shortcoming can not be overcome in the case of systems with nonlinearities and therefore, only qualitative conclusions can be drawn. The next section describes how these qualitative conclusions are employed to effect ODS decomposition.

2.4 Decomposition

The Oxford dictionary defines decomposition as"breaking down into its constituent parts." The objective of this section is to identify constituent parts of an ODS, and then, with the aid of digraph theory, attempt conceptually to break the optimization problem into smaller sub-problems. As will become evident shortly, the natural constituent parts of an ODS do not necessarily coincide with the subsystems which make up the generating system. In order to illustrate decomposition, consider the following example.

Example 2.4

Find u and v which will minimize the functional

$$J = \tfrac{1}{2} \int_0^T (x^2 + y^2 + 2\varepsilon xy + u^2 + v^2)\,dt$$

subject to

$$\dot{x} = -5x + \varepsilon y + u \qquad x(0) = x_o$$

$$\dot{y} = -7y + y^3 - 3\varepsilon x + v \qquad y(0) = y_o$$

The maximum principle yields

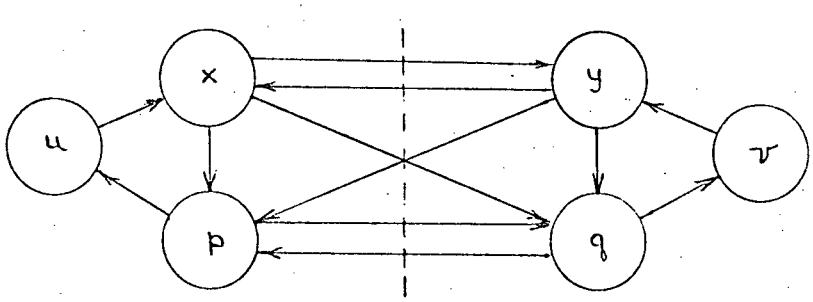
$$u = p \qquad v = q$$

where

$$\dot{p} = x + \varepsilon y + 5p + 3\varepsilon q \qquad p(T) = 0$$

$$\dot{q} = y + \varepsilon x - \varepsilon p + 7q - 3y^2 q \qquad q(T) = 0$$

The digraph for this ODS is



It is noted that although the generator systems for examples 2.1 and 2.4 are different, the ODS's have identical structures.

It is evident that the digraph has a high degree of connectivity, and because the lines have no capacities associated with them, the job of isolating constituent parts on the digraph is difficult. However, if it is known that ε represents a small quantity, and since all the lines crossing the dotted line are weighted by ε , then it would appear that the obvious choice for the ODS constituent parts would consist of (x,u,p) and (y,v,q) .Both of these constituent parts are strongly connected, and with $\varepsilon = 0$, define meaningful optimization problems. This latter underlines the fact that this writer has failed to find any problem in which it is useful to consider an output variable and its associated adjoint variable in different constituent parts. This tends to corroborate the hypothesis that if an optimization problem lends itself to successful decomposition, it must be built up from a number of interconnected optimization problems, each of which can be made meaningful when considered alone.

Now, of course, the concepts of decomposition can be applied just as well to the generating system as to the ODS, and in fact, this is done in papers dealing with multi-level systems, aggregated systems and so forth. There

is no harm in this, provided one is not concerned with a specific optimization problem. The following somewhat simple-minded example underlines the differences between decomposing the generating system and the ODS .

Example 2.5

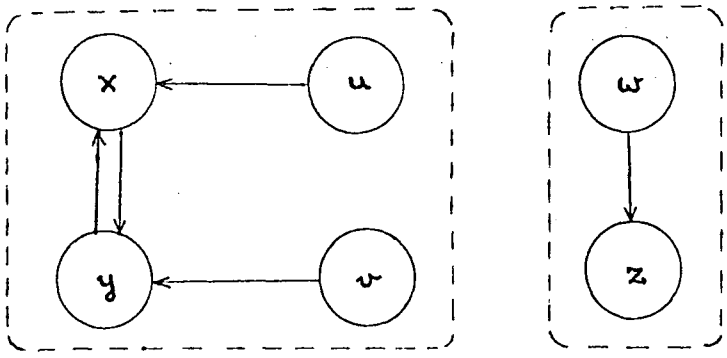
Consider a generating system with the dynamics

$$\dot{x} = f(x) + \varepsilon y + u$$

$$\dot{y} = g(y) + \varepsilon x + v$$

$$\dot{z} = -z \qquad + w$$

which gives as a digraph



If the objective of decomposition were to choose two subsystems which had the least amount of interaction between them, then the obvious choice, considering only the generating system, would be as indicated by the dotted lines above.

However, if the problem were to find, as well, the functions u, v and w such that, subject to the above dynamical constraints, the following functional was minimized :

$$J = \tfrac{1}{2} \int_0^T \{\eta(y-z)^2 + x^2 + u^2 + v^2 + w^2\}\, dt$$

then

$$H = -\tfrac{1}{2} \{\eta(y-z)^2 + x^2 + u^2 + v^2 + w^2\} + p(f(x) + \varepsilon y + u) + q(g(y) + \varepsilon x + v) + r(-z + w)$$

and

$$u = p \qquad v = q \qquad w = r$$

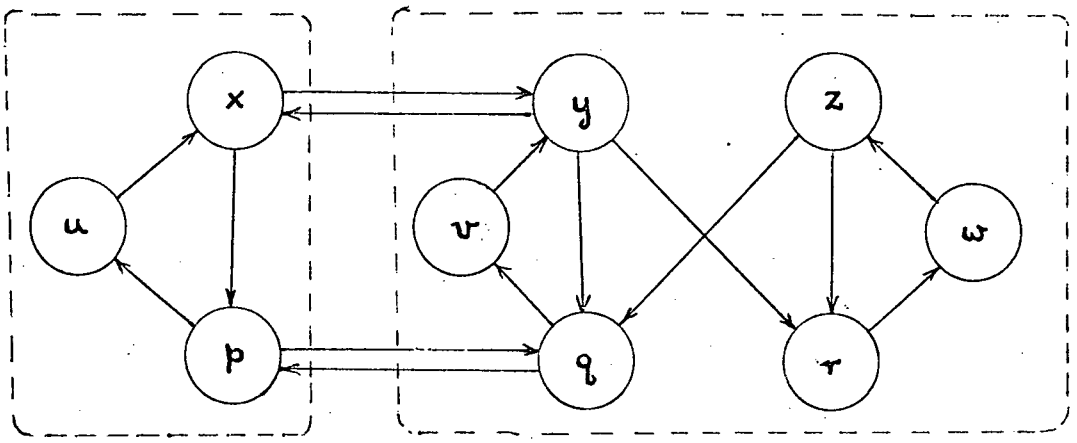
where

$$\dot{p} = x - \frac{\partial f}{\partial x} p - \varepsilon q$$

$$\dot{q} = \eta(y-z) - \varepsilon p - \frac{\partial g}{\partial y} q$$

$$\dot{r} = -\eta(y-z) + r$$

and the ODS digraph would be as follows:



If the original objective in decomposition is retained and if ε is smaller than η ,the natural decomposition appears as shown.

In striving for decomposition, it is evident that the objectives of decomposition differ somewhat between the computational problem, and the on-line synthesis problem. However in both problems the first step is to group sets of input

and output variables to be considered as sub-systems. Because relatively few results are available for the synthesis problem in general, the usefulness of the digraph to this problem is limited, except perhaps in the negative sense, that if the digraph is strongly connected to a high degree of connectedness, and if it appears that the connections have strong influences, then it may be wiser to sythesize an integrated, centralized controller. Also, for this reason, the hierarchical synthesis problem in part 4 is restricted to a special class of systems, whose digraph is rather trivial.

The objective of decomposition with computational problems will be discussed further in part 3 , but here it suffices to say that one of the goals is multi-processing. However, because the digraph of an ODS is in general strongly connected, the standard TPBVP is incompatible with multi-processing. The ideal structure for this is a number of parallel paths, but since that is an unrealistic goal, then weak connectedness is acceptable. In fact, the next part of the thesis will cover a number of techniques which reduce to adding more nodes to the ODS digraph and then restructuring it from the undesirable strongly connected configuration to a more desirable , weakly connected, sink-source configuration.

3 . DECOMPOSITION AND COMPUTATIONAL ALGORITHMS

Introduction

A major segment of optimal control theory concerns the numerical computation of the function representing the ODS trajectory, and this function is required to a high degree of accuracy in off-line optimal guidance problems. This emphasis on accuracy may be contrasted with the on-line control problems, treated in part 4 , wherein the paramount consideration is the computation time. Nevertheless, in guidance problems of any complexity, the computation time acquires the second most important position, since it ultimately determines the practical feasibility of a particular solution. Therefore, in this section, decomposition is considered not only as an attractive possibility for simplifying the numerical procedures, but as a means of achieving greater computational speed.

The objective, then, is to sub-divide the original computational problem into smaller, and hopefully simpler, sub-problems, preferably chosen in such a way that each can be solved relatively independently of others. For example, dynamic programming offers what might be termed temporal decomposition, in that at each stage (in time), the solution of a simple problem is required. Closely connected with dynamic programming is the technique, hopelessly complex for dynamical systems, for optimizing a

system with cascaded structure, as described by Fan et al [16]. Although these achieve the goal of simplifying the numerical procedures, they are impractical because of excessive computer memory requirements. If memory is extended to include tapes, then of course, computation time becomes impractical. Naturally, no decomposition scheme is acceptable if the time required to sequentially solve the set of sub-problems exceeds the solution time of the original integrated problem, assuming that some algorithm exists for doing so.

In fact, the primary reason for considering decomposition is the possibility of discovering situations where the smaller sub-problems can be solved simultaneously, thereby offering a potentially significant decrease in total computation time, if suitable multi-processing computers are available. Consequently, this part of the thesis begins with a brief discussion of the nature of parallelism in numerical algorithms. Then, some algorithms which possess an inherent high degree of parallelism are described and an attempt is made to define the structural properties of the control problems which would allow efficient use of this parallelism.

3.1 Parallelism in Computational Algorithms

When formulating a sequential numerical algorithm, one does not feel obliged to consider many restrictions arising from the fact that a computer will eventually process the algorithm. In the case of parallelism, this situation is reversed, and because such a variety of multi-processing machinery is possible, there is a strong temptation to tailor algorithms to specific computer configurations. This tendency is undesirable if one wishes to achieve any generality with the proposed algorithm.

Consequently, the discussion of proposed and actual multi-processing hardware has been relegated to Appendix B, while some universal aspects of parallelism, independent of hardware, are considered here.

Parallelism can be classified into four levels. The lowest occurs at the bit level, and consists of parallel logic and arithmetic hardware incorporated into most present-day high-speed computers. This level will not be considered any further in this thesis. The next level involves parallelism of individual instructions, and might be considered as the digital counterpart of an analogue computer, in that all instructions which can be performed simultaneously will be. Although this concept may become feasible in the future, present day hardware and

soft-ware limitations preclude this from practical considerations, except in very limited form, such as "n-step look-ahead," incorporated in some large, modern computers.

The third level, which will be emphasized subsequently, is parallelism among sub-routines, or groups of instructions, all associated with a single, overall problem. It is assumed that the programmer inserts FORK and JOIN [17], [18] instructions into the program, thereby establishing the possible extent of the subroutine parallelism. On encountering the FORK instruction, the executive program allocates free processors to the designated parallel streams, but if the number of streams exceeds the number of free processors, some streams will be executed sequentially. It is apparent, therefore, that for an efficient multi-processing computer, the executive program must be highly elaborate. Furthermore, in order to make efficient use of this parallel capability, the subroutines must be chosen so that the executive program time required per sub-routine is small in comparison to the sub-routine running time.

The highest level of parallelism is that between independent programs. Multi-processors utilizing this rather trivial form of parallelism are well on their way to commercial reality, and again, this form will not be discussed any further.

The width of the computation front at any given time, considering parallelism of type 3 , is defined as the number of parallel streams emanating from the last FORK instruction which have not yet been combined by a JOIN. It is evident that in developing algorithms, there is nothing to be gained by making the width of the computation front exceed the total number of processing elements in the computer.

When dealing with optimal control problems, by far the greatest proportion of time is consumed in repeated integrations of differential equations. In fact, since usually $2n$ equations must be integrated, where n is the dimension of the state space of the system, there arises the possibility of using $2n$ (or more) processing elements in parallel to obtain each integration step. However, because of the relatively few operations involved in each step, this approach must be rejected as the executive/sub-routine time ratio probably becomes excessive. Moreover, if n is large, this approach would require a non-conventional computer, as described in Appendix B .

An alternative is to choose groups of equations, and integrate these groups in parallel, using for interactions some appropriate iteration information. Since this approach is iterative, it would appear that if N groups are chosen to be integrated in parallel, significant gains in

computation speed could be achieved only if the required number of iterative integrations were less than N . However, as will be shown, this requirement can be relaxed somewhat, since these integration iterations can be combined effectively with those arising from the TPBVP .

These brief comments on parallelism in algorithms end this discussion, and some specific cases of optimal control problem computation are next considered in order to illustrate the potential of multi-processing.

3.2 Generalized Picard Algorithm

As already mentioned, the well-known classical method of decoupling a set of equations is to employ iteration. Thus, instead of solving the actual equations, one solves a different set, dependent on a previous iteration. In the case of the ODS , a very obvious decoupling and iteration scheme, based on a generalization of the Picard iteration for differential equations immediately arises. This will be illustrated with the following example.

Example 3.1

Assume that the application of the maximum principle to some optimization problem has given rise to the following TPBVP :

$$\dot{x} = f_1(x,y,u) \qquad x(0) = x_o$$

$$\dot{y} = f_2(x,y,v) \qquad y(0) = y_o$$

$$u = u(x,p)$$

$$v = v(y,q)$$

$$\dot{p} = g_1(x,p,q) \qquad p(T) = 0$$

$$\dot{q} = g_2(y,p,q) \qquad q(T) = 0$$

This TPBVP is reformulated as

$$\dot{x} = f_1(x,y^*,u) \qquad x(0) = x_o$$

$$\dot{y} = f_2(x^*,y,v) \qquad y(0) = y_o$$

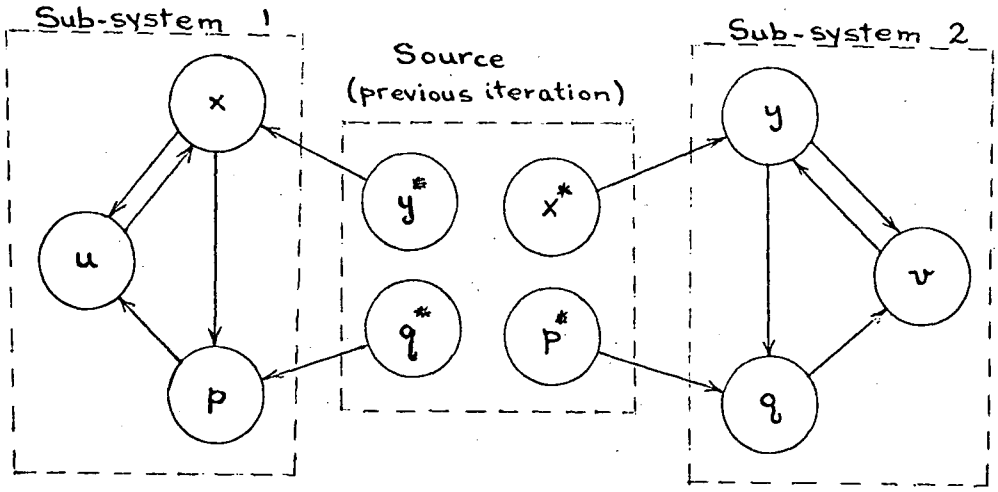
$$u = u(x,p)$$

$$v = v(y,q)$$

$$\dot{p} = g_1(x,p,q^*) \qquad p(T) = 0$$

$$\dot{q} = g_2(y,p^*,q) \qquad q(T) = 0$$

whre x*, y*, p* and q* are functions obtained at a previous iteration. Naturally, it is assumed that some starting iterate can somehow be chosen. The digraph for this second system is therefore a source-sink configuration as illustrated:



This approach provides both a weakly connected ODS structure, and a parallelism of type 3 with a computation front width acceptable to conventional multi-processing computers. Furthermore, in principle, there are no

restrictions on the applicability of the method, and non-linear as well as linear problems can be handled. However, severe difficulties can be encountered in choosing good initial iterates, and consequently, a simple-minded application of the method may lead to unsatisfactory rates of convergence, as reported by Takahara [19]. Another objection against the method is that being a generalized Picard algorithm, it also has a characteristic slow rate of convergence in comparison to Newton-Raphson algorithms. Nevertheless, later in this section, it will be shown that by introducing a number of modifications, the method has important practical significance for some classes of problems. Before this, however, some other approaches, somewhat more elegant theoretically, though of little import as practical computation schemes, will be considered.

3.3 Decomposition Algorithm Based on Duality

Probably the best known and rather aesthetic algorithm concerned with decomposition and optimal control theory is that derived by J.D.Pearson [20] [21] , based on duality and the Legendre transformation of the calculus of variations [22]. Of all the techniques considered in this thesis, this one bears the closest resemblance to the Decomposition Principle for Linear Programs, as discussed in Dantzig and Wolfe [3]. Although this latter principle has enjoyed some degree of success, there does not appear to be any

way of extending it to optimization problems with dynamical constraints, as considered here. The technique of Pearson, though similar, has wider applicability but is also a far weaker numerical algorithm. For the sake of completeness and for purposes of comparing the resultant digraph structure with the other algorithms, Pearson's technique is included here.

For illustrative purposes, consider the optimization problem of finding u and v which minimize

$$J = \tfrac{1}{2} \int_0^T (x^2 + y^2 + u^2 + v^2)dt \qquad 3.1$$

subject to

$$\dot{x} = a_{11}x + a_{12}y + u \qquad x(0) = \alpha \qquad 3.2$$

$$\dot{y} = a_{21}x + a_{22}y + v \qquad y(0) = \beta \qquad 3.3$$

$$x \in Q_1 \qquad y \in Q_2$$

where Q_1 and Q_2 are closed, bounded, convex sets. That is, the problem concerns a linear system with a quadratic performance functional, and, very importantly, bounded state variables.

By introducing

$$y = r \qquad 3.4$$

$$x = s \qquad 3.5$$

the dynamical constraints can be re-written as

$$\dot{x} = a_{11}x + a_{12}r + u \qquad 3.6$$

$$\dot{y} = a_{21}s + a_{22}y + v \qquad 3.7$$

The original problem can then be reformulated as

finding the minimum of J , subject to constraints 3.4 to 3.7 . Forming the Lagrangian,

$$L(x,y,u,v,p_1,p_2,s,r,\lambda,\mu) = \int_0^T \{ \tfrac{1}{2}(x^2 + y^2 + u^2 + v^2) + p_1 (\dot{x} - a_{11}x - a_{12}r - u) + p_2 (\dot{y} - a_{21}s - a_{22}y - v) + \lambda(y - r) + \mu(x - s) \} dt$$

the problem requires the extremization of L subject to $x(0) = \alpha$, and $y(0) = \beta$, $x,s \in Q_1$, $y,r \in Q_2$.

Now L conveniently breaks up into two parts,

$$L = L_1(x,u,p_1,r,\lambda,\mu) + L_2(y,v,p_2,s,\lambda,\mu)$$

where

$$L_1 = \int_0^T \{ \tfrac{1}{2}(x^2 + u^2) + p_1 (x - a_{11}x - a_{12}r - u) + (x\mu - \lambda r) \} d\tau$$

$$L_2 = \int_0^T \{ \tfrac{1}{2}(y^2 + v^2) + p_2 (y - a_{21}s - a_{22}y - v) + (y\lambda - \mu s) \} d\tau$$

Conditions which require L to be at an extremum also require that both L_1 and L_2 be at an extremum. This requirement leads to the conclusion that there exist two sub-problems, similar in form to the original, which, when solved, lead to the solution of the original.

