Application of Optimal Control Theory to Aerospace Problems

by

Yao-Chi Chen

A thesis
presented to the School of Graduate Studies and Research of
the University of Ottawa
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in
Electrical Engineering

Ottawa, Canada, 1982

To the memory

of

My Grandfather

If NECESSITY is the mother of invention, discontent is the father of progress.

... David Rockefeller

The School of Graduate Studies and Research of the University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

The subject of this thesis is devoted to the application of Optimal Control Theory to aerospace problems. In this thesis, a radar detection problem and a satellite orbital control problem are considered.

The radar problem is to optimize the search scanning function of radars with electronically steerable beams. The target dynamics considered is described by a system of first order ordinary linear differential equations with delayed arguments.

The satellite orbital control problem is to develop an optimal orbital control scheme for synchronous satellite. The satellite orbital dynamics is described by a system of nonlinear differential equations without delayed terms. The Optimal Control Scheme is compared with a Sequential Control Scheme proposed by Ahmed and Sen [16].

In this thesis, two different algorithms are adopted for the solution of the two problems. The radar scanning problem is solved by a strong variational technique originated from Murray and Teo [5]. While the Conjugate Gradient Descent (CGD) method is used in the orbital control problem. Finally, comparisons are made on the two algorithms.

ACKNOWLEDGEMENTS

The author expresses deep gratitude to his supervisor Prof. N.U. Ahmed for his encouragement, understanding and guidance throughout this work.

The author would like to thank members of the Systems and Control Group, especially to Mr. S.K. Biswas and Mr. T.E. Dabbous for their constructive discussions and encouragements.

Thanks to the National Research Council and to the University of Ottawa for financial assistance during the period of this research. Thanks are also due to the Communication Research Center for the financial support through the contract on the radar problem.

CONTENTS

Appendix page

LIST OF FIGURES

Chapter I

INTRODUCTION

1.1 GENERAL

Historically systems and control theory has its origin in the industrial revolution of the 19th century, World War II and postwar automation.

Over the last two decades a revolutionary development of the subject has taken place. This revolution was triggered by an outstanding breakthrough in the theory of classical calculas of variations in the late 50's, and the results is known as the Pontryagin's Maximum Principle [1]. Due to the size and speed of the computers available in the 50's, application of the results was somewhat limited. However, the extensive development of the modern high speed digital computers in the 60's made it feasible to consider the actual computation of the problem.

The Russians astonished the world by launching their first artificial satellite, Sputnik on October 4, 1957. Since the advent of the space age began in 1957, a number of space projects had been conducted which lead to the conquest of the moon in the 70's [2], exploration of the planets Mars, Jupiter and Saturn in the 80's, and the recent successful return of the U.S. space shuttle.

With the increased use of more sophisticated defense systems and space vechicles, modern control theory is one of the more important areas of engineering application in the military/aerospace fields. Control theory had found applications in other areas as well, notably transportation, power systems, life sciences, ecology, management sciences, operations research and economics [3].

In aerospace and defense systems, prime areas of concern have been the identification of aircraft parameters, adaptive control for aircraft, attitude control for satellite, jet engine control, radar tracking, nonlinear stochastic control for tactical missiles, man-machine systems, precision telescope pointing and tracking, and orbit transfer problem for interplanetary space vechicles [3].

1.2 PROBLEMS

In this thesis, two aerospace applications of Optimal Control Theory will be considered, namely optimal scanning problem for a phased array radar and the optimal orbital control of a synchronous satellite's orbit.

The aim of the first problem is to improve the performance of a radar system with electronically steerable beams to perform search scanning by optimizing the beam control function. The model describing the statistical behavior of undetected targets is originally developed by McAdam [4]. It gives an estimate of the certainty that various regions of a surveyed geographical area have not been entered by undetected targets. The certainty variation with time of each region is approximated by a differential equation. The target dynamics is used to formulate an optimal control problem which is then solved by an algorithm based on a strong variational technique developed by Murray and Teo [5]. The objective is to determine the beam steering function which results in the best detection statistics possible in the time available.

The second problem is to find an optimal control scheme to drive the orbital parameters namely, semi-major axis, eccentricity, longitude of ascending node, argument of periapsis, and true anamoly at epoch to their desired states. The well developed dynamics of the orbiting satellite [6,7] is used to formulate an optimal control problem which is then solved by the well known Conjugate Gradient Descent (CGD) method [8,9,10]. A practical control scheme based on the results is suggested on the design of the control systems of an orbiting satellite.