Thus, for a given λ and μ , both L_1 and L_2 can be extremized as

$$L_1^* = L_1^*(\lambda, \mu)$$
$$L_2^* = L_2^*(\lambda, \mu)$$

Then, using the well-established saddle point arguments in Banach space,[23] , it follows that

$$L^* = \max_{\lambda,\mu} \left[L_1^*(\lambda, \mu) + L_2^*(\lambda, \mu) \right]$$

It is noted that in the case of the sub-system extremizations, s and r are treated as sub-system controls, while λ and μ are arbitrary functions. Thus, for example, the first subsystem problem is, given λ and μ, minimize with respect to u and r,

$$J_1 = \tfrac{1}{2} \int_0^T \{(x^2 + u^2) + 2(\mu x - \lambda r)\} dt$$

subject to

$$\dot{x} = a_{11}x + a_{12}r + u \qquad x(0) = \alpha$$

$$x \in Q_1 , \qquad r \in Q_2$$

The ODS generated by this sub-problem is

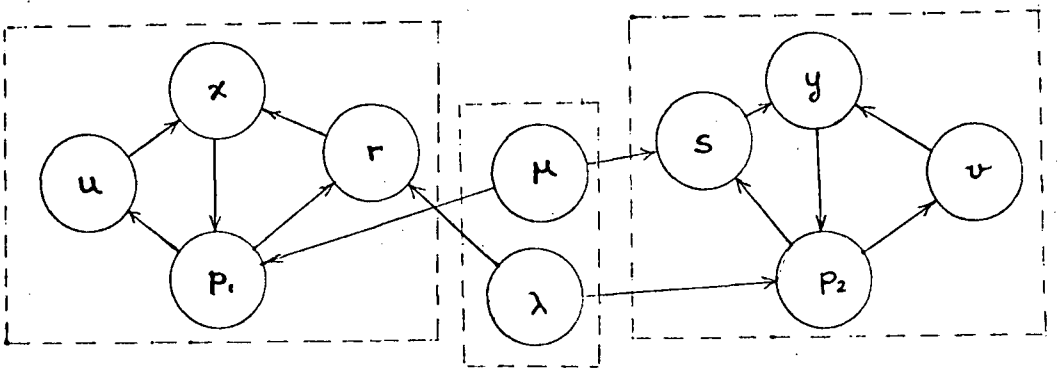
$$\dot{x} = a_{11}x + a_{12}r + u \qquad x(0) = \alpha$$

$$\dot{p}_1 = x - a_{11}p_1 + \mu \qquad p_1(T) = 0$$

$$u = p_1$$

$$r = r(p_1, \lambda)$$

where the equation for r is obtained by maximizatition of the sub-system Hamiltonian with respect to r, subject to the condition that r be in Q_2. The digraph for this entire system again has the desirable property, namely two strongly connected subgraphs, connected together weakly, as shown in the following diagram:



Therefore, ideally, the extrema of the two sub-systems are obtained as functions of λ and μ, and the sum then maximized. However, since λ and μ are elements of a function space, such explicit functional representations are impossible to obtain in practice. Realizing this, Pearson has suggested a hill-climbing scheme, where one begins with some arbitrary λ and μ, obtains L_1^* and L_2^*, and proceeds hill-climbing in the conventional manner. While this suggestion is valid, it is very questionable whether the method will ever be attractive computationally, since each step in the hill climb requires the solution of variational problems, and hill-climbing techniques are not noted for their fast convergence.

The other major objection to this theory is that it is valid only for linear systems with quadratic performance criteria, since the duality theorem has only been established for the case of bilinear functionals on a Banach space, subject to linear constraints. Furthermore, as this problem has the very attractive solution formulated by Kalman, the practical usefulness of this dual theory is questionable. It is not known whether the theory can be extended to more general problems.

Also, before leaving this technique, mention must be made of the bounded state variable requirement. In fact, this requirement is not necessary, and when absent, some form of juggling, as in reference [24], can sometimes solve

the problem. However, it is noted that, in contrast to other methods, the boundedness requirement is not merely an extension of the applicability of the theory, but rather a constraint, which can sometimes, but not always be circumvented.

3.4 Decomposition Using Penalty Functions

A decomposition scheme, based on penalty functions, has been suggested as another possibility. The method is illustrated for the same problem as in the previous sub-section, although this does not imply a restriction to linear systems with quadratic performance criteria. The problem is to minimize 3.1 subject to 3.2 and 3.3 . Again, 3.4 and 3.5 are introduced, and the dynamical constraints are rewritten as 3.6 and 3.7 .

Now the original performance functional is augmented using penalty functions in the following manner:

$$J_N = J + \tfrac{1}{2}\int_0^T \left[\mu_1(x-s^*)^2 + \mu_2(y-r^*)^2 + \mu_3(r-y^*)^2 + \mu_4(s-x^*)^2\right] dt \qquad 3.8$$

where the variables with the asterisks will be considered as the respective previous iterates. Thus, if the iterative scheme were to converge, the penalty functions would approach zero, and $J_N \longrightarrow J$. It is observed that if constraints 3.4 and 3.5 are neglected, and if fixed values are chosen for the vector μ , then minimization

of 3.8 with respect to u, v, r and s subject to constraints 3.6 and 3.7 consists of two independent problems. The ODS shall be demonstrated only for one.

That is , find u and r which minimize

$$J_1 = \tfrac{1}{2}\int_0^T [x^2 + u^2 + \mu_1(x-s^*)^2 + \mu_3(r-y^*)^2]\, dt$$

subject to

$$\dot{x} = a_{11}x + a_{12}r + u \qquad x(0) = \alpha$$

It is noted that this method allows freedom in the choice of particular penalty functions so as to impose functional convexity in both u and r , as opposed to the approach by Pearson, where, as mentioned, difficulties can arise if the problem is not originally formulated with bounded state variables.

Now, the ODS for this sub-problem is:

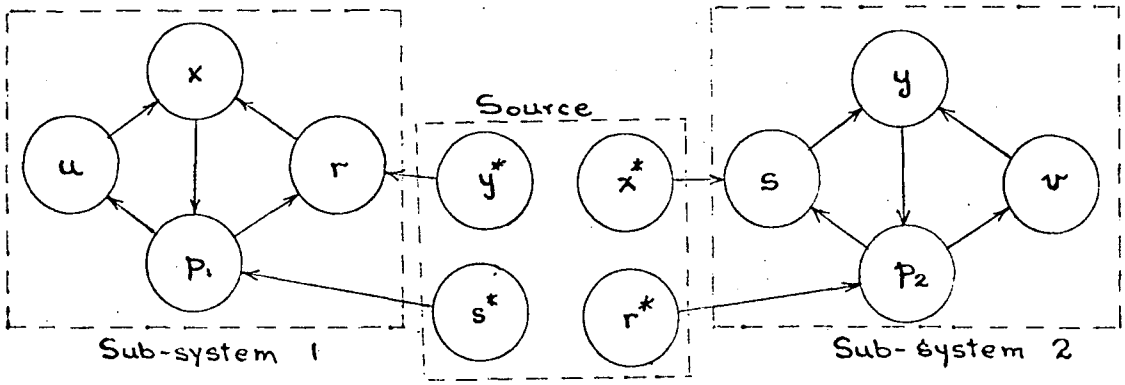
$$\dot{x} = a_{11}x + a_{12}r + u \qquad x(0) = \alpha$$

$$p_1 = x + \mu_1(x-s^*) - a_{11}p_1 \qquad p_1(T) = 0$$

$$u = p_1$$

$$r = y^* + \frac{1}{\mu_3}(a_{12}p_1)$$

The overall system digraph is:



It is evident that this structure is a combination of the two previous structures, and it is interesting to compare it to the others. To begin with, it appears that in contrast to the generalized Picard method, this technique does not require an initial p - vector to serve as a source. In addition, since after convergence, r and s should approach y and x respectively, it might be somewhat easier to choose a satisfactory initial iterate.

Consider now the role of the μ- vector. From the calculus of variations, it is known that the solution to the original problem requires infinite values for μ. Also, substitutions show that in fact the four components can not be independent, if the solution to the original problem is to be attained, but rather the following relationships should hold:

$$\mu_1 = \mu_4$$

$$\mu_2 = \mu_3$$

These two considerations indicate that perhaps by satisfying the above relationships at some suitably large values of μ_1 and μ_2, a solution arbitrarily near the optimum could be obtained. Since μ represents a measure of the effective coupling between the sub-systems, it may be expected that for small values of μ, convergence can be obtained rather quickly, and then with these values

for x and y as initial iterates, larger values of μ can be progressively considered until a practical infinity is obtained. Although some success has in fact been obtained with a slight modification of this approach, the following argument should demonstrate the weaknesses of of the method in the general case.

Although in the numerical computation, the equations representing the ODS as already shown, would be used, the following equations illustrate from an alternative viewpoint what is actually being computed. After some substitution, it is found that the TPBVP to be solved is

$$\dot{x} = a_{11}x + a_{12}(y^* + \frac{1}{\mu_3}a_{12}p_1) + p_1 \qquad x(0)=\alpha$$

$$\dot{p}_1 = x + \mu_1(x-s^*) - a_{11}p_1 \qquad p_1(T)=0$$

$$\dot{y} = a_{21}(x^* + \frac{1}{\mu_4}a_{21}p_2) + a_{22}y + p_2 \qquad y(0)=\beta$$

$$\dot{p}_2 = y + \mu_2(y-r^*) - a_{22}p_2 \qquad p_2(T)=0$$

But since r^* and s^* are obtained from previous iterates, which are denoted by double superscripts, the TPBVP can be re-written as

$$\dot{x} = a_{11}x + a_{12}(y^* + \frac{1}{\mu_3}a_{12}p_1) + p_1 \qquad x(0)=\alpha$$

$$\dot{p}_1 = x + \mu_1(x - [x^{**} + \frac{1}{\mu_4}a_{21}p_2^*]) - a_{11}p_1 \qquad p_1(T)=0$$

$$\dot{y} = a_{21}(x^* + \frac{1}{\mu_4}a_{21}p_2) + a_{22}y + p_2 \qquad y(0)=\beta$$

$$\dot{p}_2 = y + \mu_2(y - [y^{**} + \frac{1}{\mu_3}a_{12}p_1^*]) - a_{22}p_2 \qquad p_2(T)=0$$

Using $\mu_1 = \mu_4$, $\mu_2 = \mu_3$, these equations reduce to :

$$\dot{x} = a_{11}x + a_{12}y^* + p_1 + \frac{1}{\mu_3} a_{12}^2 p_1 \qquad x(0) = \alpha$$

$$\dot{p}_1 = x - a_{11}p_1 - a_{21}p_2^* + \mu_1(x-x^{**}) \qquad p_1(T) = 0$$

$$\dot{y} = a_{21}x^* + a_{22}y + p_2 + \frac{1}{\mu_4} a_{21}^2 p_2 \qquad y(0) = \beta$$

$$\dot{p}_2 = y - a_{12}p_1^* - a_{22}p_2 + \mu_2(y-y^{**}) \qquad p_2(T) = 0$$

These equations, being an alternate representation of the algorithm, illustrate the difficulties in the application of this technique. For small values of μ, the terms with the μ's in the denominator can cause trouble, while for large values, the terms $\mu_1(x-x^{**})$ and $\mu_2(y-y^{**})$ make the system extremely sensitive, and computational errors very difficult to control.

Before leaving this section, it should be pointed out that the penalty functions employed here for illustrative purposes are probably the simplest possible. Other more complex alternatives might be considered, the only restriction being that they retain the convexity properties previously mentioned. However, because of the limited success that other researchers have had with penalty function approaches in other applications, it is felt that the chances for success with this approach are not very high.

3.5 Parametric Trajectory Method (PART I)

In the previous sub-sections, three different computational algorithms have been discussed, and with each, difficulties in their application have been pointed out. The first method, the generalized Picard algorithm, is now reconsidered along with the introduction of some major modifications. This modified method, for reasons which will become evident shortly, has been called the Parametric Trajectory Method.

The primary drawback with the generalized Picard method concerns the difficulties in selecting convergent initial iterates. A class of optimization problems is now defined for which a basic technique for circumventing this difficulty can be obtained.

Basic Problem:

Consider the optimization problem of determining the control u which will minimize the functional

$$J = \int_0^{t_f} \sum_{i=1}^{N} [g_i(x_i,u_i) + \varepsilon h_i(x,u)]\, dt$$

subject to the dynamical constraints

$$\dot{x}_i = f_i(x_i,u_i) + \varepsilon l_i(x,u)$$
$$x_i(0) = x_{i_o}$$
$$u_i \in \Omega_i$$
$$i = 1,2,\ldots N$$

t_f either free or fixed,

$x_i(t_f)$ either free or fixed

where x_i is an n_i - vector, the system output

u_i is an m_i - vector, the system control, or input

x is the composite n - vector, consisting of all x_i

u is the composite m - vector, consisting of all u_i

g_i and h_i are scalar functions

f_i and l_i are n_i - vector functions

Ω_i are closed convex regions of m_i -space

$$\sum_{i=1}^{N} n_i = n \qquad n_i \geq 1 \qquad i = 1,2,\ldots N$$

$$\sum_{i=1}^{N} m_i = m \qquad m_i \geq 1 \qquad i = 1,2\ldots N$$

ε is the scalar decoupling parameter, and with no loss of generality, it can be stipulated that the solution to the problem is required at $\varepsilon = 1$.

Define

$P(\varepsilon) \triangleq$ The foregoing optimization problem as a function of the decoupling parameter ε.

$Sol(\varepsilon) \triangleq$ The solution, as described by the functions u(t) and x(t) as well as p(t), the adjoint composite variable, of the problem $P(\varepsilon)$.

With these definitions, the range of the applicability of the Parametric Trajectory Method can be stated.

Restriction on Basic Problem:

Sol (ε) exists and is unique throughout the closed interval [0,1] of the parameter ε .

An all encompassing statement of necessary and sufficient conditions required to meet this restriction, even if that were possible, is beyond the range of this thesis, and the interested reader is referred to a current survey article [12] where this question is considered in detail, and where a further list of references is provided. It might be noted that in most practical problems, if the mathematical modeling has been done correctly, these general questions rarely arise. However, in this particular case, the problem of existence and uniqueness is not unimportant, especially at $\varepsilon = 0$, since P(0) is in reality N completely isolated sub-problems, and many systems problems lose their meaning when decoupled and taken out of their systems context. This restriction, therefore, provides an illustration of the hypothesis made in part 2 of the thesis, that problems susceptible to decomposition consist of meaningful interconnected sub-problems.

Regardless of whether the foregoing restriction is satisfied or not, Cullum [25] has shown that under relatively mild conditions, Sol (ε) is continuous in ε .

The standard Parametric Trajectory Method is a modification of the generalized Picard algorithm, based on the foregoing restriction and the continuity of Sol(ε). Because P(0) consists of N independent sub-problems, it is anticipated that Sol(0) can be obtained more quickly and/or with considerably less effort than Sol(1).* Having obtained Sol(0), ε is increased by a factor $\Delta\varepsilon$ to ε_1, and Sol(0) is used as an initial iterate at the new ε value. With $\Delta\varepsilon$ sufficiently small, this initial iterate is within the (finite) region of convergence, and Sol(ε_1) can be computed. This process is then repeated for successively increasing values of the parameter ε until 1 is reached. Throughout this process, ε generates a discrete trajectory in the solution space, and this gives rise to the name of the algorithm. Figure 3.1, based on Example 3.2 to be discussed shortly, illustrates one of these trajectories in the initial co-state space. If the step size in ε is taken to be too large, the previous solution no longer serves as a satisfactory initial iterate, and the next step does not converge. One can therefore imagine a hypothetical region of convergence around each solution point, as shown in Figure 3.1, and if this region does not overlap the next solution point, smaller steps in ε are required.

* In fact, P(0) is ideally suited for parallel processing.

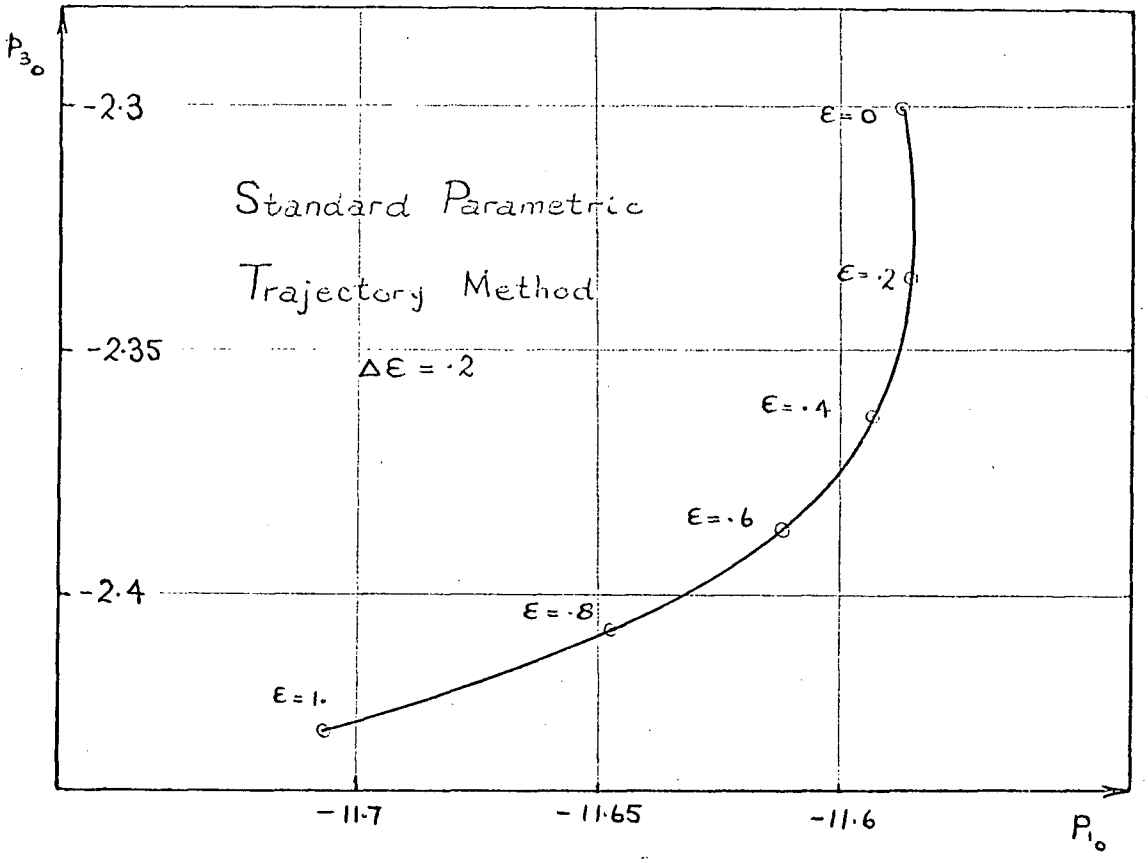


Figure 3.1 Trajectory in the initial co-state space, Example 3.2 .

The work of Cullum [25] ensures that if $\Delta\epsilon$ is chosen small enough, convergence is obtained, but it gives no information as to how small in fact $\Delta\epsilon$ should be in order to have a practical computational algorithm. Unfortunately, as in hill-climbing methods, only computational experience serves as a guide for this choice. However, in order to increase the optimal step size, the following modifications, using linear interpolation, can be employed.

Assume that $Sol(\varepsilon_n)$ has been obtained. Then, for an initial iterate for $P(\varepsilon_{n+1})$, instead of $Sol(\varepsilon_n)$, the following can be used:

$$\text{Initial Iterate} = Sol(\varepsilon_n) + \frac{\partial Sol(\varepsilon_n)}{\partial \varepsilon} \cdot (\varepsilon_{n+1} - \varepsilon_n)$$

This naturally requires the computation of $\frac{\partial Sol(\varepsilon_n)}{\partial \varepsilon}$, which can be approximated by

$$\frac{\partial Sol(\varepsilon_n)}{\partial \varepsilon} \cong \frac{Sol(\varepsilon_n + \delta\varepsilon) - Sol(\varepsilon_n)}{\delta\varepsilon}$$

where $\delta\varepsilon$ is a very small change in ε. $Sol(\varepsilon_n + \delta\varepsilon)$ is therefore obtainable in at most two iterations.

If the problem is being solved using a Newton-Raphson iteration in the initial adjoint variable space to null the final adjoint variables, the following equivalent approach might be considered. Since the initial state variables remain constant, the final adjoint variables can be written as

$$p(T) = \mathcal{P}(p_o, \varepsilon)$$

where $\mathcal{P}(p_o, \varepsilon)$ is a symbolic representation of the mapping $E^n \times E^1 \rightarrow E^n$, $p_o \in E^n$, $\varepsilon \in E^1$ and $p(T) \in E^n$, and p_o is the initial adjoint variable. Using Frechet derivatives, this operator can be expanded up to first order terms

$$\mathcal{P}(p_{o_{n+1}}, \varepsilon_{n+1}) = \mathcal{P}(p_{o_n}, \varepsilon_n) + \mathcal{P}_{p_o}|_n (p_{o_{n+1}} - p_{o_n}) + \mathcal{P}_{\varepsilon}|_n (\varepsilon_{n+1} - \varepsilon_n)$$

Assume that the correct p_{o_n} which will null p(T) at $\varepsilon = \varepsilon_n$ is known, i.e.

$$\mathcal{P}(p_{o_n}, \varepsilon_n) = 0$$

Then the linear approximation to p_o which will null p(T) with $\varepsilon = \varepsilon_{n+1}$ is given by

$$p_{o_{n+1}} = p_{o_n} - (\mathcal{P}_{p_o} |_n)^{-1} (\mathcal{P}_{\varepsilon} |_n)(\varepsilon_{n+1} - \varepsilon_n)$$

Since the operator $(\mathcal{P}_{p_o} |_n)^{-1}$ is estimated (using differences) at each value of ε, this interporation scheme requires only the additional calculation of $(\mathcal{P}_{\varepsilon} |_n)$, which is obtained in one integration sweep.

The decision whether to use a standard Parametric Trajectory Method or the linear interpolation modification is again difficult. While the interpolation promises a larger effective step size $\Delta\varepsilon$, this is at the expense of additional computation. As demonstrated in the example at the end of the next section, the interpolation did in fact result in a decrease in the overall computation time, but this decrease was not significant. However, as expected, the optimum step size was increased, and to a value which was not only non-optimum for the standard method, but which did not even provide a convergent iteration for the standard method.

3.6 Parametric Trajectory Method (PART II)

In the previous section, the Parametric Trajectory Method was introduced as a technique for circumventing the difficulties of choosing the initial iterate with the generalized Picard algorithm. Although the method accomplished this task admirably, it also eliminated the parallelism inherent in the algorithm, since neither the standard nor the interpolation modification is suited to multi-processing computers. In fact, it is easy to show that the digraph for the standard Parametric Trajectory Method, except at $\varepsilon = 0$, is strongly connected. A further modification is now proposed which retains the facility for the choice of the initial iterate, while at the same time being fully applicable to multi-processing computers.

Because of the decoupled nature of P(0), Sol(0) can be obtained on a multi-processing computer. However, in order to compute Sol(ε_1), each sub-system must have information about the interaction functions, and this requirement eliminates the usefulness of multi-processing. But, if instead of the current interaction information, all sub-systems were to use available solutions at the previous ε-value, parallelism would be retained. Naturally, the solution obtained thus at $\varepsilon = \varepsilon_1$ would then not be Sol(ε_1), but rather some approximation,

whose quality would depend on how close the previous ε-value solution was to Sol(ε_1), and how closely the sub-systems are coupled. The original basic problem, stated on page 38 , is used to illustrate these ideas. The Hamiltonian for this problem is

$$H = -\left[\sum_{i=1}^{N}\left[g_i(x_i,u_i) + \varepsilon h_i(x,u)\right]\right] + \sum_{i=1}^{N} p_i'\left[f_i(x_i,u_i) + \varepsilon l_i(x,u)\right]$$

where

$$\dot{p}_i = -H_{x_i} = g_{i_{x_i}} - f_{i_{x_i}}' p_i + \varepsilon \sum_{j=1}^{N} (h_{j_{x_i}} - l_{j_{x_i}}' p_j)$$

Assuming that the fixed time, free end point problem is under consideration, the terminal conditions on p_i are:

$$p_i(t_f) = 0$$

Maximization of the Hamiltonian results in

$$H_{u_i} = -g_{i_{u_i}} - \varepsilon \sum_{j=1}^{N} (h_{j_{u_i}}) + f_{i_{u_i}}' p_i + \varepsilon \sum_{j=1}^{N} l_{j_{u_i}}' p_j = 0$$

and it is assumed that this set of equations can be solved for u_i as

$$u_i = u_{i_1}(x_i,p_i) + \varepsilon u_{i_2}(x,p)$$

Substituting this expression for u_i in the state and adjoint equations provides the standard TPBVP:

$$\dot{x}_i = f_i(x_i, \left[u_{i_1}(x_i,p_i) + \varepsilon u_{i_2}(x,p)\right]) + \varepsilon l_i(x,u(x,p))$$

$$\dot{p}_i = g_{i_{x_i}} - f_{i_{x_i}}' p_i + \varepsilon \sum_{j=1}^{N} (h_{j_{x_i}} - l_{j_{x_i}}' p_j)$$

$$x_i(0) = x_{i_o}$$

$$p_i(t_f) = 0$$

If this TPBVP were being solved by means of quasi-linearization [26], it would be sufficient to consider Sol(ε) as the composite vector function

$$\text{Sol}(\varepsilon) \triangleq \begin{pmatrix} x(t) \\ p(t) \end{pmatrix}_{\text{opt}}$$

over the interval $t \in [0, t_f]$.

In order to obtain Sol(0), because the problem is decoupled, N smaller TPBVP's can be solved in parallel. Then, in order to retain parallelism, and assuming for example, that .25 is chosen as a step size in $\Delta\varepsilon$, instead of solving for Sol(.25), the following TPBVP is solved:

$$\dot{x}_i = f_i(x_i, [u_{i_1}(x_i, p_i) + .25u_{i_2}(\text{Sol}(0))]) + .25l_i(\text{Sol}(0))$$

$$\dot{p}_i = g_{i_{x_i}} - f'_{i_{x_i}} p_i + .25F(\text{Sol}(0))$$

$$x_i(0) = x_{i_o}$$

$$p_i(t_f) = 0$$

where

$$F(\text{Sol}(0)) = \sum_{j=1}^{N} [h_{j_{x_i}}(\text{Sol}(0)) - l'_{j_{x_i}}(\text{Sol}(0))\ p_{\varepsilon=0}]$$

If Sol(0) does not differ appreciably from Sol(.25), or if the ε terms have weak effects, the solution to this problem, called Approx(.25), does not differ appreciably from Sol(.25). At this point, two possibilities arise. First, assuming that Approx(.25) is a better approximation

to Sol(.25) than was Sol(0), the above TPBVP can again be solved, but using instead of Sol(0), Approx(.25) . This process, referred to as Picard cycling, could be continued until Approx(.25) converges to Sol(.25). Then ϵ can be increased to .50 , and the process repeated with Sol(.25) instead of Sol(0) as interactions.

The second possibility is to assume that the first Approx(.25) is a sufficiently good approximation to Sol(.25), increase ϵ to .50 immediately, and solve the new TPBVP with Approx(.25) in the interaction position. The process is then repeated with successive steps in ϵ and if the error build-up has not been extreme, Picard cycling can then be employed at $\epsilon = 1$, to obtain Sol(1). It will be shown in the following numerical example that this procedure can result in a fairly successful computational algorithm.

Example 3.2

In order to illustrate the foregoing theory, a satellite angular velocity control system is considered. In this case, it is assumed that there are three independent controls available for stabilizing the three angular velocities, and consequently, the problem is treated as a collection of three sub-systems, coupled together by the products of inertia terms.

The problem is to choose u in order to minimize the functional

$$J = \tfrac{1}{2} \int_0^1 \sum_{i=1}^{3} (10 x_i^2 + u_i^2)\, dt$$

subject to the constraint

$$\dot{x}_1 = - x_1 + \varepsilon x_2 x_3 + u_1 \qquad x_1(0) = 5$$

$$\dot{x}_2 = - x_2 + \varepsilon 2 x_1 x_3 + u_2 \qquad x_2(0) = -5$$

$$\dot{x}_3 = - x_3 - \varepsilon 3 x_2 x_1 + u_3 \qquad x_3(0) = 1$$

$$|u_i| \leq 10 , \quad i = 1, 2, 3$$

$x_i(1)$ are free.

With the aid of the Maximum Principle, this optimization problem can be converted to the following TPBVP:

$$\dot{x}_1 = - x_1 + \varepsilon x_2 x_3 + 10\, \mathrm{sat}(p_1/10) \qquad x_1(0) = 5$$

$$\dot{x}_2 = - x_2 + \varepsilon 2 x_1 x_3 + 10\, \mathrm{sat}(p_2/10) \qquad x_2(0) = -5$$

$$\dot{x}_3 = - x_3 - \varepsilon 3 x_2 x_1 + 10\, \mathrm{sat}(p_3/10) \qquad x_3(0) = 1$$

$$\dot{p}_1 = 10 x_1 + p_1 - \varepsilon (2 x_3 p_2 - 3 x_2 p_3) \qquad p_1(1) = 0$$

$$\dot{p}_2 = 10 x_2 + p_2 - \varepsilon (x_3 p_1 - 3 x_1 p_3) \qquad p_2(1) = 0$$

$$\dot{p}_3 = 10 x_3 + p_3 - \varepsilon (x_2 p_1 + 2 x_1 p_2) \qquad p_3(1) = 0$$

where

$$\mathrm{sat}\,(y) = \begin{cases} y & \text{if } |y| < 1 \\ 1 & \text{if } y \geq 1 \\ -1 & \text{if } y \leq -1 \end{cases}$$

The parameter ε has been included explicitly, but it is assumed that the original problem requires a solution with $\varepsilon = 1$. Clearly, the problem falls into the basic category for the parametric trajectory method, since it is evident that a unique solution exists for all values of ε in the closed interval $[0 , 1]$. However, because of the discontinuities in the right hand side of the equations, the efficient method of quasi-linearization is inapplicable, and, therefore, a technique based on a Newton-Raphson algorithm on zeroing the final p - vector with the choice of the initial p - vector was employed to obtain the solution.

The solutions were obtained using all the different modifications of the parametric trajectory method, and a comparison of the computer times is provided in Table 3.1 . These computer times refer to the actual computation time with an IBM 7044 computer, and do not include compiling or assembly of the program. These times are rather sensitive to factors such as the norms determining the convergence of an iterative algorithm, and consequently, these numbers are to be considered as a confirmation of anticipated trends rather than as useful quantitative data.

A Step Size	B No. of Steps	C	D	E	F
1.0	1	no conv	51.2	142.0	123.6
.5	2	51.4	63.5	124.8	105.5
.33	3	62.7	99.6	141.4	83.3
.25	4	77.7	106.5	170.7	81.2
.20	5	84.3	123.2	-	85.8
.166	6	-	-	-	84.9
.143	7	-	-	-	89.3

Table 3.1 : A comparison of computation times with different modifications of the Parametric Trajectory Method, Example 3.2 , all times quoted in seconds.

Column A : Step size in parameter ε .

Column B: Number of steps in parametric trajectory.

Column C : Standard Parametric Trajectory Method.

Column D : Standard Method with Linear interpolation.

Column E : Parallel modification, Picard cycling at each ε step.

Column F : Parallel modification, Picard cycling at $\varepsilon = 1$.

Figure 3.2 and 3.3 illustrate the optimal trajectories and the Lagrange multipliers, respectively, at $\varepsilon = 1$, while Figure 3.4 depicts graphically the various computation times. Here, the optimum step sizes are clearly evident. In fact, these occur rather logically, in that the optimum step sizes for the two standard techniques tend to be greater than those for the parallel modifications. For the standard methods, as expected, the linear interpolation increases the optimum step size, and in this example, the overall computation time is also slightly decreased. In the case of parallel methods, the smallest optimum step size, as expected, was obtained with the terminal cycling modification, and this occurred at $\varepsilon = .25$. Using Picard cycling at each ε step tended to consume too much time, and since the terminal cycling was capable of producing convergence, this modification seemed highly preferable when computation times were taken into consideration.

In comparing the overall computation times for the different methods, it should be mentioned that all times were obtained using a standard sequential general purpose computer. If a multi-processing computer with at least three processors were available, and if the executive program did not use a significant proportion of the overall computation time, then it is reasonable to assume that the parallel modifications should have their computation

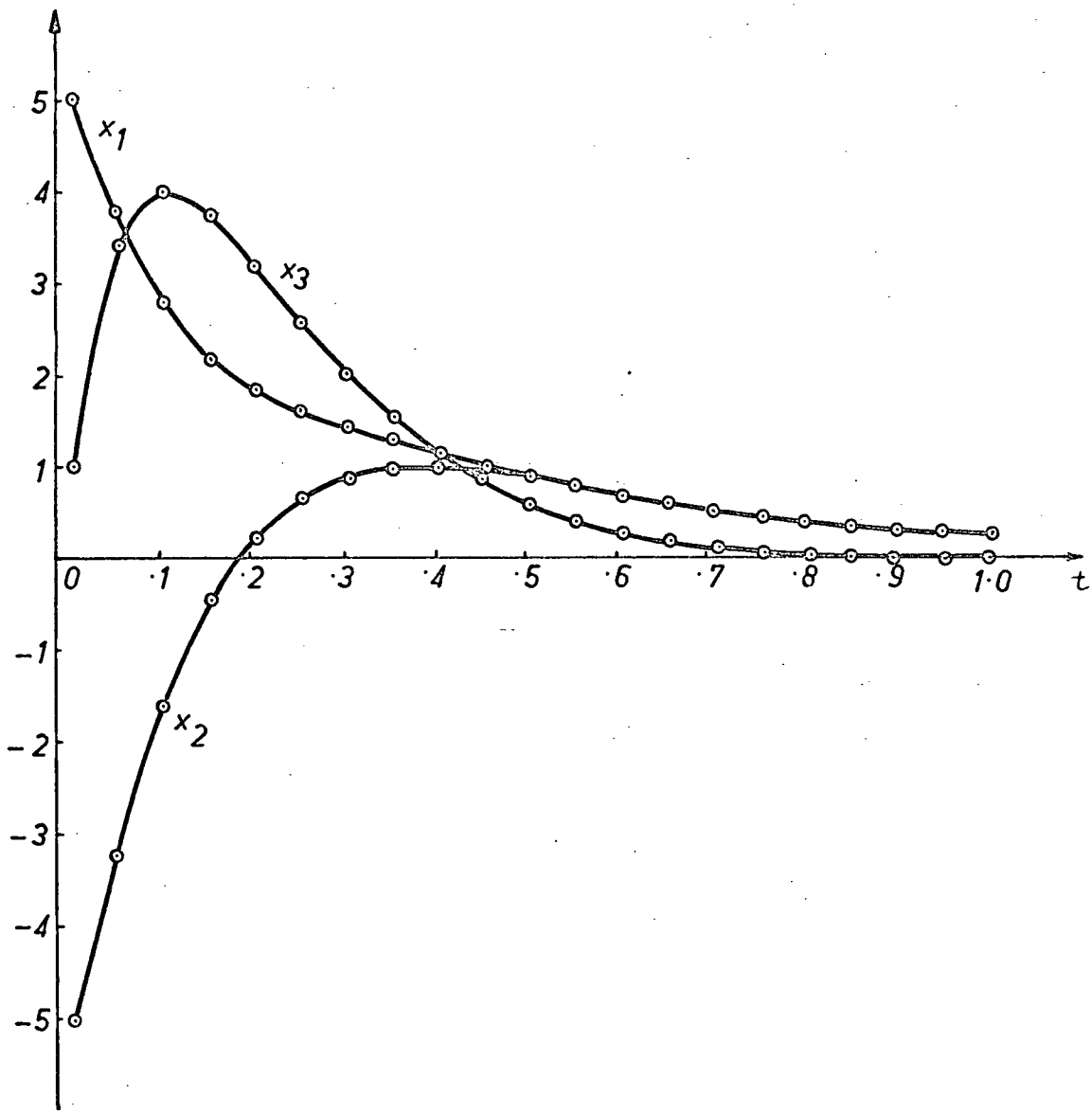


Figure 3.2 : Optimal State Trajectories, Example 3.2 , $\varepsilon = 1$ ..

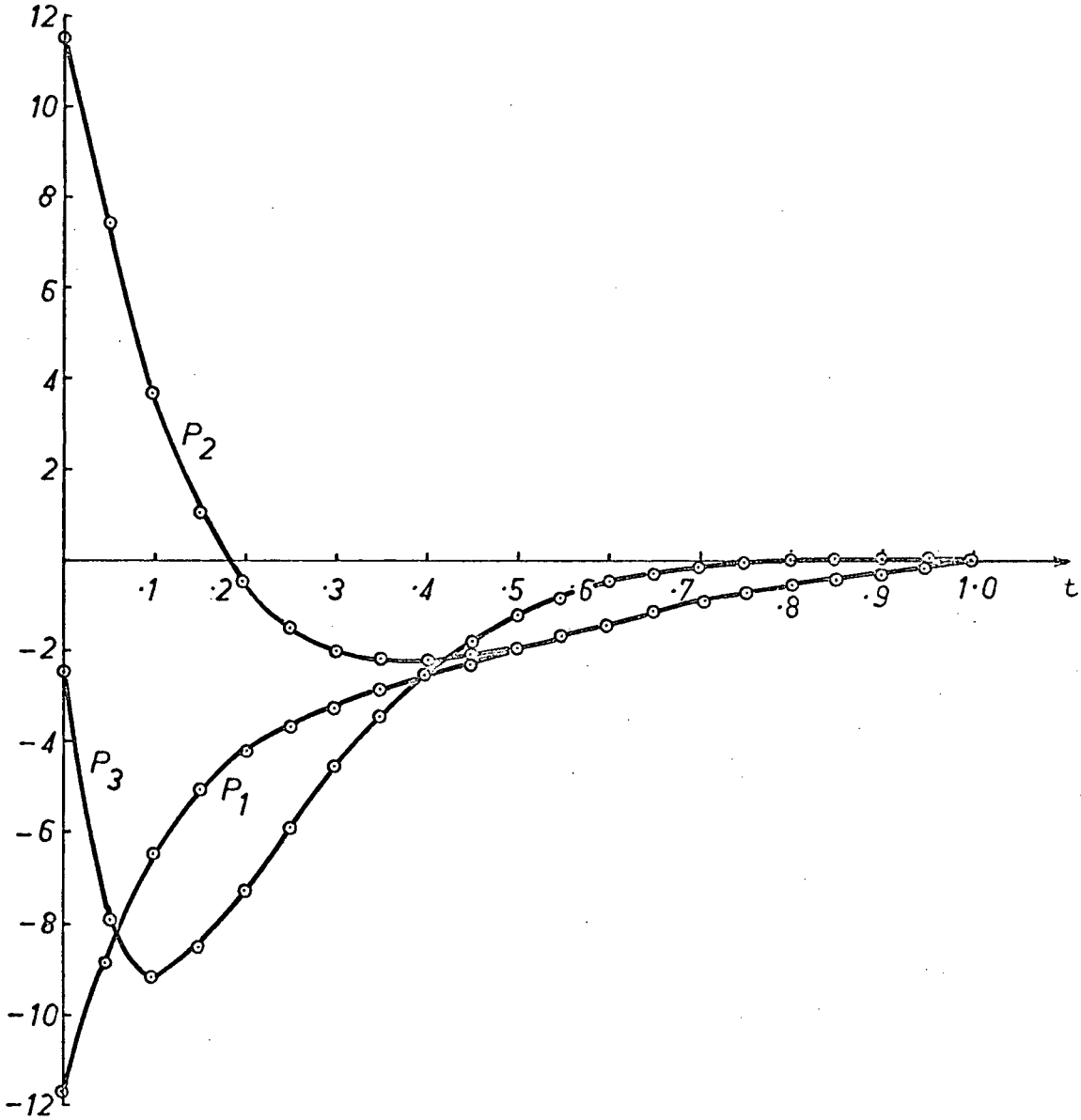


Figure 3.3 : Optimal Co-state Trajectories, Example 3.2 , $\varepsilon = 1$.