To the knowledge of the author, the target dynamics and the optimization of the radar system considered in this thesis have never been attempted by others. Furthermore, no publication on the IEEE Transactions on Aerospace and Electronics Systems has considered the optimal control of the orbital parameters of a satellite.

1.3 OUTLINE OF THE THESIS

In the second chapter a brief review of the theoretical background for the Pontryagin's Maximum Principle is made. The theorems presented are used in the solution of the problems described previously.

In the third chapter the model of the statistical behavior of undetected targets is derived. The model for the radar scanning problem is used to formulate an optimal control problem. The algorithm based on strong variational technique selected for the computation of the optimal control is presented. The computational results of the optimization problem are presented latter in the chapter.

In chapter four, the classical orbital dynamics, composed of six differential equations, is introduced without any derivation [6]. A description of the coordinate systems and orbital elements are also provided to facilitate the understanding of the readers. An optimal control problem is formulated for the synchronous satellite. The Conjugate

Gradient Descent method chosen to solve the two-point boundary value problem is described. A brief review of the Sequential Control Scheme [16] is provided and comparisons are made between the Optimal and Sequential Control Schemes. The results and guidelines for the design of the practical implementation of the orbital control systems are presented at the end of the chapter.

Finally, in the last chapter, discussions, conclusions, comments, and suggestions for further research are presented.

Chapter II

A REVIEW OF PONTRYAGIN'S MAXIMUN PRINCIPLE

In this chapter, the Pontryagin's Maximun Principle for the processes with delay [1:213-217] will be stated without proof. The principle can be easily extended to the process without delay [1:17-21] and will not be presented here.

2.1 STATEMENT OF PROBLEM, BASIC ASSUMPTIONS AND DEFINITIONS

Let the control system be described by the following delay-differential equation on the fixed time interval $(t_0, t_f]$

$$\dot{x}(t) = \sum_{j=0}^{s} f^{j}(t-h_j, x(t-h_j), u(t-h_j)), \tag{2.1}$$

where

$x = (x_1, x_2, \ldots, x_m)^T \varepsilon R^m$ is the state vector,

$u = (u_1, u_2, \ldots, u_r)^T \varepsilon R^r$ is the control vector,

$f^j = (f_1^j, f_2^j, \ldots, f_m^j)^T \varepsilon R^m$ $(j=0,\ldots,s)$ is an m-vector valued function, and h_j are the time-delays, ordered so that

$$0 = h_0 < h_1 < h_2 < \ldots < h_s < t_f \; ; \; s < \infty.$$

The initial function for (2.1) is

$$x(t) = p(t),\ t\varepsilon[-h_s,0], \tag{2.2}$$

where $p = (p_1, p_2, \ldots, p_m)^T$ is a given, absolutely continuous function on $[-h_s,0]$ with values in R^m.

Let $\mathcal{U}_a$ be the class of all admissible controls defined by $\mathcal{U}_a = \{ u :$ u is a function from $[-h_s, t_f]$ into U, piecewise continuous on $[0,t_f]$ and with $u(t) = q(t)$ on $[-h_s, 0]$ $\}$, where q is a given piecewise continuous function on $[-h_s, 0)$ with values in U, and U is a compact and convex subset of R^r.

The function f^j_i and $\frac{\partial f^j_i}{\partial x_k}$ ($j = 0,\ldots,s$; $i = 0,\ldots,m$; $k = 1,\ldots,m$) are piecewise continuous on $[t_0, t_f]$ for all $(x,u)\varepsilon R^m \times U$ and continuous on $R^m \times U$ for each $t \varepsilon [t_0, t_f]$.

The optimal control problem to be solved is to find a control $u \varepsilon \mathcal{U}_a$ that will minimize the cost functional

$$J(u) = \int_{t_0}^{t_f} \sum_{j=0}^{s} f^j_o(t-h_j, x(u)(t-h_j), u(t-h_j))\ dt \tag{2.3}$$

subject to the dynamics contraint (2.1) with the initial condition (2.2).

Definition 2.1

For each $u \in \mathcal{U}_a$, the absolutely continuously function

$$\psi(u) = (\psi_0(u), \psi_1(u), \ldots, \psi_m(u))^T : [t_0, t_f] \to R^{m+1}$$

is the solution of

$$\dot{\psi}(t) = -\sum_{k=0}^{m} \sum_{j=0}^{s} \frac{\partial f_k^j(t, x(t-h_j), u(t))}{\partial x_i^j} \psi_k(t+h_j) \qquad (2.4)$$

$$\psi_0(t) = -1 \qquad (i = 1, \ldots, m),$$

with the final condition

$$\psi_i(t) = 0 \quad \text{for all } t \geq t_f \quad (i=1,\ldots,m)$$

and,

$$\psi_0(t) = 0 \quad \text{for all } t > t_f, \qquad (2.5)$$

where x_i^j denotes $x_i(t-h_j)$.

Definition 2.2

The hamiltonian function $H : [t_0, t_f] \times R^m \times U \times R^{(m+1)(s+1)} \to R^1$ is defined as

$$H(t, x, v, \Psi) = \sum_{k=0}^{m} \sum_{j=0}^{s} f_k^j(t, x(t-h_j), v)\, \psi_k(t+h_j), \qquad (2.6)$$

where $\Psi(t)$ denotes $(\psi(t+h_0), \psi(t+h_1), \ldots, \psi(t+h_s))$.

Definition 2.3

Let S_1 be the smooth k-dimensional manifold in the space R^m where $0 \leq k \leq m-1$, and L is the (k+1)-dimensional manifold which is the direct product of S_1 with the x_0 axis, where

$$x_0(t) = \int_{t_0}^{t_f} \sum_{j=0}^{s} f_0(t-h_j, x(u)(t-h_j), u(t-h_j))\, dt.$$

With the definitions defined above, one is ready to state the necessary conditions for optimality for the given problem in the form of a Maximum Principle.

2.2 PONTRYAGIN'S MAXIMUM PRINCIPLE

Let $u(t)$, $t_0 \leq t \leq t_f$ be an admissible control such that the corresponding trajectory $x(t)$, $t_0-h_j \leq t \leq t_f$, of system (2.1) with initial function $p(t)$, $t_0-h_j \leq t \leq t_f$, passes through some point of the manifold L at some time $t_f > t_0$. In order that $u(t)$ and $x(t)$ be optimal, it is necessary that there exist a nonzero vector function $\psi(t) = (\psi_0(t), \psi_1(t), \ldots, \psi_m(t))$, corresponding to the function $u(t)$ and $x(t)$, such that :

1) The maximum condition

$$H(\Psi(t), x(t), x(t-h_j), u(t)) = M(\Psi(t), x(t), x(t-h_j)) \qquad (2.7)$$

holds for all t, $t_0 \leq t \leq t_f$; where

$$M(\psi(t), x(t), x(t-h_j)) = \max_{u \in \mathcal{U}_a} H(\Psi(t), x(t), x(t-h_j); u(t)).$$

2) At the terminal time t_f, the relations

$$\psi_0(t_f) \leq 0, \quad M(\ \Psi(t_f), x(t_f), x(t_f - h_j)\ ) = 0 \tag{2.8}$$

are satified.

3) The vector $(\ \psi_1(t_f), \ldots, \psi_m(t_f)\ )$ is orthogonal to the tangent plane of the manifold S_1 at point $x(t_f) = (\ x_1(t_f), \ldots, x_m(t_f)\ )$.

2.3 COMPUTATIONAL METHODS IN OPTIMAL CONTROL

To solve the optimization problem described in equations (2.1)-(2.3), it is required to introduce another system of differential equations called the adjoint (costate) equations (2.4), and the Hamiltonian function (2.6).

The computation of optimal control as usual involves solving the dynamics (state) equations forward in time, and the adjoint equations backward in time for an initial guess of control, $u^0 \varepsilon \mathcal{U}_a$. With the states known, the cost functional $J(u^0)$ is computed. A sequence of control policies $u^1, u^2, \ldots, u^i, u^{i+1}, \ldots; u^i \varepsilon \mathcal{U}_a$ is constructed in a suitable way such that $J(u^{i+1}) < J(u^i)$. It should be noted that the computation of every new control involves solving the state and adjoint equations, and computing the hamiltonian function over the entire interval $t \varepsilon (t_0, t_f]$. The control which maximizes the hamiltonian is used to compute the cost functional. The previous control will be replaced only if the newly computed control has improved the cost functional. Since the optimization procedures require a considerable amount of computation, an algorithm which computes the new control efficiently is desirable.

In this thesis, two different computation algorithms are considered. An algorithm based on strong variational technique developed by Murray and Teo [5] is used to solve the radar scanning problem. For the satellite orbit control problem, the Conjugate Gradient Descent method is adopted. The detailed explanations of each algorithm will be presented in the latter chapters.