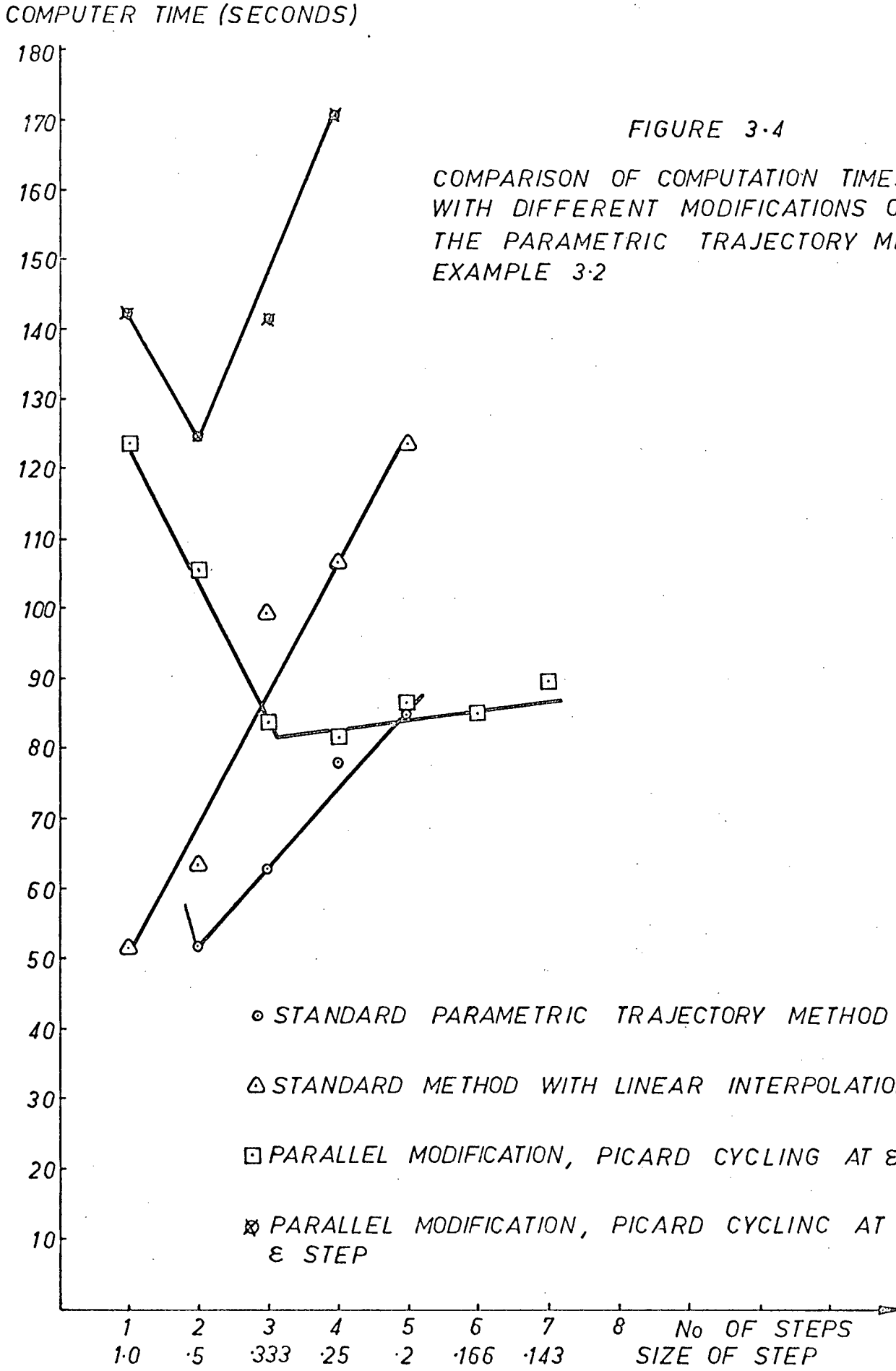


FIGURE 3·4

COMPARISON OF COMPUTATION TIMES WITH DIFFERENT MODIFICATIONS OF THE PARAMETRIC TRAJECTORY METHOD. EXAMPLE 3·2

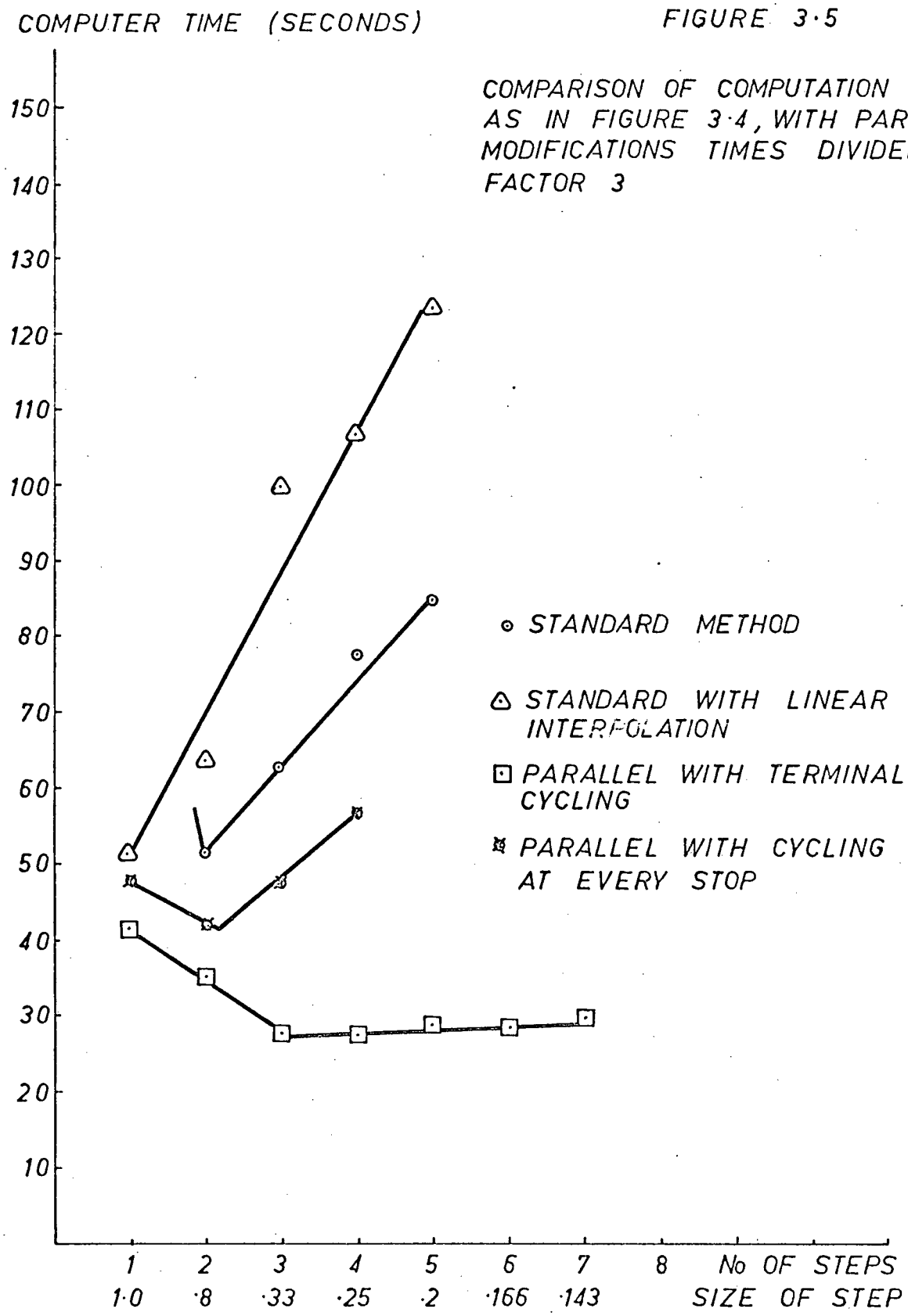


FIGURE 3·5

COMPARISON OF COMPUTATION TIMES, AS IN FIGURE 3·4, WITH PARALLEL MODIFICATIONS TIMES DIVIDED BY FACTOR 3

times cut be a factor nearly equal to three. Taking this time division factor to be the ideal, three, the new computation times are plotted in Figure 3.5 . Clearly, the relative merits of the different methods take on new perspective. The shortest computation time is now obtained by the parallel method with terminal Picard cycling, and even the less efficient method of cycling at each ε step provides a lower optimum computation time than the standard methods. Indeed, it should be pointed out that this particular example by no means represents a system with weak coupling, and it is safe to assume that if this were the case, the improvements offered by the parallel methods would be even more significant. Furthermore, as will be shown by the analysis in the next section, greater improvements in computing times can also be expected if systems of higher order are considered.

3.7 Discussion and Conclusion

In the previous five sections, various methods compatible with the subroutine type of parallelism have been considered. The theory of directed graphs, although of little practical computing significance here, has been suggested as an underlying and unifying concept illustrating the re-structuring of the different algorithms for use with multi-processing computers.

All the methods described, except the standard

parametric trajectory method which is not suitable for multi-processing, essentially achieve their parallelism at the expense of inter-subsystem iteration. In order to make these statements concrete, the following elaboration is presented. Assume that a Newton-Raphson algorithm requiring integration sweeps is employed, and suppose the system under consideration consists of N sub-systems, each represented on the average, by n differential equations. Let I_i represent the number of iterations required to solve the optimization problem using some integrated form of the algorithm, and since $[(Nn)^2 + Nn]$ equations are to be integrated per iteration, the total sequential solution time, T_i, assuming a single integration of one equation requires t_i seconds, is

$$T_i = (Nn)(Nn + 1) I_i t_i$$

On the other hand, if a multi-processing computer is available, assume it takes I_p iterations to obtain a solution using a parallel algorithm. Since only $[n^2 + n]$ equations are to be integrated (sequentially) per iteration, and assuming t_p seconds are required for the integration of one equation, the total multi-processing solution time, T_p, would be

$$T_p = n(n+1)I_p t_p$$

The ratio of these is

$$\frac{T_p}{T_i} = \frac{(n + 1) I_p t_p}{N(Nn + 1) I_i t_i} \qquad 3.9$$

Into t_p has been incorporated the fact that the executive program in a multi-processor would take some, but hopefully small, fraction of the total computing time. Thus, t_p can be represented as $t_p = t_i(1 + K)$, where K should be a small positive number. Also it is clear that I_p would be significantly larger than I_i , since not only are sub-problems being solved iteratively, but these sub-problems must be iteratively co-ordinated. If it is assumed that $t_p \cong t_i$, and n is large , then the above formula can be approximated by

$$\frac{T_p}{T_i} \cong \frac{I_p}{N^2 I_i} \qquad 3.10$$

Since the successful use of a parallel algorithm requires a much more powerful computer, at least in terms of the number of central processing units, then it might be argued that in order to have a successful algorthm, the N times as powerful computer should produce an N - fold decrease in the overall computation time. In that case, the ratio I_p/I_i needs to be less than, or equal to N . As a matter of interest, the number of iterations using the parallel modification of the parametric trajectory method with terminal Picard cycling and the number with the standard parametric trajectory method are shown in Table 3.2 . From this Table, the ratio of the optimal number of iterations is 44/13 or 3.4 . Thus it is seen that for this

$\Delta\epsilon$	Standard Method	Parallel with Terminal Cycling
1.	∞	-
.5	13	55
.33	17	45
.25	20	44
.20	22	46

Table 3.2 : Total number of iterations to obtain solution, Example 3.2 .

example, this technique places the computational process in the region of diminishing returns, in that tripling the number of central processors does not cut the overall computation time by a factor of three. Nevertheless, the fact remains that the total computation can be decreased. With this iteration ratio, the use of equation 3.9 (along with the assumption that $t_p = t_i$) would indicate a theoretical computation time ratio of

$$\frac{T_p}{T_i} = \frac{(1+1)}{3(3+1)} \frac{44}{13} = .56$$

On the other hand, the actual optimum T_i for the standard method from Table 3.1 is 51.4 seconds, and the optimum T_p for the terminal cycling modification, if a multiprocessor were available, would be 27.1 seconds. Thus, the actual ratio would be

$$\frac{T_p}{T_i} = \frac{27.1}{51.4} = .53$$

indicating that this analysis provides computation time ratios of the right order of magnitude.

The fundamental difficulties associated with any optimization problem are centered on the rates of convergence, error propagation and suitable step size, either in integration or in some search procedure. The parallel processing techniques discussed in this section convert the large integrated computational problems into groups of smaller and, hopefully, more manageable sub-problems. Even so, these sub-problems employ state and co-state variables and the latter are notorious for causing computational instability and associated problems of error propagation. In an attempt to overcome this difficulty, a method has been developed which replaces the co-state vectors with a set of bounded vectors. A description is included in Appendix D. Such a technique could then be used both as an alternative approach to the sub-problem computation and as a justification for decomposing the system into fewer but larger sub-systems.

These remarks conclude the part of the thesis on computational techniques, and the next part is concerned with the synthesis of on-line controllers. The difference between these parts stems mainly from the time available for performing computations, and the fact that on-line controllers have stringent time requirements places another severe constraint on the overall problem. Consequently, while de-

composition applied to the computational problem provided results of some generality, success with on-line controllers, as will be shown presently, can be claimed only for a special class of systems.

4. DECOMPOSITION AND MULTILEVEL CONTROL SYSTEMS

Introduction

This section concerns the synthesis of on-line controllers for the optimal control problem, and emphasis is placed on the development of a hierarchical controller structure. It is evident that the standard optimization problem formulated in terms of a scalar optimization functional puts no a priori requirements on the structure of the controller, and therefore, the added complexity of the hierarchical structure must be justified by engineering considerations. These include factors such as increased reliability and ease of maintenance, adaptability to future system expansion and reorganization, spatial separation of sub-systems and the consequent problem of data communication. Indeed, by the use of weighting factors, it is conceptually possible to incorporate these considerations in a scalar performance functional, but in practice the possibilities for doing so appear remote. On the other hand, defining vector valued performance criteria is of no avail, since no useful theory exists for treating such problems. Nevertheless, the hierarchical controller should be investigated, since such structures have evolved as regulators and control systems in complex biological, sociological and industrial systems.

4.1 Hierarchical Structure

In most on-line control situations, the time constants associated with the system are sufficiently small so that any possibility of approaching the problem with the view of repeatedly solving two-point boundary problems using some iterative algorithm must be ruled out as impractical. Rather, an algebraic relationship mapping the current state vector into the control vector is required, and this in turn requires the elimination of the adjoint system. In some instances, such as the Norm-invariant system and the linear regulator, [11] , this elimination can be accomplished analytically. Other attempts [27] , [28] , have been made to achieve this numerically, but with systems with any degree of complexity, this approach becomes impractical. A third alternative is to solve some different but judiciously chosen problem, the solution of which is easy to obtain, implement the control law from this problem to the original, and then compensate for the fact that it is a different problem. This has been attempted by Friedland [29], but the applicability of his method remains to be demonstrated. A similar approach is examined and applied to a special class of problems in a subsequent section.

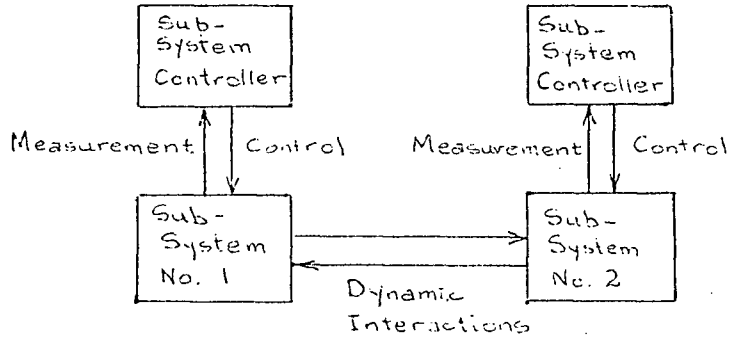
It is assumed that the set of system outputs and controls has been partitioned into a set of disjoint subsets

in such a manner that each subset contains at least one output and one input or control. This assumption restricts the class of systems studied and is closely linked with the basic restriction in Part 3 of the thesis and the hypothesis on page 17.

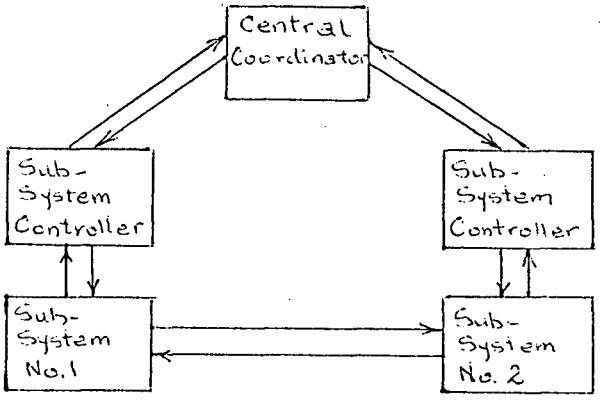
If the system posesses some suitable spatial properties, then this partitioning and assignment is obvious. For example, if the system consists of a long string of moving vehicles, each subset might consist of the position, velocity and control of one vehicle. It is further assumed that the sub-system controller can measure only the output of its sub-system and apply a control only to the sub-system. In the case of a linear generating system of the form

$$\dot{x} = Ax + Bu$$

this means that B is block diagonal, and consequently, the i'th component of u affects the j'th component of x only through the coupling in the matrix A. The structure of the system, assuming only two sub-systems for illustration, is then as shown:



Now, while the goal is still the optimal control as stated in the optimization problem in part 2, it is clear that because of the strong connectedness property of the ODS, the achievement of this goal will be jeopardized with the above structure. This is a consequence of the fact that each control must be, in general, a function of all the system outputs. If the structure is augmented by the addition of a "second-level", which receives communications from each of the sub-system controllers and in some sense, coordinates the sub-systems, then optimal control is again feasible.



The three approaches to on-line control previously mentioned are now examined with a view of incorporating such a second-level coordinator. In the first two cases, it becomes apparent that the role of the second-level is reduced to that of an information distribution centre, since the local controllers are provided with complete transformation laws, and require only the missing state information for optimal control implementation.

In the case of the third approach, what is envisioned is an active second-level which, upon obtaining state information, performs either some calculations, simulation as in 4.4 , or some maximization, as in the dual decomposition method discussed in Part 3.

To generalize, all three approaches lead to a similar hierarchical structure, but in the first two, the role of the second level is reduced to the transmission of data, while in the third, some information processing is actually performed. The advantages of this structure decline when the processing capability required of the coordinating second level begins to approach the capability of a single integrated controller, in which case, the additional engineering considerations mentioned in the introduction lose their relevance.

4.2 An Inverse Problem

Before pursuing this line of thought any further, it is important to inquire into the following inverse problem: Under what conditions is the second-level central coordinator not required? Or to state it in a different manner, under what conditions can the generating system be partitioned in such a way that the i'th subset of the control variables requires only the i'th subset of the output variables in order to generate the optimal control?

Mathematically, this inverse problem can be formulated

within the framework of the following optimization problem:

Given a dynamical system

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

find u which will minimize the functional

$$J = \int_0^T g(x,u)\, dt$$

where x is the output n-vector,

u is the control m-vector.

Under what conditions is it possible to write the optimal control law, $u^* = u(x)$, as $u_1^* = u_1(x_1)$, $u_2^* = u_2(x_2)$, where $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, x_1 is an n_1-vector, x_2 is an n_2-vector, $n_1 + n_2 = n$, and where $u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$, u_1 is an m_1-vector, u_2 is an m_2-vector, $m_1 + m_2 = m$?

In fact, this inverse problem can again be divided into two cases. The first concerns the standard case, wherein $u_i^* = u_i(x_i)$ is restricted to be an algebraic mapping of part of the state space into part of the control space. The second allows this mapping to include differential equations. This second case, involving local state estimation, is considered in section 4.3. A preliminary report of the first problem appeared in [30], and here, some of these results are recalled, and some further observations presented.

A most useful sufficient condition which ensures that a second level coordinator is not required is available in the case of no coupling boundary conditions or control or state constraints, and this demands that the Hamiltonian of the optimization problem be decomposable into a sum of sub-Hamiltonians. For example, in the problem stated on the previous page, the Hamiltonian is

$$H(x,p,u) = - g(x,u) + p' f(x,u)$$

Therefore, if H can be re-written as

$$H = \sum_{i=1}^{2} H_i(x_i, p_i, u_i)$$

and if there are no non-decomposable boundary constraints such as

$$G(x(t_f)) = 0$$

where

$$G(x(t_f)) = \{G_i(x_i(t_f))\} \qquad i = 1,2$$

or similar constraints connecting the control variables, then the optimal control can be written in terms of algebraic mappings as

$$u_i^* = u_i(x_i) \quad , \qquad i = 1,2$$

A practical example of where such a control is used is in the design of aircraft, where, under sufficiently small disturbances about a steady state condition, the Hamiltonian can be decoupled in this manner. Consequently,

a decoupled optimal control system in autopilots is not only feasible, but standard practice.

Unfortunately, as shown in [30] , this sufficient condition is not necessary, and it would be desirable to have some simple necessary and sufficient conditions. However, even if the class of optimization problems is restricted to those with linear dynamics and quadratic performance criteria, to the best of this writer's knowledge, no entirely satisfactory criteria have been obtained.

Consider the folowing optimization problem:

Minimize

$$J = \tfrac{1}{2} \int_0^T (x'Q\,x + u'R\,u)\,dt \qquad 4.1$$

subject to

$$\dot{x} = Ax + Bu \qquad 4.2$$

The case of $B=I$, $R=I$, $T=\infty$, and $\dim(u) = \dim(x)$, was considered in [30] .

Here it is assumed that B and R are block diagonal, Q is symmetric, and equation 4.2 can be written as

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + B_{11}u_1$$

$$\dot{x}_2 = A_{21}x_1 + A_{22}x_2 + B_{22}u_2$$

where $\dim(x_i) = n_i$, $n_1 + n_2 = n$

$\dim(u_i) = m_i$, $m_1 + m_2 = m$

and m is not necessarily equal to n.

Therefore, $BR^{-1}B'$ is block diagonal, and it can be represented as

$$BR^{-1}B' = C = \begin{pmatrix} C_{11} & 0 \\ 0 & C_{22} \end{pmatrix}$$

This problem gives rise to the following TPBVP:

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + C_{11}p_1 \qquad x_1(0) = x_{1_o}$$

$$\dot{x}_2 = A_{21}x_1 + A_{22}x_2 + C_{22}p_2 \qquad x_2(0) = x_{2_o}$$

$$\dot{p}_1 = Q_{11}x_1 + Q_{12}x_2 - A_{11}'p_1 - A_{21}'p_2 \quad p_1(T) = 0$$

$$\dot{p}_2 = Q_{12}'x_1 + Q_{22}x_2 - A_{12}'p_1 - A_{22}'p_2 \quad p_2(T) = 0$$

This has a solution

$$p_1 = K_{11}x_1 + K_{12}x_2$$

$$p_2 = K_{12}'x_1 + K_{22}x_2$$

$$K_{11}(T) = 0 \quad , \quad K_{12}(T) = 0 \quad , \quad K_{22}(T) = 0$$

where the K's satisfy the well known matrix Riccati equations.