Chapter III

OPTIMAL SCANNING POLICY FOR PHASED ARRAY RADAR

In this chapter, a radar detection problem is considered. The aim is to optimize the search scanning function of radars with electronically steerable beams. The design method requires a model of target dynamics and radar detection statistics to act as a constraint in the minimization of a cost criterion. The objective is to determine a beam steering function which results in the best detection statistics possible in the time available. An optimization program was developed to solve this problem [11].

3.1 INTRODUCTION

With some notable exceptions, the vast majority of radars have to cover an area in searching and/or tracking, rather than always being pointed in the same direction. This implies that the antenna will have to move, although some limited beam movement can be produced by multiple feeds or by a moving feed antenna. As long as antenna motion is involved in moving the beam, limitations caused by inertia will always exist. However, such limitation can be overcame by employing phased array radar.

The phased array is a directive antenna made up of individual radiating antennae, or elements, which generate a radiation pattern whose shape and direction is determined by the relative phases and amplitudes of the currents at the individual elements. By properly varying the relative phases, it is possible to steer the direction of the radiation. The radiating elements might be dipoles, open-ended waveguides, slot cut in waveguide, or any type of antenna. The inherent flexibility offered by the phased array radar in steering the beam by means of electronics control in what has made it of interest for radar. It has been considered in those radar applications where it is necessary to shift the beam rapidly from one position in space to another, or when it is required to obtain information about many targets at a flexible, rapid data rate.

With the advent of phased array radars, a new dimension has been added to radar operations: the ability to perform fast and accurate beam control. In an intrusion detection system, beam control serves the functions of search scanning and tracking. Each function may control the radar in an arbitrary time-shared manner. The performance of the system depends on its ability to estimate the movement of targets, both detected and as yet undetected. This work is concerned with as yet undetected targets; that is, search scanning.

Optimization of radar search scanning requires, firstly, a system model based on the statistics of undetected targets in the area of coverage and the radar detection process, and, secondly, a measure of system performance. The model reflects the certainty that regions of the geographical area of coverage have not been entered by undetected targets. The variable, certainty, is then a function of the statistical properties of the particular detection process used by the radar and of the movements of intruders. The model is generally non-linear, but will be assumed linear here as a first approximation. The solution of the problem results in a set of instructions indicating the sequence of areas of a surface to be illuminated by a radar with the purpose of finding undetected targets in an optimal way. The set of instructions are determined a priori and are not effected by target detections during their execution, as may be the case when tracking.

In the following scetions the system model will be presented, and an optimal control problem is formulated. This problem is then solved by an algorithm based on a strong variational technique [5].

3.2 BASIC CONCEPTS OF PHASED ARRAY RADAR

The concept of directive radiation from fixed (nonsteered) phased array antennae was conceived during World War I. The first use of the phased array antenna in commercial broadcasting transmission was in the early 30's. The first large steered directive array for the reception of transatlantic short-wave communication was installed in the late 30's by the Bell Laboratories. A major advance in phased array technology was made in the early 50's with the replacement of mechanically actuated phase shifters by electronics phase shifters.

Basically, an array antenna consists of a number of individual radiating elements suitably spaced with respect to one another. The relative amplitude and phase of the signals applied to each of the elements are controlled to obtain the desired radiation pattern from the combined action of all the elements. Two common geometrical forms of array antennae of interest in radar are the linear array and the planar array. A linear array consists of elements arranged in a straight line in one dimension. A planar array is a two-dimensional configuration of elements arranged to lie in a plane.

As a simple illustration, only linear array made up of N elements equally spaced a distance d apart is shown in Figure 1. For convenience, the array is shown as a receiving antenna, but because of the reciprocity principle, the results obtained apply equally well to a transmitting antenna. The outputs of all the elements are summed via lines of equal length to give a sum output voltage E_a. Element 1 will be taken as the reference signal with zero phase. The difference in the phase of the signals in adjacent elements is $\psi = \{2\pi d \sin\theta\}/\lambda$, where θ and λ are the direction and wavelength of the incoming radiation respectively.

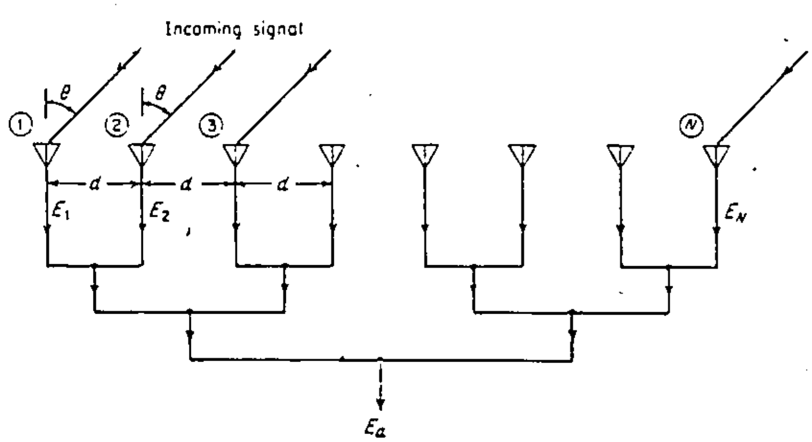


Figure 1: N-Element Linear Array

The beam of an array antenna may be steered rapidly in space without moving large mechanical masses by properly varying the phase of the signals applied to each element. An array in which the relative phase shift between elements is controlled by the electronic devices is called an electronically scanned array. A schematic showing steering of an antenna beam with variable phase shifters is illustrated in Figure 2 [12].

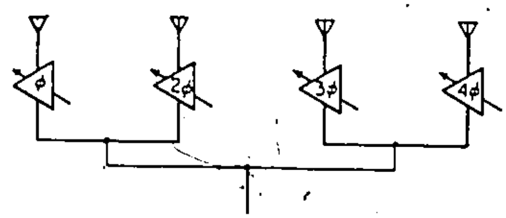


Figure 2: Steering of an Antenna Beam With Variable Phase Shifters

3.3 SYSTEM MODEL

Assume the area to be protected is a plain surveyed by a downward looking radar and that is divided into m regions. Each region, i, has an associated certainty, $x_i(t)$ $(i = 1,\dots,m)$, that it has not been penetrated by a target which has yet to be detected. Here the term certainty is used to mean the probability that the event has not occured by some instant of time. The measure of certainty is clearly a function of time. A measure of unity denotes absolute confidence that no undetected target is present. A measure of zero indicates a complete lack of knowledge of the region. A variable $u_i(t)$ associated with each region takes one of two possible values, unity, indicating the region is being scanned, and zero, indicating the region is not being processed by the radar.

The model must describe the certainty of a region when it is being scanned by the radar, and also when it is not being scanned.

In order to develop the model of the certainty of a region which is being scanned, a geographical region as shown in Figure 3 is divided into a number of cells, in this case 100, which are illuminated at a frequency of 1666 Hz. In this case, it is assumed that an illumination area is processed as a single cell. A look at each cell contains two noncoherently integrated pulses to achieve a signal-to-noise ratio of 3 dB in each burst with a probability of false alarm of 10^{-6} per look. This detection analysis is based on the valid assumption that the target does not move from one region to another during the entire process that is the time spent scanning a region is much less than the time required by the targets to move any significant distance. The probability of detecting a target in a cell with one look is 0.02 [13].

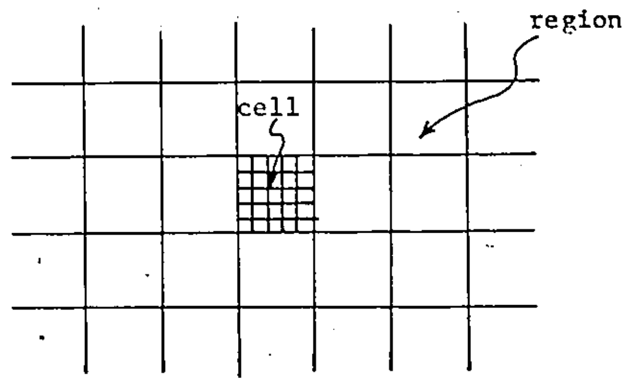


Figure 3: Cells Arrangement in a Region

One is concerned with the detection of the first target to penetrate, say, the i th region. Let the event that this target is in cell j (which is contained in the i th region) at time of scanning be denoted by k_j. It is assumed that the target is equally likely to be found in any one of the N cells of the i th region, then the probability that the target is in the j th cell of the i th region is

$$P_i(k_j) = 1/N. \tag{3.1}$$

The events k_j are mutually exclusive since the single penetrating target remains in one cell. Let the event of a detection being declared in cell j, given that the target is in that cell be denoted by D_j. Hence $P_i(D_j | k_j)$ is the probability of detection. The total probability of detecting the target in region i is then

$$x'_i(N\Delta t) = \sum_{j=1}^{N} P_i(D_j \mid k_j)\; P_i(k_j) \tag{3.2}$$

where Δt is the look period and N is the number of looks required to cover the region once. In this example,

$$P_i(D_j \mid k_j) = 0.02$$

and

$$P_i(k_j) = 1/N$$

It takes 0.06 seconds to scan the entire region once, and the certainty achieved is 0.02.