Since $B_{11}u_1 = C_{11}p_1$

and $B_{22}u_2 = C_{22}p_2$

the inverse problem requires that

$$K_{12} = 0$$

This, in turn, requires that the Riccati matrices K_{ij} satisfy the following equations :

$$\dot{K}_{11} = - K_{11}A_{11} - A_{11}'K_{11} - K_{11}C_{11}K_{11} + Q_{11} \qquad 4.3$$

$$\dot{K}_{22} = - K_{22}A_{22} - A_{22}'K_{22} - K_{22}C_{22}K_{22} + Q_{22} \qquad 4.4$$

$$K_{11}A_{12} + A_{21}'K_{22} = Q_{12} \qquad 4.5$$

In the case of $m_i = n_i = 1$, it is possible to use equations 4.3 and 4.4 to eliminate the K's from equation 4.5 entirely, thereby obtaining a necessary relationship among the A's , Q's , and C's . In the general case , this does not appear to be feasible, but some useful information can still be extracted from equations 4.3 , 4.4 , and 4.5 .

1. In case $A_{12} = A_{21} = 0$, only $Q_{12} = 0$ will satisfy 4.5 . This, of course, is the case covered by the foregoing sufficiency condition.

2. If $Q_{12} = 0$, $K_{11}A_{12} = - A_{21}'K_{22}$.

3. If $T = \infty$, $A_{11} = A_{22}$, $Q_{11} = Q_{22}$, $Q_{12} = 0$, $C_{11} = C_{22}$, then $K_{11} = K_{22}$, and if $n_1 = n_2$, then

$$K_{11} = - \int_0^\infty e^{A_{21}'t} Q_{12} e^{A_{12}t} dt$$

Furthermore, Bellman [31] shows that in this case,

a necessary and sufficient condition that at least some matrix K_{11} will satisfy 4.5 is that

$$\lambda_i + \mu_j \neq 0$$

where λ_i and μ_j are the characteristic roots of A_{12} and A_{21} respectively.

Clearly, equation 4.5 severely limits the admissible systems, especially when sensitivity considerations are included. The smallest change in A_{ij} will mean that equation 4.5 is no longer satisfied, and therefore, the controller structure with no second level is no longer capable of achieving the optimal control.

Although these conclusions have been drawn only for linear systems with quadratic performance criteria, it is expected that the severity of the requirement will carry over to other optimization problems, since many non-linear systems approach linear behaviour near the origin.

4.3 Local State Estimation

In the previous section, the inverse problem was considered wherein the control law mapping the state vector into the control vector was restricted to be algebraic. In this section, this restriction is lifted and the problem is thereby transformed into one of observability. Assuming that the local controller has a true optimum feedback control law available, but lacks only information

about the other states of the system, is it still possible to circumvent the requirement for a second-level coordinator by synthesizing local estimators, which, by measuring the local state variables, can provide a satisfactory approximation to the entire state vector? If this were indeed possible, then the cost of the added complexity of the local controllers could be weighed against the cost of an overall data transmission network, or even an integrated, centralized controller. As in the previous section, it is assumed that the state and control of the overall system have been partitioned as

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} , \qquad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

and a feedback control law of the form

$$u_i^* = u_i(x_1, x_2)$$

is available to the i'th local controller. However, this control can not be implemented because, say x_2, is unknown. Clearly, the problem of state estimation is then reduced to a problem of observability. However, because no theory of observability exists for non-linear systems, only linear systems will be considered, and in this restricted case, the theory originally developed by Kalman [32] and extended by Luenberger [33], [34] to the noise-free situation is applicable here.

The dynamical generating system to be considered is

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + B_1u_1$$

$$\dot{x}_2 = A_{21}x_1 + A_{22}x_2 + B_2u_2$$

where the outputs are

$$y_1 = C_1x_1$$

$$y_2 = C_2x_2$$

Using some infinite time, quadratic performance criterion, it is known that

$$u_1 = K_{11}x_1 + K_{12}x_2$$

$$u_2 = K_{12}x_1 + K_{22}x_2$$

where, again, the K's satisfy appropriate Riccati equations.

The problem, then, for the i'th sub-system is to generate u_i using only the information in its measurement y_i.

Using the notation that (A,C) is observable if the system

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

is observable, then the following results due to Luenberger [34] can be stated:

1. Letting $A = \{A_{ij}\}$, the foregoing inverse problem is unsolvable if (A, C_i) is unobservable, $i = 1, 2$.
2. For each (A, C_i) that is observable, $(v-1)$'th order observers can be constructed, where v is the index of observability, such that each row of u_i can be generated using the output of this observer in linear combination with y_i.

Therefore, in the case where observers can be constructed, an estimate for the entire state vector is obtained. In the case of an error arising from a discrepancy in initial conditions, this error can be made to diminish arbitrarily quickly, limited only by the fact that too fast a response would make the system extremely sensitive to any noise in the output measurements.

The question of the second-level would therefore be settled, at least for linear systems, if the particular observability conditions required were a common occurrence. Unfortunately, in the experience of this writer, these conditions have not been met by systems of any complexity. For example, in the problem of a string of moving vehicles, [35], allowing each vehicle to measure its own position and velocity along with, perhaps, those of the preceding vehicle, is still not nearly enough to satisfy the observability requirements.

Other approaches to state estimation, based on game theory or Markovian models, have been suggested, but for one reason or another, no satisfactory state estimation based on local measurements alone appears feasible. It must therefore be concluded that in most practical problems, any hope of achieving local state estimation is negligible, and in order to implement the optimal feedback control law, a second level must be established.

4.4 A Multi-level Controller Based on the Second Variation

In this section, the hierarchical controller structure is synthesized for a class of optimization problems. This class of problems contains systems so large and complex that obtaining an analytical expression for the feedback control law in the form $u^* = u(x)$ is impossible. And, for the majority of the systems considered, even functional approximations as suggested in [27] and [28] are completely impractical. Thus, not even an expensive communications network can overcome the sub-system coordination problem. Instead, an active second-level coordinator is proposed which will receive information from the lower level controllers, perform functional operations on this information, and then provide lower levels some useful coordination data. With the aid of this data, the lower level controllers can then

improve their control laws, thereby obtaining a good approximation to the optimum. Naturally, much of the control calculation is done by the local controllers, and the second-level coordinator is therefore, relatively simple in comparison to a central integrated controller.

The decomposition technique based on duality (see Part 3) was developed to operate on these principles, and Pearson has claimed it to be a workable alternative [24] . However, it will suffice to say that except in extremely simple cases of scalar, linear systems with quadratic performance criteria, any possibility of on-line application of this method appears optimistic. This becomes immediately evident when the method is applied to any problem of significance, since, in effect, the resulting second-level coordination problems turn out to be variational problems of greater difficulty than the originally stated, integrated problem. Therefore, the complexity of the second-level coordinator, even if one were feasible, would exceed the complexity of a central, integrated controller, and the gains of decomposition and hierarchy would be lost. That a successful approximation technique can be developed from this theory has yet to be demonstrated.

One of the standard approaches which might lead to a multi-level controller structure is the well known neighborhood controller, treated in detail by Kelley [36] , and by Breakwell, Speyer and Bryson [37] , and there appears

no point in further elaboration here, except to say that in general, the linearized system remains coupled, and one is left with one high level (nominal trajectory) controller, and one low level (linear regulator) controller. It has the further unfortunate property that the linear region is very small and minor perturbations tend to throw the system so far from the nominal that true optimality is lost unless new nominal trajectories can be calculated rapidly.

The controller developed here can be fully understood using only two sub-systems, and extension to N sub-systems is routine. The notation used here is defined in Appendix A. Consider the problem of determining the control vector u_i, i=1,2 , which will minimize the functional

$$J = \tfrac{1}{2} \sum_{i=1}^{2} \int_{o}^{T} (x_i' Q_i x_i + u_i' R_i u_i)\, dt \qquad 4.6$$

subject to the constraint

$$\dot{x}_i = A_i x_i + \varepsilon f_i(x_1, x_2) + B_i u_i \qquad 4.7$$

$$x_i(0) = x_{i_o} \quad , \quad i = 1, 2$$

Here, it is assumed that the function $\varepsilon f_i(x_1, x_2)$ is small in some sense, as discussed later, so that the system consists primarily of two linear systems, coupled

together weakly. Also, any small subsystem non-linearities have been lumped into the f_i functions.

With the aid of the Maximum Principle, the optimal control law can be written as

$$u_i = R_i^{-1} B_i' (-K_i x_i + h_i)$$

where K_i and h_i are governed by

$$\dot{K}_i = - A_i' K_i - K_i A_i + K_i B_i R_i^{-1} B_i' K_i - Q_i \qquad 4.8$$

$$K_i(T) = 0$$

$$\dot{h}_i = -(A_i - B_i R_i^{-1} B_i' K_i)' h_i + \varepsilon K_i f_i - \varepsilon \sum_{j=1}^{2} f_{j x_i}' (-K_j x_j + h_j) \qquad 4.9$$

$$h_i(T) = 0$$

This control law immediately suggests a hierarchical controller structure, in that the $-R_i^{-1} B_i' K_i x_i$ term in the control law acts as a local feedback control, while the $R_i^{-1} B_i' h_i$ term plays the role of the coordination function. As ε goes to zero, h_i becomes vanishingly small, and no coordination is required, while, as ε gets large, the interaction term begins to dominate the dynamics, and consequently, the linear system-quadratic performance functional control law becomes insignificant in comparison to the coordination function.

As the control law stands, no approximations have been made. On the other hand, since the integration of 4.9 requires the vector x, the present form offers merely an alternate, non-trivial, two point boundary value problem.* However, the attractiveness of this formulation becomes more evident when systems with "small" interactions are considered. Writing the control law as

$$u_{i_{opt}} = u_{A_i} + u_{B_i} \qquad 4.10$$

where

$$u_{A_i} = - R_i^{-1} B_i' K_i x_i \qquad 4.11$$

$$u_{B_i} = R_i^{-1} B_i' h \qquad 4.12$$

then in the case of weak interactions, the u_{A_i} term would dominate the control while the u_{B_i} term plays the role of a secondary improvement. Of course, under the stated conditions, the u_{A_i} can be implemented exactly by the local controllers. It is suggested therefore, that the u_{B_i} term, which is difficult to implement exactly, be approximated. Because of the secondary nature of this term, this approximation need not be particularily good. Thus, instead of using 4.12 in 4.10, u_{B_i} is chosen to be

* In fact, limited computational experience has shown no particular advantages with this formulation.

$$u_{B_i} = R_i^{-1} B_i' h_i^* \qquad 4.13$$

where

$$h_i^* = -(A_i - B_i R_i^{-1} B_i' K_i)' h_i^* + \varepsilon K_i f_i(x^*) - \varepsilon \sum_{j=1}^{2} f_{j_{x_i}}'(x^*)(- K_j x_j^* + h_j^*) \qquad 4.14$$

$$h_i^*(T) = 0$$

In 4.14 , x^* is taken to be somesuitable, a priori estimate of the state trajectory. A natural choice for x^* would be

$$x^*(t) = U(t)\, x_o \qquad 4.15$$

where

$$\dot{U} = (A - BR^{-1}B'K)\, U \quad , \qquad U(0) = I \qquad 4.16$$

Here, and in the subsequent development, the non-subscripted variables and matrices are taken to mean the composite values, made up of the subscripted varibles. For example,

$$A = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}$$

The following definitions are now employed. The Approximate Control Law, u_A , is taken to be

$$u_A = - R^{-1} B' K x \qquad 4.17$$

and the value of the performance functional J arising from the use of this control law is defined to be J_A .

The Compensated Control Law , u_C , indicating that some compensation for the coupling and the non-linearities has been made, is taken to be

$$u_C = - R^{-1}B'Kx + R^{-1}B'h^* \qquad 4.18$$

and the value for the corresponding functional J is defined as J_C .

The Optimal Control Law , u_{opt} , is defined as

$$u_{opt} = - R^{-1}B'Kx + R^{-1}B'h \qquad 4.19$$

with the corresponding performance functional J_{opt}

Naturally, the complexity of the controller for implementing these three control laws progressively increases from 4.17 to 4.19 , and it is important to study the improvements in J , if any, which are to be expected by using the more complex control laws. Ideally, it is desired to obtain analytic expressions for the functionals J_A, J_C, and J_{opt}, and then compare the values of these for different coupling functions and ε's . However, the problem is too complex to allow such a course of action, and therefore three alternatives are considered.

The first of these involves obtaining, for a rather trivial problem, approximate analytic expressions for the J's , and this development is relegated to Appendix C. The second, which is treated at the end of this section, in-

volves numerical simulation of different systems to illustrate the effects of these control laws on the performance functionals. The third is a development based on the second variation approach of the calculus of variations to support the assertion that J_C is less than J_A under normal circumstances, and is presented here.

Application of the control law u_A of equation 4.17 to the system 4.7 results in the state trajectory x_A and the performance functional J_A. A variation in u_A, called δu, results in a variation of the state trajectory, δx, about x_A, and in a variation of the performance functional, δJ, about J_A. Expansion of 4.7 about this nominal trajectory, retaining terms up to the second order, yields the following differential relationship between δx and δu :

$$\dot{\delta x} = (A + \varepsilon f_x(x_A))\delta x + B\delta u + \frac{\varepsilon}{2}\eta \qquad 4.20$$

$$\delta x(0) = 0$$

where the notation and the second order term η is defined in Appendix A. Also, the variation, δJ, expanded up to second order terms becomes

$$\delta J = \int_0^T (x_A'Q\delta x + u_A'R\delta u)dt + \tfrac{1}{2}\int_0^T (\delta x'Q\delta x + \delta u'R\delta u)\,dt \qquad 4.21$$

On substituting control law 4.17 and integrating the time derivative term, this expression can be re-written as

$$\delta J = \varepsilon \int_0^T (f'K + x_A' K f_x)\delta x \, dt + \tfrac{1}{2} \int_0^T (\delta x' Q \delta x + \delta u' R \delta u + \varepsilon x_A' K \eta) \, dt \qquad 4.22$$

Since J_A is not the minimum value of J, there exists some δu such that $\delta J < 0$. In fact, a new optimization problem of determining the control δu which will minimize 4.22, subject to the constraint 4.20, can be posed. Thus, an auxiliary Hamiltonian is defined as

$$H_A = -\varepsilon(f'K + x_A' K f_x)\delta x - \tfrac{1}{2}(\delta x' Q \delta x + \delta u' R \delta u + \varepsilon x_A' K \eta) + q'([A + \varepsilon f_x]\delta x + B\delta u + \tfrac{\varepsilon}{2}\eta) \qquad 4.23$$

where

$$\dot{q} = Q\delta x - (A + \varepsilon f_x + \tfrac{\varepsilon}{2}\Lambda')'q + \varepsilon(Kf + f_x' K x_A + \Lambda K x_A) \qquad 4.24$$

$$q(T) = 0$$

and Λ, the second order term in δx, is defined in Appendix A.

Application of the Maximum Principle to this auxiliary problem results in

$$\delta u = R^{-1} B' q \qquad 4.25$$

and the TPBVP arising from equations 4.20, 4.24 and 4.25 has as a solution

$$q = -L\delta x + \Phi \qquad 4.26$$

where

$$\dot{L} = -L(A + \varepsilon f_x) - (A + \varepsilon f_x)'L + LCL - Q \qquad 4.27$$

$$\dot{\Phi} = -(A + \varepsilon f_x - CL)'\Phi + \frac{\varepsilon}{2}(L\eta - 2\Lambda[-L\delta x + \Phi]) +$$

$$\varepsilon(K f + f_x' K x_A + \Lambda K x_A) \qquad 4.28$$

$$C = BR^{-1}B' \quad , \quad L(T) = 0 \quad , \quad \Phi(T) = 0 \quad . \qquad 4.29$$

It is appropriate to define the matrix

$$M \triangleq K - L \qquad 4.30$$

and by direct substitution, using 4.8 and 4.27 , it is seen that

$$\dot{M} = -(A + \varepsilon f_x - CK)'M - M(A + \varepsilon f_x - CK) - MCM$$

$$+ \varepsilon(Kf_x + f_x' K) \qquad 4.31$$

$$M(T) = 0$$

Then the equation for Φ can be re-written as

$$\dot{\Phi} = -(A + \varepsilon f_x - CK + CM)'\Phi + \varepsilon(Kf + f_x' K x_A)$$

$$+ \frac{\varepsilon}{2}\left[(K-M)\eta + 2\Lambda(K-M)\delta x - 2\Lambda\Phi + 2\Lambda K x_A\right] \qquad 4.32$$

$$\Phi(T) = 0$$

The optimum control variation therefore becomes

$$\delta u = R^{-1}B'\left[(M-K)\delta x + \Phi\right] \qquad 4.33$$

and then, if instead of the control law u_A , the law

$$\begin{aligned} u_S &= u_A + \delta u \\ &= R^{-1}B'\left[- Kx_A + (M-K)\delta x + \Phi\right] \\ &= R^{-1}B'\left[- K(x_A + \delta x) + M\delta x + \Phi\right] \end{aligned} \qquad 4.34$$

were used, a performance functional, J_S , would result which would be less than J_A , if expansions up to the second order terms provide a sufficiently good approximation to the original problem. It is important to note that if g is defined as

$$g \triangleq M\delta x + \Phi \qquad 4.35$$

then the equation governing g is found to have a structure identical to equations 4.9 and 4.14 , namely,

$$\dot{g} = - (A - CK)'g + \varepsilon Kf(x_A) - \varepsilon f_x'(x_A)(-Kx_A + g) \qquad 4.36$$

$$g(T) = 0$$

And the equation for u_S , noting the similarity to 4.18 and 4.19 , is

$$u_S = R^{-1}B'(- Kx_S + g) \qquad 4.37$$

where x_S is the state trajectory resulting from the control u_S .

The effectiveness of the control law 4.37 is dependent on how closely the original system can be approximated by second order expansions. If these approximations

are good, and this is to be expected with normal dynamical systems, then u_S is a control law close to the optimum, and therefore $J_S < J_A$. Since h^* is an approximation to g, then under normal circumstances, it is also expected that the control law u_C will provide

$$J_S \cong J_C < J_A$$

The controller structure envisaged is then as shown in figure 4.1. Each controller continuously measures its local state variable, and at time zero, transmits this value to the coordinator. The latter, after obtaining $x(0)$, generates and stores h^* by means of one integration sweep of 4.14 in conjunction with equation 4.15. Then, throughout the time interval $[0, T]$, it supplies h^* continuously to the sub-system controllers which combine this with their local feedback control law, $-K_i x_i$, to form u_{C_i}. Naturally, in an on-line situation, this process would be carried out on a sampled basis, sampling period being T seconds.

If the sub-system transients were sufficiently slow, it might even be feasible to implement control law u_S, 4.37. However, this would require the coordinator to make two integration sweeps, a forward sweep to generate x_A, and then a backward sweep to obtain g.

Before proceeding to the numerical examples, the special

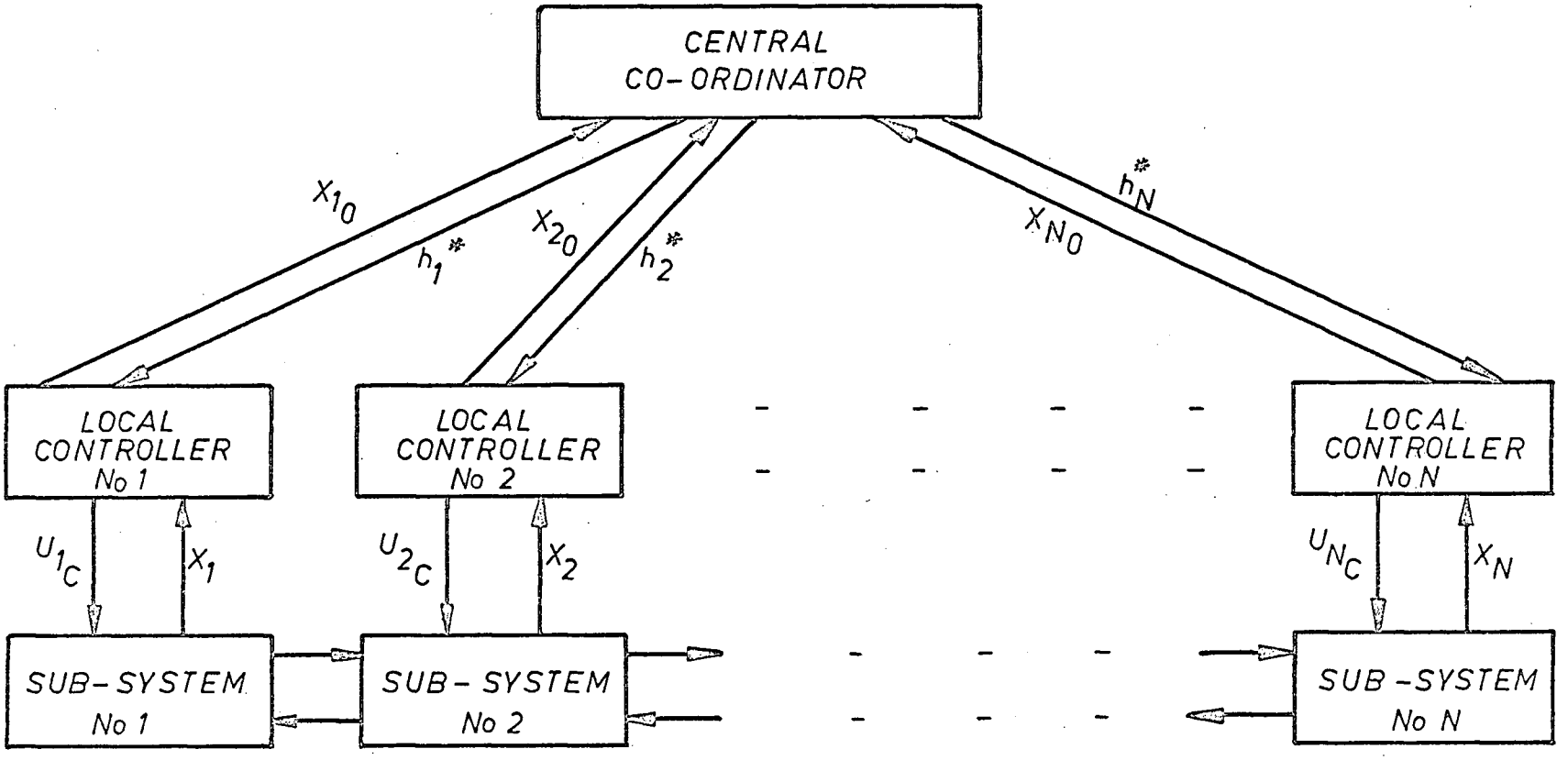


Figure 4.1: Proposed Hierarchical Structure for Controller

case of linear coupling will be considered. If the coupling function, $f(x)$, can be represented to a satisfactory degree of accuracy by the form

$$f(x) = F(t)x \tag{4.38}$$

then $\eta = 0$, and $\Lambda = 0$, and the equations for M and Φ reduce to

$$\dot{M} = -(A + \varepsilon F - CK)'M - M(A + \varepsilon F - CK) - MCM + \varepsilon(KF + F'K) \tag{4.39}$$

$$\dot{\Phi} = -(A + \varepsilon F - CK + CM)'\Phi + \varepsilon(KF + F'K)x_A \tag{4.40}$$

Moreover, the equation for g becomes

$$\dot{g} = -(A + \varepsilon F - CK)'g + \varepsilon(KF + F'K)x_A \tag{4.41}$$

$$g(T) = 0$$

Because of the linearity of these equations, it becomes feasible to write the explicit solution for g as

$$g(t) = \left[\varepsilon Y(t) \int_t^T Z(s)\{K(s)F(s) + F(s)'K(s)\}Z(s)ds\right] x_o \tag{4.42}$$

where

$$\dot{Y} = -(A + \varepsilon F(t) - CK(t))'Y$$

$$Y(0) = I$$

$$\dot{Z} = (A + \varepsilon F(t) - CK(t))Z$$

$$Z(0) = I$$

or $$g(t) = X(t)\, x_o$$

where $X(t)$ is the matrix in the brackets in equation 4.42. Thus, by storing or having available, the matrix $X(t)$, the coordinator can supply the lower level controllers the function $g(t)$ with no integration sweeps necessary, and in this case, the control law u_S is readily implemented. Some numerical examples are considered next.

Example 4.1

The first example concerns two, coupled, first-order systems, each with a scalar control input. The low order of the system tends to belie the complexity of its behaviour, especially in regions of the state space where the non-linear coupling effects are not insignificant. In fact, the uncontrolled system is highly unstable in most regions of the state space.

The optimization problem consists of determining controls u_1 and u_2 which minimize the functional

$$J = \tfrac{1}{2} \int_0^T (100x_1^2 + 100x_2^2 + u_1^2 + u_2^2)\, dt$$

subject to

$$\dot{x}_1 = -x_1 + u_1 + \varepsilon x_1 x_2^3 , \qquad x_1(0) = x_{1_o}$$

$$\dot{x}_2 = -2x_2 + u_2 + x_1^2 \varepsilon , \qquad x_2(0) = x_{2_o}$$

The optimal solutions for different initial conditions and ϵ values were obtained using the Riccati transformation in conjunction with the generalized Newton-Raphson algorithm, as described in [38]. Figure 4.2 shows plots of typical trajectories, starting at different initial points in the state space, and using the three different control laws, u_A, u_C and u_{opt}. Because of the near symmetry of the system about the variable x_2, only the right half plane is shown. Table 4.1 gives values of J associated with these trajectories. Table 4.2 illustrates the effect of different values of ϵ, the initial condition being held fixed at (3,3), a point in a particularly unstable region of the state space. Indeed, a slight increase in ϵ above .9 at this starting point is sufficient to allow the system to escape if only the approximate control law, u_A, is used with no compensation.

Example 4.2

The second example concerns the optimal regulation of two second order oscillators which are coupled by cross-product terms. A problem of this type could arise, for example, in flight vehicles, where the effects of the products of inertia are not negligible, and where it may be desireable to compensate for these effects in some optimal manner.

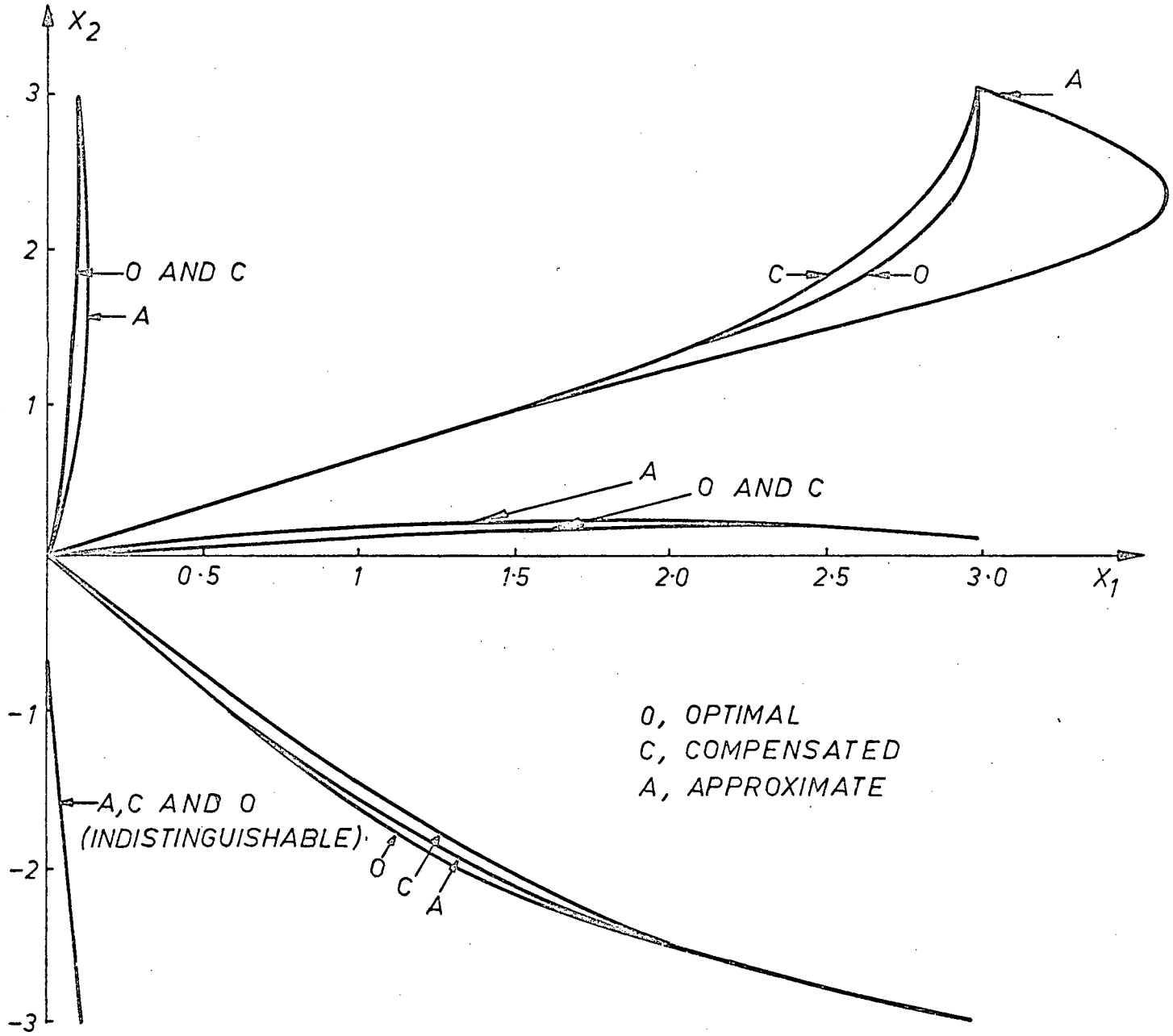


Figure 4.2 Trajectories in State Space Resulting from Different Control Laws. Example 4.1. $\varepsilon = .8$.

Initial Conditions x_{1_o}, x_{2_o}	J_{opt}	J_C	J_A	$\frac{J_C - J_{opt}}{J_{opt}}$ %	$\frac{J_A - J_{opt}}{J_{opt}}$ %
0.1, 3.0	74.01	74.01	74.04	0.	0.041
3.0, 3.0	252.15	264.73	408.91	4.99	62.2
3.0, 0.1	82.54	82.54	82.67	0.	.157
3.0,-3.0	102.87	104.06	107.44	1.157	4.443
0.1,-3.0	73.81	73.81	73.82	0.	.014

Table 4.1 : Trajectories from different initial conditions with $\varepsilon = .8$, as in Figure 4.2 , Example 4.1 .

ε	J_{opt}	J_C	J_A	$\frac{J_C - J_{opt}}{J_{opt}}$ %	$\frac{J_A - J_{opt}}{J_{opt}}$ %
.1	165.55	165.78	166.29	.14	.45
.3	188.16	189.03	195.68	.50	4.00
.5	212.70	214.92	241.67	1.0	13.60
.7	238.68	245.56	326.65	2.9	36.90
.9	265.92	288.45	594.60	8.5	123.6

Table 4.2 : Effect of different values of ε , using initial conditions (3,3) , Example 4.1 .

The problem is to determine u_1 and u_2 which will minimize

$$J = \tfrac{1}{2} \int_0^T \sum_{i=1}^{2} (q_i x_i^2 + p_i y_i^2 + u_i^2) \, dt$$

subject to

$$\left.\begin{aligned} \dot{x}_i &= y_i, & x_i(0) &= x_{i_o} \\ \dot{y}_i &= -a_i y_i + u_i + \varepsilon f, & y_i(0) &= y_{i_o} \end{aligned}\right\} \quad i=1, 2$$

where $f \triangleq y_1 y_2$

The parameters used in the numerical example were as follows :

$$q_1 = q_2 = 100$$

$$p_1 = p_2 = 1$$

$$a_1 = .5 \, , \qquad a_2 = .1 \, ,$$

and T , ε and the initial conditions were varied.

The optimal trajectories in this case were obtained using a standard parametric trajectory method along with the unmodified Newton-Raphson algorithm in function space. Tables 4.3 and 4.4 provide a comparison of performance functionals using the three different control laws for various values of T , ε and initial conditions. Figure 4.3 shows typical state trajectories with these control laws.

X_1	Y_1	X_2	Y_2	Jopt	J_C	J_A	$\frac{J_C - Jopt}{Jopt}$ %	$\frac{J_A - Jopt}{Jopt}$ %	$\frac{J_A - Jopt}{J_C - Jopt}$
.5	.5	.5	.5	17.684	17.703	17.706	.107	.124	1.159
1.0	1.0	1.0	1.0	71.440	71.511	71.609	.099	.237	2.394
1.5	1.5	1.5	1.5	162.758	162.894	163.309	.084	.339	4.036
2.0	2.0	2.0	2.0	293.559	293.753	294.707	.066	.378	5.727
2.5	2.5	2.5	2.5	466.018	466.295	467.846	.059	.392	6.644
- .5	- .5	- .5	- .5	17.504	17.517	17.573	.074	.394	5.324
-1.0	-1.0	-1.0	-1.0	70.016	70.036	70.060	.029	.063	2.172
-1.5	-1.5	-1.5	-1.5	158.054	158.244	164.893	.120	4.327	36.058
-2.0	-2.0	-2.0	-2.0	282.703	284.421	313.995	.608	11.069	18.206
-2.5	-2.5	-2.5	-2.5	445.457	450.546	542.373	1.142	21.757	19.052

TABLE 4.3: The Effect of Different Initial Conditions on Example 4.2, with $\varepsilon = .5$, $T = 2$

ε	J_{opt}	J_C	J_A	$\frac{\%}{J_C - J_{opt}}{J_{opt}}$	$\frac{\%}{J_A - J_{opt}}{J_{opt}}$	$\frac{J_A - J_{opt}}{J_C - J_{opt}}$
.0	121.122	121.122	121.122	.0	.0	1.0
.2	121.979	122.044	124.649	.053	1.697	32.019
.4	125.505	125.571	142.153	.053	13.265	250.283
.6	131.208	133.157	207.574	1.485	58.202	39.139
.8	138.507	169.062	727.068	22.060	424.932	19.263

Table 4.4 : Effect of different values of the parameter ε in Example 4.2 , using initial conditions (-2, 2, -2, 2) and T = 2 .

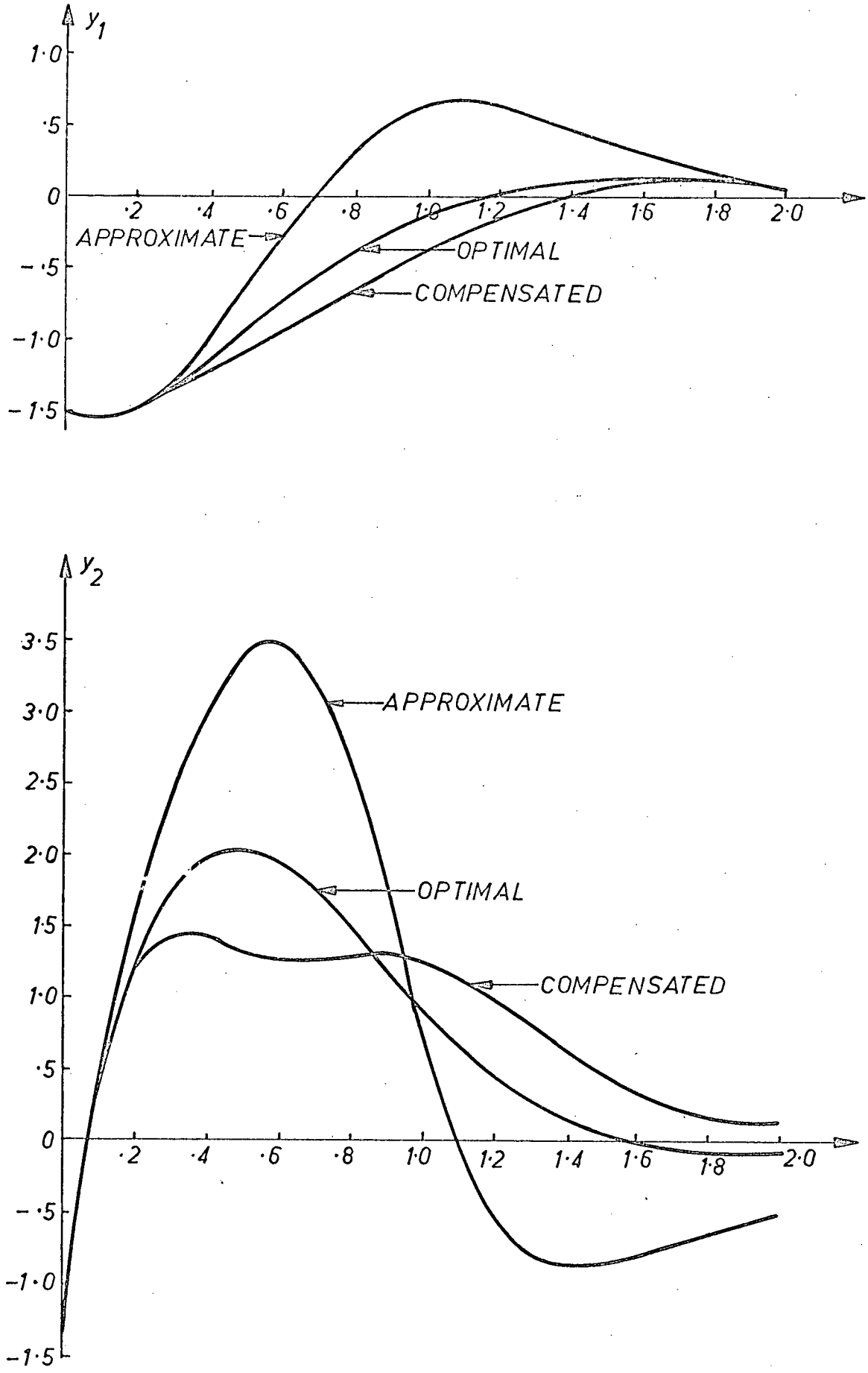


Figure 4.3: State Trajectories Resulting from Different Control Laws for Example 4.2, $\varepsilon = .5$, $T = 2$.

Discussion of Examples

The first point to note is that in every case considered, the compensated control law improved the system performance as measured by the performance functionals. This result is in agreement with the scalar problem treated in Appendix C , and might have been expected from the foregoing second variation theory. It is also noted that different portions of the state space provide very different performance criteria and this difference is reflected in the comparison of the various control laws.

Also, it was anticipated that the percentage improvement of the compensated control law over the approximate control law would not continue indefinitely with increasing values of ε , and this fact is borne out by column 7 of Table 4.4 , where the maximum percentage improvement occurs at $\varepsilon = .5$. This tendency is again verified by the scalar example.

On the whole, the compensated control law provides a performance very close to that of the optimum, and since the implementation of this control law is very much simpler than that of the optimal, it should be seriously considered. Naturally, the implementation of the approximate control law is even simpler still, but it is questionable whether the resultant performance degradation is acceptable.

4.5 Conclusion

The preceding sections have considered the optimal control synthesis problem for a class of dynamical systems. This class consists of systems composed of a number of sub-systems, preferably weakly interacting. It was first argued that, although engineering considerations may favor a completely decentralized control, the restrictions arising from the inverse problem investigation necessitate the use of controller coordination. Accordingly, a hierarchical controller structure was synthesized, and numerical examples indicate that such a controller is worth considering as an alternative to an integrated, centralized controller. Among the desireable characteristics of this controller are ease of control implementation and satisfactory behaviour in a poor inter sub-system communication environment. Inherent in the hierarchical structure is the property that if the central coordinator fails, the local controllers remain operating, and the system continues to function although further from the optimum.

5. DISCUSSION AND CONCLUSION

The thesis began with a statement of the general optimization problem under consideration, and by means of the Maximum Principle, this problem was reduced to determining the trajectory of the optimal dynamical system. The structure of this system was then represented by directed graphs, which in turn provided a basic framework from which to study different aspects of decomposition.

Decomposition was next considered both for the computational problem as well as for the synthesis problem. In the former, the fundamental contribution of decomposition was to suggest methods of parallel processing, while in the latter, decomposition was found to be the first step in synthesizing a hierarchical structure for the optimal feedback controller.

The one factor which detracts from the practical usefulness of these studies is the scope of the problem that has been considered. For the sake of mathematical convenience, the optimization problem was stated as the extremization of a scalar functional subject to differential constraints. Unfortunately, in practical situations, it appears extremely difficult to define a scalar functional, which, when extremized, guarantees a desireable control system with satisfactory performance. That a hierarchical

structure is sought in the synthesis reflects the fact that considerations other than the extremization of a scalar functional are in evidence. However, the idea that these other considerations may define other functionals, and that these may be combined to form a vector performance criteria is not impossible, but unlikely. In the first place, these considerations, which may involve ease of maintenance and readily implementable redundancy, do not lend themselves to quantitative mathematical expressions. Secondly, even if they did, the vector valued optimization problem would be of much greater complexity, and its usefulness would be questionable.