The probability of detection with repeated scanning is calculated according to

$$1 - x'_i(k(N\Delta t)) = (1 - x'_i(N\Delta t))^k \tag{3.3}$$

where k is the number of scans and successive scans are treated as independent events. The certainty function described by (3.3) can be approximated by an exponential function which is the solution of a first order differential equation

$$\hat{\tau}_i \dot{x}_i(t) = 1 - x_i(t) \tag{3.4}$$

The time constant $\hat{\tau}_i$ of the certainty dynamics is determined by minimizing the mean squared error

$$e^2(\hat{\tau}_i) = \int_0^\infty \left| x'_i(t) - x_i(t) \right|^2 dt, \quad i = 1, \ldots, m \tag{3.5}$$

In order to describe the certainty of a region when it is not being scanned, it is necessary to model the motion of the target statistically. The heading of a target is assumed to have the following probability density function

$$f_{\Theta}(\Theta) = \begin{cases} 1/\pi & , -\pi/2 \leq \Theta \leq \pi/2 \\ 0 & , \text{elsewhere} \end{cases} \tag{3.6}$$

The probability density function of penetration time T of the next (immediate) region with the specified distribution of target headings is

$$f_T(t) = \begin{cases} \dfrac{2}{\pi} \dfrac{T_o}{t^2\sqrt{1 - \left(\dfrac{T_o}{t}\right)^2}} & t > T_o \\ 0 & t \leq T_o \end{cases} \tag{3.7}$$

where T_0 is the time required to travel from one region to the next in the most direct path, i.e. $\Theta = 0°$. It should be noted that different partitioning of regions will result in different crossing times. It is assumed that the target speed is maintained constant and the target maintains a constant heading while crossing a region.

Consider a target crossing an area divided into m regions as shown in Figure 4. The distribution of the target heading and penetration time from one region to another are as given in (3.6) and (3.7). A typical target movement is illustrated in Figure 4 in which the random crossing time from one region to another is indicated at the boundary of each region.

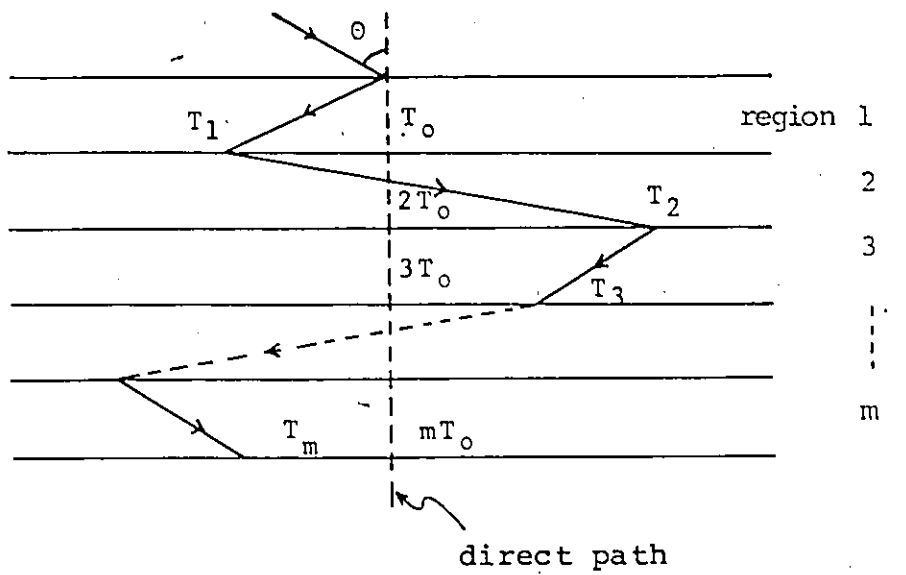


Figure 4: Typical Target Crossing of Regions

Therefore the total time taken by the target to transverse the first j-regions is given by $w_j = \sum_{i=1}^{j} T_i$. Since the random variables T_i's $(i=1,..,m)$ are independent, the probability density function of the random variable w_j is given by the following convolution

$$p_j\,(w_j = t) = (f_T(\cdot) * \ldots * f_T(\cdot))\,(t). \tag{3.8}$$

The distribution of penetration time of the next region given that of the previous regions is

$$p_{j+1}(t) = (f_T * p_j)\,(t). \tag{3.9}$$

The function $g_T(t)$ given by

$$g_T(t) = f_T(t+T_o), \quad t>0 \tag{3.10}$$

can be closely approximated by $g(t)$ which has the following form

$$g(t) = \frac{1}{\tau_i}\exp(-t/\tau_i), \quad t>0 \tag{3.11}$$

where τ_i is chosen in such a way that the mean square error

$$e^2(\tau_i) = \int_0^\infty |g(t) - g_T(t)|^2 \, dt \tag{3.12}$$

is minimum.