Another conclusion which arises from the thesis is that the structure of the system controller is highly dependent on not only the performance criterion but the problem statement. It was shown in Part 2 how two uncoupled generating systems become closely coupled by either the performance functional, or the boundary conditions requiring both to reach the origin in a certain time. In problems where a fixed performance functional is defined, and it is without a doubt, the one required to be optimized, then of course there is no alternative. However, if as in most synthesis problems, a performance functional is chosen merely on the basis of providing a systematic procedure for obtaining the control law, then a great deal of thought should be given to whether a different choice

for this functional might result in a far simpler controller structure, while giving almost the same system behaviour.

In conclusion, the moderately well known principle in systems engineering might be considered [39] . "Do not try too hard to optimize the small pieces of a tightly interrelated system because it will cost you more than you gained when you put the parts together.". On the other hand, the results of this thesis indicate that if the system under consideration is not "tightly interrelated", then significant gains can be anticipated by decomposition and decentralized optimization.

Appendix A : Notation

Throughout the thesis, the following notation has been employed.

$$f_x = \begin{bmatrix} f_{1x_1} & f_{1x_2} & \cdots & f_{1x_n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ f_{nx_1} & f_{nx_2} & \cdots & f_{nx_n} \end{bmatrix}$$

$$\eta = \begin{bmatrix} \eta_1 \\ \cdot \\ \cdot \\ \eta_n \end{bmatrix}$$

where

$$\eta_i = \delta x' \, f_{ixx} \, \delta x$$

and

$$f_{ixx} = \begin{bmatrix} f_{ix_1x_1} & f_{ix_1x_2} & \cdots & f_{ix_1x_n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ f_{ix_nx_1} & f_{ix_nx_2} & \cdots & f_{ix_nx_n} \end{bmatrix}$$

Furthermore,

$$\Lambda = \{\Lambda_{ij}\} = \left\{ \sum_{l=1}^{n} f_{ix_ix_j} \delta x_l \right\}$$

Note that $\Lambda' \neq \Lambda$

Using this notation, the expression for $\dfrac{\partial(s'\eta)}{\partial(\delta x)}$,

where s is any vector independent of δx becomes

$$\frac{\partial(s'\eta)}{\partial(\delta x)} = 2\left\{ \sum_{k=1}^{n} s_k \left(\sum_{j=1}^{n} f_{k x_i x_j} \delta x_j \right) \right\}$$

$$= 2 \Lambda' s$$

Also, with this notation, the following expressions can be used :

$$f(x + \delta x) \cong f(x) + f_x'(x)\delta x + \tfrac{1}{2}\eta$$

$$f_x(x + \delta x) \cong f_x(x) + \Lambda$$

Appendix B : Multi-Processing Computers

The objective of this appendix is to present a brief general description of the present status of multi-processing computers. For a more detailed study of this subject, a list of references is included, and for survey purposes, [40], [41] and [42] are particularily recommended. The machines to be considered are classified into two categories, here referred to as the conventional and the unconventional. In the former category belong machines such as the IBM 9020 [43] which have a pair to a dozen central processors plus a similar number of additional peripheral data channel control units and which represent typical present day computer hardware. The latter category includes machines such as SOLOMON II [41], with a 16 by 16 array of processing elements.

The unconventional machines, though an exciting development, are still faced with a number of not entirely unrelated problems. One concerns memory allocation, and how much local memory to provide for individual processing elements. Another involves the design of a flexible data distribution network between processing elements and the central control unit which will not be overwhelmed by hardware complexity. In the case of SOLOMON II, each processing element is provided with a local memory not exceeding one thousand words of 24 bits, while the central

program memory, also relatively small, is used for system instructions as well as acting as an overflow for the local memories. A very difficult task remaining is to develop software which achieves satisfactory hardware utilization factors. This type of machine holds great promise for the solution of partial differential equations, as indicated by some recent publications [44] , and consequently, in the context of optimal control theory, the question of explicit solutions of Bellman's partial differential equation is again raised.

The conventional machines are usually designed so that each of the processing elements, having very little or no local memory, has direct access to the central memory, which, as in the case of IBM 9020, is available in 32 K word modules. Because no single major element is critical to the operation of the system, it can be programmed to work in one extreme as a single processor sequential machine using the entire memory, or in the other extreme as a number of independent smaller computers. Thus, each processor has a vast and yet variable memory at its disposal.

However, most complex problems would probably have a degree of parallelism far in excess of that provided by a conventional multi-processor computer, and yet expanding the number of processors with this configuration

becomes impractical. The primary limitation seems to be storage interference, or more than one processor wishing access to the same memory module, as described in reference [42]. Also, although software design for the conventional multi-processing computer seems easier than for a non-conventional machine, high efficiency will almost certainly require a combination of multi-programming as well as multi-processing, and because multi-programming even on a sequential machine is a non-trivial task, a great deal of work remains to be done.

Appendix C : Scalar Example

As an addition to the development in Part 4, Section 4 , this appendix includes the derivation of approximate analytic expressions for the value of the performance functional in the case where the dynamical constraint is represented by a scalar, linear differential equation, and the performance functional is quadratic. Although in this case, decomposition is not possible, it is felt that the characteristics of the three control laws, 4.17 , 4.18 and 4.19 will be illustrated.

The problem is to choose u to minimize the functional

$$J = \tfrac{1}{2} \int_0^T (Qx^2 + u^2)\,dt$$

subject to

$$\dot{x} = ax + bx + u$$

$$x(0) = x_o$$

On the assumption that T is large, the optimal feedback control law 4.19 is known to be

$$u_{opt} = - Mx$$

where

$$M^2 - 2(a + b)M - Q = 0$$

or

$$M = (a + b) + ((a + b)^2 + Q)^{\frac{1}{2}}$$

With this control law, the optimal trajectory is governed by the equation

$$\dot{x} = (a + b - M)x$$

Letting $\alpha = -((a+b)^2 + Q)^{\frac{1}{2}}$

then the optimal trajectory can be written as

$$x_{opt} = e^{\alpha t} x_o$$

and

$$J_{opt} = \tfrac{1}{2} M x_o^2$$

However, suppose that b is not available for accurate measurement, and the approximate control law, 4.17,

$$u_A = -Kx$$

is used instead, where

$$K^2 - 2aK - Q = 0$$

or $$K = a + (a^2 + Q)^{\frac{1}{2}} \quad .$$

Then the system trajectory is governed by the equation

$$\dot{x}_A = (a + b - K)x_A$$

and by letting

$$\beta = (b - (a^2 + Q)^{\frac{1}{2}})$$

this trajectory is given by

$$x_A = e^{\beta t} x_o$$

Letting $\Delta J_A \triangleq J_A - J_{opt}$ (to first order terms)

it is found that

$$\frac{\Delta J_A}{x_o^2} = \frac{bK(\alpha - \beta)}{\beta(\alpha + \beta)}$$

or $$\frac{\Delta J_A}{J_{opt}} = \frac{bK(\alpha - \beta)}{M\beta(\alpha + \beta)}$$

On the other hand, the compensated control law, 4.18 , is given by

$$u_C = - Kx + h^*$$

where K is the same as above, and h^* is the solution of the equation

$$\dot{h}^* = - (a - K)h^* + bKx^* - b(-Kx^* + h^*)$$
$$= - \beta h^* + 2bKx^*$$
$$h^*(T) = 0$$

Choosing x^* to satisfy the equation

$$\dot{x}^* = (a - K)x^*$$
$$x^*(0) = x_o \ ,$$

letting

$$\gamma = - (a^2 + Q)^{\frac{1}{2}}$$

and assuming that T is large, the function h^* is found to be

$$h^*(t) = 2bKx_o \, e^{\gamma t} / (\beta + \gamma)$$

The trajectory x_C arising from the use of the control u_C is then approximated by

$$x_C = \left[(1 + \frac{2K}{\gamma + \beta}) \, e^{\beta t} - (\frac{2K}{\gamma + \beta}) \, e^{\gamma t} \right] x_o$$

and letting

$$\Delta J_C \triangleq J_C - J_{opt}$$

ΔJ_C is found to be

$$\Delta J_C = 2b^2K \left[1 + \frac{2K}{(\gamma+\beta)} \right] \left[\frac{1}{(\alpha+\beta)(\alpha+\gamma)} - \frac{\gamma(\gamma+\beta) - 2Kb}{2\beta\gamma(\gamma+\beta)^2} \right] x_o^2$$

These expressions were calculated for a range of values of a, b and Q, and a typical result is shown in Figure C.1. In Table C.1 are listed numerical values for $\Delta J_C / \Delta J_A$ for different values of b. It is found that for every case, this ratio remains less than 1, indicating that for the relatively wide range of values covered, the compensated control law always performed better than the approximate.

b	$\frac{\Delta J_C}{\Delta J_A}$, Q=100	$\frac{\Delta J_C}{\Delta J_A}$, Q=10
-10	.342	.985
- 8	.218	.659
- 6	.122	.388
- 4	.053	.180
- 2	.013	.047
0	.000	.000
2	.012	.051
4	.044	.207
6	.088	.465
8	.134	.775
10	.170	.979

Table C.1 Effect of b on the ratio $\Delta J_C / \Delta J_A$, for the scalar problem, with $a = -10$.

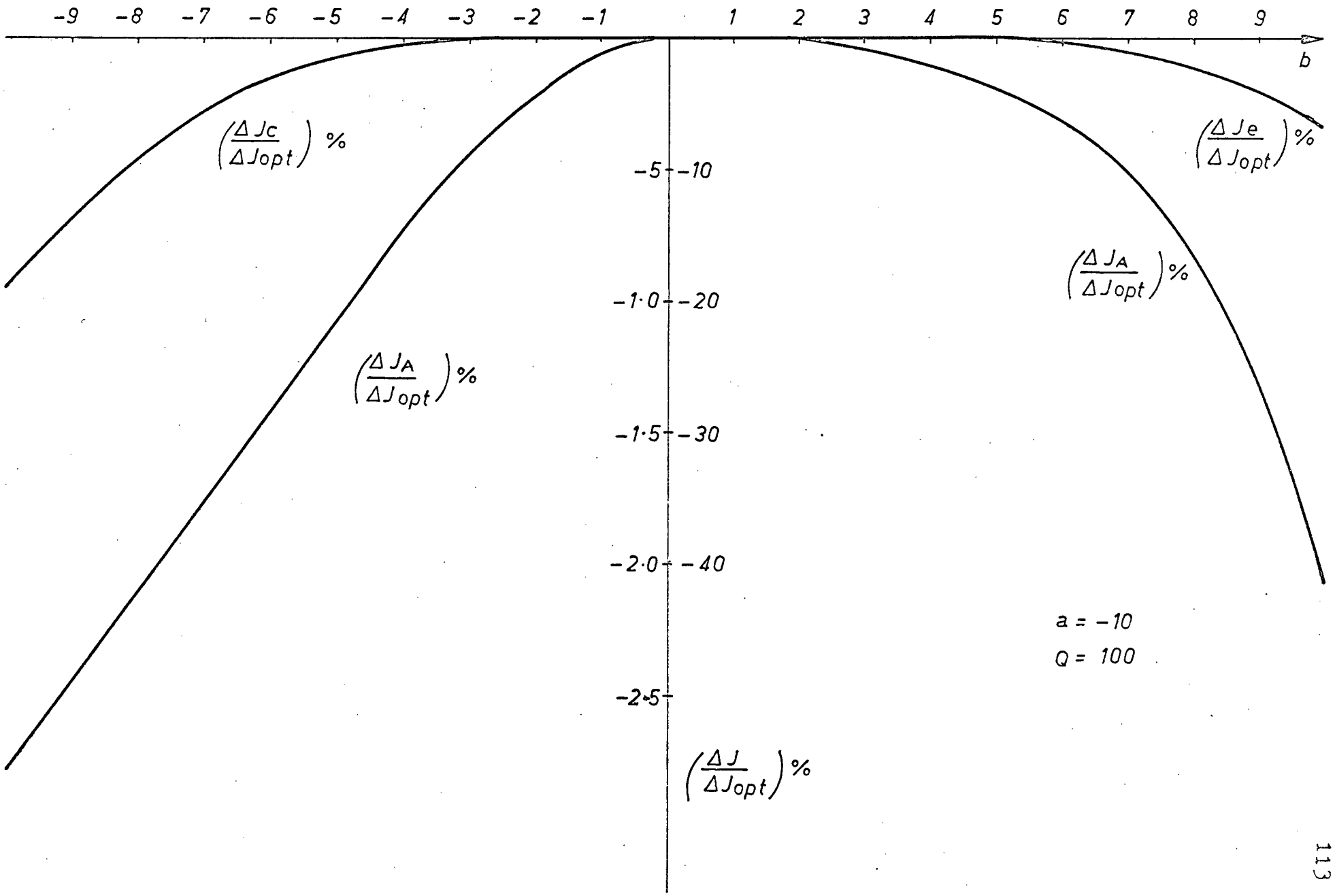


Figure C.1: A Comparison of Normalized Performance Functionals in Scalar Example

Appendix D : Tangent Plane Method

The optimal dynamical system as described in parts 2 and 3 of the thesis is defined by the state, control and the co-state variables of the problem. As is well known, these co-state variables are almost always governed by unstable equations and, consequently, the numerical difficulties associated with error propagation when the two point boundary value problem is being solved are notorious. The purpose of this appendix is to introduce a new set of variables related to the co-state variables, which possess appealing boundedness properties.

The development of this technique is based on the geometrical concepts of optimal control theory as described by Leitmann [45] , and Blaquière and Leitmann [46] , and the notation subsequently employed is identical to that of chapter 1 in reference [45] .

Consider the optimization problem of minimizing the functional

$$V = \int_{t}^{t_1} f_o\,(x,u)\,dt$$

subject to the constraint

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

and $$\Psi(x(t_1)) = 0\ .$$

This is equivalent to minimizing $x_o(t_1)$, subject to

$$\dot{x}_o = f_o(x,u)$$

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

$$\Psi(x(t_1)) = 0 \quad .$$

As in [45] , the limiting surface in the (n+1)-space is denoted by $\sum$, and the n-dimensional tangent plane to $\sum$ at the point $x^*(t)$ is denoted by $T_\Sigma(x^*(t))$. Provided the initial conditions are suitably chosen, any (n+1)-vector in this tangent plane, denoted by η , is governed by

$$\dot{\eta} = f_x^* \eta$$

where the asterisk indicates that the derivative matrix

$$f_x = \begin{bmatrix} f_{o_{x_1}} & f_{o_{x_2}} & \cdots & f_{o_{x_n}} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ f_{n_{x_1}} & f_{n_{x_2}} & \cdots & f_{n_{x_n}} \end{bmatrix}$$

is calculated at the optimum trajectory. Defining the vector λ to satisfy

$$\dot{\lambda} = -f_x^{*'} \lambda$$

Therefore, $\lambda'\eta =$ constant,

and if the initial λ is chosen as orthogonal to $T_\Sigma(x^*(t))$, then λ remains orthogonal throughout the entire trajectory. Since it can be shown [45] that λ is also orthogonal to $\begin{pmatrix} f_o^* \\ f^* \end{pmatrix}$, it follows that $\begin{pmatrix} f_o^* \\ f^* \end{pmatrix}$ lies entirely in the plane $T_\Sigma(x^*(t))$.

If $\begin{pmatrix} w_o^j \\ w^j \end{pmatrix}_{x^*(t)}$, $j = 1,\ldots.n$ represents a time-varying orthonormal set of n $(n+1)$-vectors spanning $T_\Sigma(x^*(t))$, then $\begin{pmatrix} f_o \\ f \end{pmatrix}^*$ can be represented as

$$\begin{pmatrix} f_o \\ f \end{pmatrix}^* = \sum_{j=1}^{n} a_j \begin{pmatrix} w_o^j \\ w^j \end{pmatrix} \qquad \text{D.1}$$

where because of ortho-normality,

$$a_j = (w_o^j \ , \ w^{j'}) \begin{pmatrix} f_o \\ f \end{pmatrix}^* = w_o^j \ f_o^* + w^{j'} f^* \ . \qquad \text{D.2}$$

Let $a = \begin{pmatrix} a_1 \\ \cdot \\ \cdot \\ a_n \end{pmatrix}$, $W = \begin{pmatrix} w_1^1 & w_1^2 & \cdots & w_1^n \\ \cdot & & & \cdot \\ w_n^1 & w_n^2 & \cdots & w_n^n \end{pmatrix}$, $w_o = \begin{pmatrix} w_o^1 \\ \cdot \\ \cdot \\ w_o^n \end{pmatrix}$

Therefore, D.1 and D.2 can be written as

$$f_o^* = w_o' \, a$$
$$f^* = W \, a$$
$$a = f_c^* \, w_o + W' f^* \ .$$

Combination of these equations yields

$$f_o^* = w_o' \, (f_o^* \, w_o + W' f^*)$$

and this can be re-written as

$$f_o^* = \frac{(Ww_o)' \, f^*}{(1 - w_o' w_o)} \qquad \text{D.3}$$

Define $\beta \triangleq 1 - w_o' w_o$

$$v = Ww_o$$

Equation D.3 then becomes

$$f_o^* = \frac{v_o'}{\beta} f^*$$

or

$$- f_o^* + \frac{v_o'}{\beta} f^* = 0 \qquad \text{D.4}$$

This is merely a restatement of the fact that the Hamiltonian for a non-autonomous system at the optimal trajectory is zero. Furthermore, since the standard Hamiltonian is defined as

$$H = - f_o + p'f \quad ,$$

it follows that

$$p = \frac{v}{\beta} \quad , \qquad \text{D.5}$$

where the role of β is to act as a scaling factor. The primary advantage of this new set of variables over the co-state variables , p , arises from the boundedness properties.

Because of ortho-normality,

$$w_o w_o' + W'W = I_n \quad ,$$

where I_n denotes an n by n identity matrix. Therefore,

$$w_o' (w_o w_o' + W'W) w_o = w_o' w_o$$

or $(1 - \beta)^2 + v'v = (1 - \beta)$

or $v'v = \beta(1 - \beta)$ D.6

Since $v'v \geq 0$,

then $0 \leq \beta \leq 1$ D.7

Moreover, D.5 implies that

$$\max \; v'v = \max \; \beta(1-\beta) = \tfrac{1}{4} \qquad \text{D.8}$$

With the aid of equations D.5 and D.6, the differential equations governing v and β can be derived. Since

$$\dot{p} = f_{ox} - f_x' p \quad ,$$

using D.5,

$$\frac{d}{dt}\left(\frac{1}{\beta} v\right) = \frac{\dot{v}}{\beta} - \frac{v}{\beta^2}\dot{\beta} = f_{ox} - f_x' \frac{v}{\beta}$$

or $$\dot{v} - \frac{v}{\beta}\dot{\beta} = \beta f_{ox} - f_x' v \qquad \text{D.9}$$

Therefore,

$$v'\dot{v} - \frac{1}{\beta} v'v\,\dot{\beta} = \beta v' f_{ox} - v' f_x' v \quad ,$$

which, with the aid of D.6, can be reduced to

$$\frac{1}{2}\frac{d}{dt}(\beta(1-\beta)) - (1-\beta)\beta = \beta v' f_{ox} - v' f_x' v$$

or $$\dot{\beta} = 2\beta v' \left(-f_{ox} + f_x' \frac{v}{\beta}\right)$$

Writing the Hamiltonian as

$$H = -f_o + \frac{v}{\beta}' f \quad , \qquad \text{D.10}$$

the equation for β becomes

$$\dot{\beta} = 2\beta v' H_x \qquad \text{D.11}$$

Consequently, equation D.9 can be used to obtain

$$\dot{v} = (2 v' H_x) v - \beta H_x \qquad \text{D.12}$$

The transversality conditions require that

$$\frac{1}{\beta_i} v_f = K \Psi_{x_f} \qquad \text{D.13}$$

where K is some positive constant, and the subscript f denotes terminal values. Equation D.12 can be written as

$$\frac{1}{\beta_f^2}\, v_f' v_f = K^2\, \Psi_{x_f}' \Psi_{x_f} \quad ,$$

which, with the aid of identity D.6 yields

$$\beta_f = \frac{1}{1 + K^2 \Psi_{x_f}' \Psi_{x_f}} \qquad \text{D.14}$$

Consequently,

$$v_f = \frac{K \Psi_{x_f}}{1 + K^2 \Psi_{x_f}' \Psi_{x_f}} \qquad \text{D.15}$$

With the aid of the scaling factor β and the vector v, the maximum principle together with the transversality conditions D.14 and D.15 can be used to generate a new TPBVP consisting of the state equations and equations D.11 and D.12. The requirement that H as defined in equation D.10 be a maximum with respect to the control variable u is sufficient to provide conditions under which the control variable can be replaced by a function of x, v and β.

This method is illustrated on a simple second order system in order to demonstrate its application. The problem is to drive the system

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = u$$

from its initial state $x_{1_o} = 0$, $x_{2_o} = -3$ to the circle

$$\Phi(x(1)) = x_1(1)^2 + x_2(1)^2 - 1 = 0$$

so that

$$J = \tfrac{1}{2} \int_0^1 u^2 \, dt$$

is minimized.