Let us consider the probability density function of the random variable $w_{i+1}=w_i+T_{i+1}$ where w_i has the density $p_i(\cdot)$ and T_{i+1} has the density $g_T(t)$. Using its approximate counterpart $g(t)$ one has

$$p_{i+1}(t) = \int_0^t g(t-\theta)\, p_i(\theta)\, d\theta. \tag{3.13}$$

Differentiating once, one obtains

$$\dot{p}_{i+1}(t) = g(0)\, p_i(t) + \int_0^t \frac{\partial}{\partial t} g(t-\theta)\, p_i(\theta)\, d\theta, \tag{3.14}$$

since $\frac{d}{dt} g(t) = -\frac{1}{\tau} g(t)$, thus

$$\dot{p}_{i+1}(t) = \frac{1}{\tau_i} p_i(t) - \frac{1}{\tau_i} p_{i+1}(t). \tag{3.15}$$

Define the probability of the i th region being penetrated by time t as

$$z_i(t) = \int_0^t p_i(\theta)\, d\theta \tag{3.16}$$

and the certainty function of undetected target as

$$x_i(t) = 1 - z_i(t) \tag{3.17}$$

with initial condition

$$p_i(t) = 0 \text{ for } t < 0. \tag{3.18}$$

Under condition (3.18), integration of (3.15) yields

$$\tau_i \dot{z}_{i+1}(t) = z_i(t) - z_{i+1}(t)$$

or,

$$\tau_i \dot{x}_{i+1}(t) = x_i(t) - x_{i+1}(t). \tag{3.19}$$

This result can be easily extended to a two dimensional model by considering the probability that a target might enter the following adjacent regions from its present region (see Figure 5), i.e.

$$\tau_i \dot{x}_{i+1}(t) = -x_{i+1}(t) + \sum_{j=1}^{m} P_{i+1,j}\, x_j(t-T_{ij}) \qquad 0 \leq i \leq m-1. \tag{3.20}$$

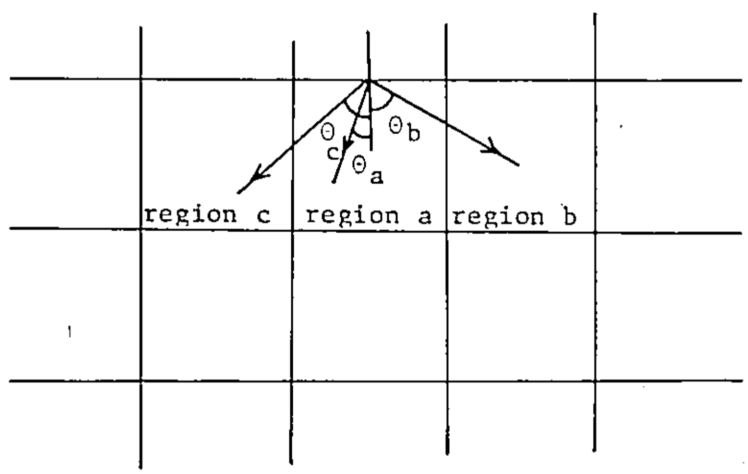


Figure 5: Penetration of Regions in Two-Dimensional Model

To summarise the above, a model with time delayed arguments was formulated :

$$\dot{x}_i(t)=f_i(x(t),x(t-T_{i1}),\ldots,x(t-T_{im}),u(t),t) \quad i=1,\ldots m \quad (3.21)$$

where $x(t)$ and $u(t)$ are m vectors with elements $x_i(t)$ and $u_i(t)$, $1 \leqslant i \leqslant m$ respectively. T_{ij} is the time taken by the target to travel from the i th region to the j th region.