The standard approach is to define the Hamiltonian as

$$H = -\tfrac{1}{2} u^2 + p_1 x_2 + p_2 u$$

where

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

$$\dot{p}_2 = -p_1 \quad .$$

The maximum principle then yields

$$u = p_2$$

which, when substituted, gives H as

$$H = \tfrac{1}{2} p_2^2 + p_1 x_2 \quad .$$

The fact that $H(0) = 0$ provides a relationship between $p_1(0)$ and $p_2(0)$ as

$$p_1(0) = -\frac{p_2^2(0)}{2x_2(0)}$$

A numerical algorithm based on the generalized Newton-Raphson technique similar to that described on page 43 of the thesis is employed to determine the unknown quantity $p_2(0)$ which will null the function $\Phi(x(1))$.

The present method begins by defining the Hamiltonian as

$$H = -\tfrac{1}{2} u^2 + \frac{v_1}{\beta} x_2 + \frac{v_2}{\beta} u$$

where

$$\dot{\beta} = 2 \beta v' H_x = 2 v_1 v_2$$

$$\dot{v}_1 = (2 v' H_x) v_1 - \beta H_{x_1} = 2 \frac{v_1^2 v_2}{\beta}$$

$$\dot{v}_2 = (2\, v' H_x)\, v_2 - \beta H_{x_2} = \frac{2 v_2^2 v_1}{\beta} - v_1$$

Again the maximum principle yields

$$u = \frac{v_2}{\beta}$$

and, this, when substituted into H gives

$$H = \tfrac{1}{2} \frac{v_2^2}{\beta^2} + \frac{v_1}{\beta} x_2 \quad .$$

Thus, the $H(0) = 0$ condition implies that

$$v_2^2(0) = -2\beta(0)\, v_1(0)\, x_2(0) \quad .$$

Moreover, the identity

$$\beta(1-\beta) = v'v$$

which is valid for all t, provides the other missing initial condition as

$$\beta(0)\,(1 - \beta(0)) = v_1^2(0) + v_2^2(0)$$

or

$$\beta(0)(1 - \beta(0)) = v_1^2(0) - 2\beta(0) v_1(0) x_2(0)$$

or

$$v_1(0) = \beta(0) x_2(0) + \sqrt{\beta(0) + \beta^2(0)(x_2^2(0) - 1)}$$

and therefore,

$$v_2(0) = -2\beta(0) x_2(0) \left[\beta(0) x_2(0) + \sqrt{\beta(0) + \beta^2(0)(x_2^2(0)-1)}\right]$$

As before, the problem is reduced to finding the missing scalar quantity $\beta(0)$ such that $\Phi(x(1))$ will be nulled.

Table D.1 provides a comparison of the number of iterations required for convergence between the standard method and the tangent plane method. These results are displayed graphically in Figure D.1 .

In Table D.1 , S refers to the quantity

$$S = \frac{p_2(0) - p_2^*(0)}{p_2^*(0)}$$

and T refers to

$$T = \frac{\beta(0) - \beta^*(0)}{\beta^*(0)}$$

where $p_2(0)$ and $\beta(0)$ are the guessed initial conditions as tabulated in the first and fourth column respectively, $p_2^*(0)$ and $\beta^*(0)$ are the correct initial conditions for solving the problem, and these happen to be 6. and .0137 , respectively, for this particular problem. These normalized quantities S and T are useful for comparing the ranges of convergence of the two methods, and are used as the common abscissa in Figure D.1 . Figure D.2 shows the optimal trajectories for β , v_1 and v_2 .

As shown in Figure D.1 , the range of convergence for the tangent plane method is greater than that for the standard method. In fact, this difference is even more noteworthy when it is realized that the "no convergence" condition of the tangent plane method at $T = -.5$ occurs very close

to one extreme of β . To solve the problem, this would not be a logical initial condition with which to begin the iteration. Therefore, the fact that β is known to lie between 0 and 1 should be of significant assistance in choosing initial iterates. The same can not be said of $p_2(0)$ which may lie anywhere in the range $-\infty \leq p_2(0) \leq \infty$.

Aside from the facility of choosing initial conditions, the boundedness conditions, D.7 and D.8 , because they always apply, should hold down the rate of error propagation during forward integration. The control variable is then calculated by normalizing the v-vector with β , and both of these would be available to a good degree of accuracy.

It should be noted that with the tangent plane method, each iteration required the integration of (n+1) equations as compared to n equations for the standard method. By using the identity D.6 , it may be possible to eliminate the equation for β entirely. If such is not the case, then this drawback must be weighed against the other advantages of the method.

STANDARD METHOD			TANGENT PLANE METHOD		
Guessed Initial Condition $p_2(0)$	S	Number of iterations for Convergence	Guessed Initial Condition $\beta(0)$	T	Number of iterations for Convergence
18.0	2.	n.c.	.15	9.96	n.c.
12.0	1.	n.c.	.1	6.28	6
9.0	.5	n.c.	.05	2.65	5
7.5	.25	8	.03	1.19	4
6.5	.083	4	.02	.46	4
6.0	0.	1	.0137	0.	1
5.5	-.083	4	.013	-.05	3
3.0	-.5	6	.01	-.27	3
0.0	-1.0	8	.008	-.416	4
-6.	-2.0	10	.007	-.49	5
-12.	-3.0	11	.005	-.635	n.c.

Table D.1 : A comparison of the Number of Iterations Required for Convergence Using the Standard and Tangent Plane Methods. (n.c. means no convergence)

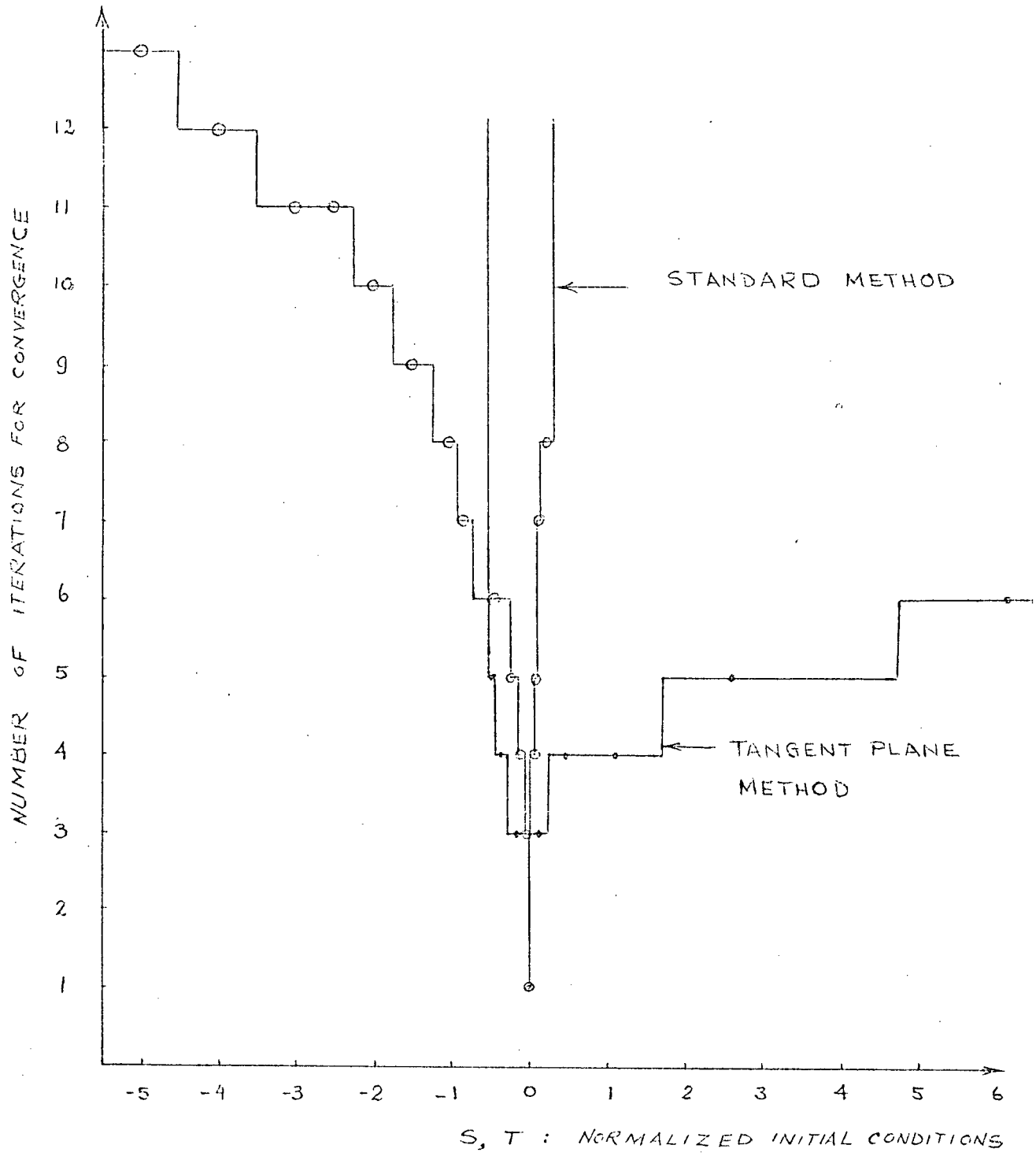


Figure D.1 : Comparison of the Regions of Convergence for the Standard Method and the Tangent Plane Method.

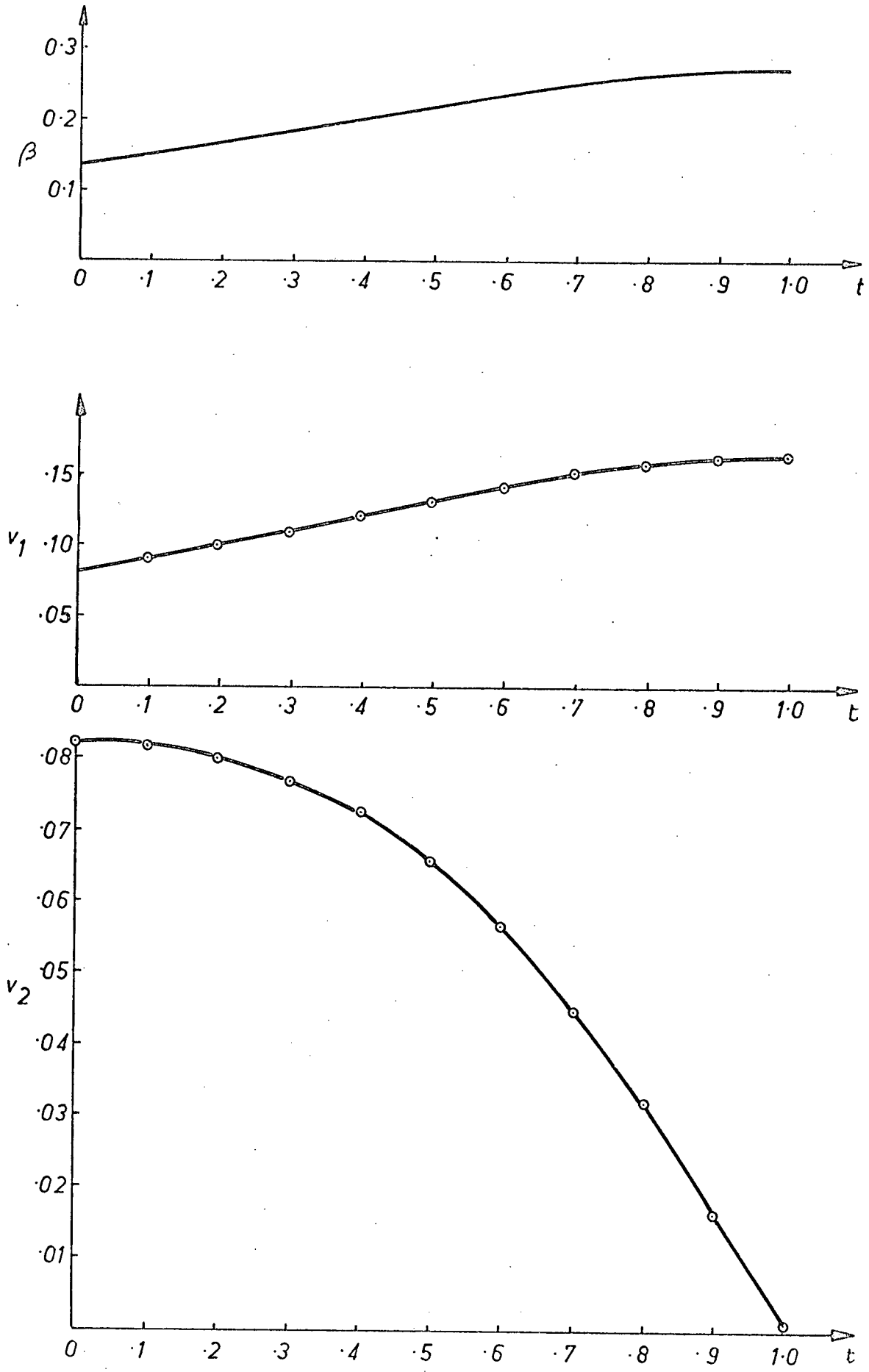


Figure D.2 : Optimal Trajectories for β , v, and v_2 .

REFERENCES

1 . Kron, G. "Diakoptics, the piecewise solution of large-scale systems" London, Macdonald (1963).

2 . Ritter, K. "A Decomposition Method for Structured Quadratic Programming Problems" Journal of Computer and System Sciences, Vol. 1, No. 3, Oct. 1967.

3 . Dantzig G. B. and Wolfe, P. "The Decomposition Algorithm for Linear Programming" Econometrica Vol. 29, No. 4, Oct. 1961.

4 . Mesarovic, M. D. "Self-Organizing Control Systems" IEEE Transactions on Applications and Industry, Sept. 1964.

5 . Kulikowski, R. "Optimum Control of Multi-dimensional and Multi-Level Systems" Advances in Control Systems, Vol. 4, Ed. C. T. Leondes, Academic Press, New York (1966).

6 . Straszak, A. "Suboptimal Supervisory Control" in Functional Analysis and Optimization edited by E. R. Caianiello, Academic Press, New York 1966.

7 . Beckenbach, E. F. "Applied Combinatorial Mathematics" Wiley and Sons, New York (1964) p. 219 et seq.

8 . Zadeh, L. A. and Desoer, C. "Linear System Theory" McGraw Hill, New York (1963).

9 . Balakrishnan, A. V. "Foundations of the State-Space Theory of Continuous Systems" Journal of Computer and System Sciences Vol. 1, No. 1, Apr. 1967, pp. 91-116.

10 . Pontryagin, L. S. et al. "Mathematical Theory of Optimal Processes" Wiley, New York, (1962).

11 . Athans, M. and Falb, P. "Optimal Control" McGraw Hill, New York, (1966).

12 . Athans, M. "Status of Optimal Control Theory and Applications for Deterministic Systems" IEEE Transactions on Automatic Control, Vol. AC-11, No. 3, July 1966.

13 . Kelly, H. J., Kopp, R. E. and Moyer, H. G. "Singular Extremals" Topics in Optimization, ed. G. Leitmann, Academic Press, New York (1967).

14 . Letov, A. M. "The Synthesis of Optimal Regulators" Proceedings of the Second Congress IFAC, p. 249, Butterworths, London (1964).

15 . Harary, F., Norman, R. Z. and Cartwright, D. "Structural Models; An Introduction to Theory of Directed Graphs" Wiley, New York (1965).

16 . Fan, L. T. and Chen, T. C. "The Continuous Maximum Principle, a study of complex system Optimization" Wiley, New York (1966).

17 . Critchlow, A. J. "Generalized Multi-processing and Multi-programming Systems" 1963 Fall Joint Computer Conference, AFIPS Conference Proceedings, Vol. 24. Academic Press.

18 . Conway, M. E. "A Multi-processor System Design" 1963 Fall Joint Computer Conference AFIPS Conference Proceedings, Vol. 24, Academic Press.

19 . Pearson, J. D. and Takahara, Y. "On the Synthesis of a Multi-level Structure" Systems Research Center Report SRC 70-A-65-25, CASE Institute of Technology.

20 . Pearson, J. D. "Duality and a Decomposition Technique" Journal SIAM on Control, Vol. 4, pp. 164-172, Feb. 1966.

21 . Pearson, J. D. "Decomposition, Co-ordination and Multi-level Systems" IEEE Transactions on Systems Science and Cybernetics, Vol. SSC-2, Aug. 1966, pp. 36-40.

22 . Courant, R. and Hilbert, D. "Methods of Mathematical Physics, Volume I" pp. 233-238, Interscience (1953).

23 . Hurwicz, L. and Uzawa, H. "A Note on Lagrangian Saddle Points" Studies in Linear and Non-Linear Programming, ed. Arrow, K. J., Hurwicz, L., and Uzawa, H. Stanford University Press (1958).

24 . Pearson, J. D. "On Controlling a String of Moving Vehicles" IEEE Transactions of Automatic Control, Vol. AC-12, No. 3, pp. 328-329, June 1967.

25 . Cullum, J. "Perturbations of Optimal Control Problems" SIAM Journal on Control, Vol. 4, 1966., No.3, pp. 473-487.

26. McGill, R. and Kenneth, P. "Solution of variational Problems by Means of a Generalized Newton-Raphson Operator" American Institute of Aeronautics and Astronautics J., Vol. 2, pp.1761-1766, October 1964 .

27 . Longmuir, A. G. and Bohn, E. V. "The Synthesis of Suboptimal Feedback Control Laws" IEEE Transactions on Automatic Control, Vol. AC-12, No. 6, Dec. 1967.

28 . Smith, F. W. "Design of Quasi-Optimal Minimum Time Controllers" IEEE Transactions on Automatic Control, Vol. AC-11, No. 1, Feb. 1968.

29 . Friedland, B. "A Technique for Quasi-Optimum Control" Journal of Basic Engineering, Series D, Vol. 88, No. 2, pp. 437-443, June 1966.

30 . Masak, M. "An Inverse Problem on Decoupling Optimal Control Systems" IEEE Transactions on Automatic Control, Vol. AC-13, No. 1, Feb. 1968, p. 109.

31 . Bellman, R. "Introduction to Matrix Analysis" p. 231, McGraw Hill, New York (1960).

32 . Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems" Journal of Basic Engineering, Series D, Vol. 82, 1960.

33 . Luenberger, D. G. "Observers for Multivariable Systems" IEEE Transactions on Automatic Control, Vol. AC-11, No. 2, April 1966.

34 . Luenberger, D. G. "Observing the State of a Linear System" IEEE Transactions on Military Electronics, April 1964. Vol. MIL-8, No. 2.

35 . Levine W. S. and Athans, M. "On the Optimal Error Regulation of a String of Moving Vehicles" IEEE Transactions on Automatic Control, Vol. AC-11, No. 3, July 1966.

36 . Kelley, H. J. "An Optimal Guidance Approximation Theory" IEEE Transactions on Automatic Control, Vol. AC-9, No. 4, Oct. 1964.

37 . Breakwell, J. V., Speyer, J. L. and Bryson, A. E. "Optimization and Control of Non-linear Systems Using the Second Variation" SIAM Journal on Control, Vol. 1, No. 2, 1963.

38 . Schley, C. H. Jr. and Lee, I. "Optimal Control Computation by the Newton-Raphson Method and the Riccati Transformation" IEEE Transactions on Automatic Control, Vol. AC-12, No. 2, pp. 139-144, April 1967.

39 . Hamming, R. W. "Numerical Methods for Scientists and Engineers" p. 375, McGraw Hill, New York (1962).

40 . Murtha, J. C. "Highly Parallel Information Processing Systems" in Advances in Computers, Vol. 7, ed. Alt, F. and Rubinoff, M., Academic Press (1966).

41 . Barnum, A. A. and Knapp, M. A. (ed.) "Workshop on Computer Organization" Spartan Books (1963)
articles by: 1) Slotnick, D. L. et al.
2) Comfort, W. T.
3) Estrin, G. et al.

42 . Lehman, M. "Survey of Problems and Preliminary Results Concerning Parallel Processing and Parallel Processors" Proceedings of the IEEE, Vol. 54, No. 12, pp. 1889-1901, Dec. 1966.

43 . IBM Staff "An Application Oriented Multiprocessing System" IBM Systems Journal, Vol. 6, No. 2, 1967.

44 . Carroll, A. B. and Wetherald, R. T. "Application of Parallel Processing to Numerical Weather Predictions" Journal for the Association of Computing Machinery, Vol. 14, No. 3, July 1967.

45 . Leitmann, G. "An Introduction to Optimal Control" McGraw Hill, New York (1966).

46 . Blaquiere, A. and Leitmann, G. "On the Geometry of Optimal Processes" in Optimization Techniques, Vol. II, ed. G. Leitmann, Academic P., New York (1967).