For a two-dimensional geographical region, equations (3.4) and (3.20) can take the form of a system of first order linear differential equations given by

$$\tau_i \dot{x}_i(t) = - x_i(t) + \sum_{\substack{j=1 \\ j\neq i}}^{m} P_{ij}x_j(t - T_{ij}) , \quad i = 1,\ldots,m, \qquad (3.22)$$

when the region i is not being scanned, i.e. $u_i(t)=0$,

and,

$$\hat{\tau}_i \dot{x}_i(t) = u_i(t) - x_i(t) , \quad i = 1,\ldots,m, \qquad (3.23)$$

when the region i is being scanned, i.e. $u_i(t)=1$,

where τ_i and $\hat{\tau}_i$ $(i = 1, \ldots, m)$ are time constants associated with the i th region, and P_{ij} $(i, j = 1, \ldots, m)$ is the probability of a target entering region j from region i $(P_{ii} = 0)$. The parameters τ_i, $\hat{\tau}_i$ and T_{ij} are chosen in such a way that the model response closely approximates the actual statistical data available for the given region based on target dynamics and radar detection statistics. It is assumed that the time constant $\hat{\tau}_i$ associated with the increase of certainty of a region when being scanned is much smaller than τ_i. Indeed the scanning time required to raise a region's certainty is much shorter than the time for any appreciable drop of certainty to occur because of target movement.

For convenience of formulation of the optimization problem, equations (3.22) and (3.23) are combined to form the following system of equations

$$\dot{x}_i = -\beta_i u_i (x_i - 1) - \alpha_i (1-u_i)[x_i - \sum_{\substack{j=1 \\ j\neq i}}^{m} p_{ij} x_i (t-T_{ij})], \quad i = 1, \ldots, m \qquad (3.24)$$

where,

β_i ($i = 1, \ldots, m$) denotes the reciprocal of the time constant, τ_i when contol u_i is applied to the i th region,

and

α_i ($i = 1, \ldots, m$) denotes the reciprocal of the time constant, $\hat{\tau}_i$ when no control is applied to the i th region.

3.4 FORMULATION OF OPTIMAL CONTROL PROBLEM

The purpose of computing an optimal scanning policy for a phased array radar is to minimize the uncertainties about the presence or absence of an enemy target with a minimum scanning effort. Further, it is desirable to maintain a high certainty in each region at the end of each scanning period.

This objective can be achieved by minimizing the following cost functional

$$J(u) = \sum_{i=1}^{m} \lambda_i (1-x_i(t_f))^2 + \int_{t_0}^{t_f} \{\sum_{i=1}^{m} \sigma_i u_i(t) + \gamma_i (1-x_i(t))^2\} dt \tag{3.25}$$

where,

λ_i is the weighting factor of the terminal uncertainty $(1-x_i(t_f))$ of the i th region,

σ_i is the weighting factor of the control in the i th region, and

γ_i is the weighting factor of the uncertainty $(1-x_i(t))$ in the i th region in the interval $(t_o, t_f]$.

The values of λ_i and γ_i indicate the relative importance of the regions. For example, regions with high probability of target entry such as the fence regions, should be assigned larger weights. σ_i indicates the cost of scanning a region. For example, a large σ_i indicates a reluctance to scan region i.

The optimal control problem to be solved is to minimize the cost functional (3.25) subject to the system dynamics (3.24). By doing this, an optimal scanning policy for a phased array radar resulting in least deviation from the desired certainty (unity) can be obtained.

In this work, it is assumed that no two regions will be scanned at the same time, and the radar is fully occupied with search scanning during the interval $t \varepsilon (t_o , t_f]$. Due to these constraints, the admissible class of controls consists of m-vector valued functions $u(t) = (u_1(t), \ldots, u_m(t))'$ such that $\sum_{i=1}^{m} u_i(t) = 1$ for all $t \varepsilon (t_o, t_f]$. For convenience this class of controls is denoted by the set $\mathcal{U}_a$.

This problem is solved by use of Optimal Control Theory as presented in Chapter 2. An algorithm for the numerical computation of the optimal control is presented in the following section.

3.5 METHOD OF SOLUTION

The main drawback of the method of solution as described in Section 2.3 is that it requires exact maximization of the Hamiltonian for all $t \varepsilon (t_o, t_f]$. However, the algorithm developed by Murray and Teo [5] requires only approximate maximization of the hamiltonian for a suitably selected set of points from the interval $(t_o, t_f]$. Indeed consider the function H_{max} defined as

$$H_{max}(t, x(u)(t), \Psi(u)(t)) = \max H(t, x(u)(t), v, \Psi(u)(t)),$$

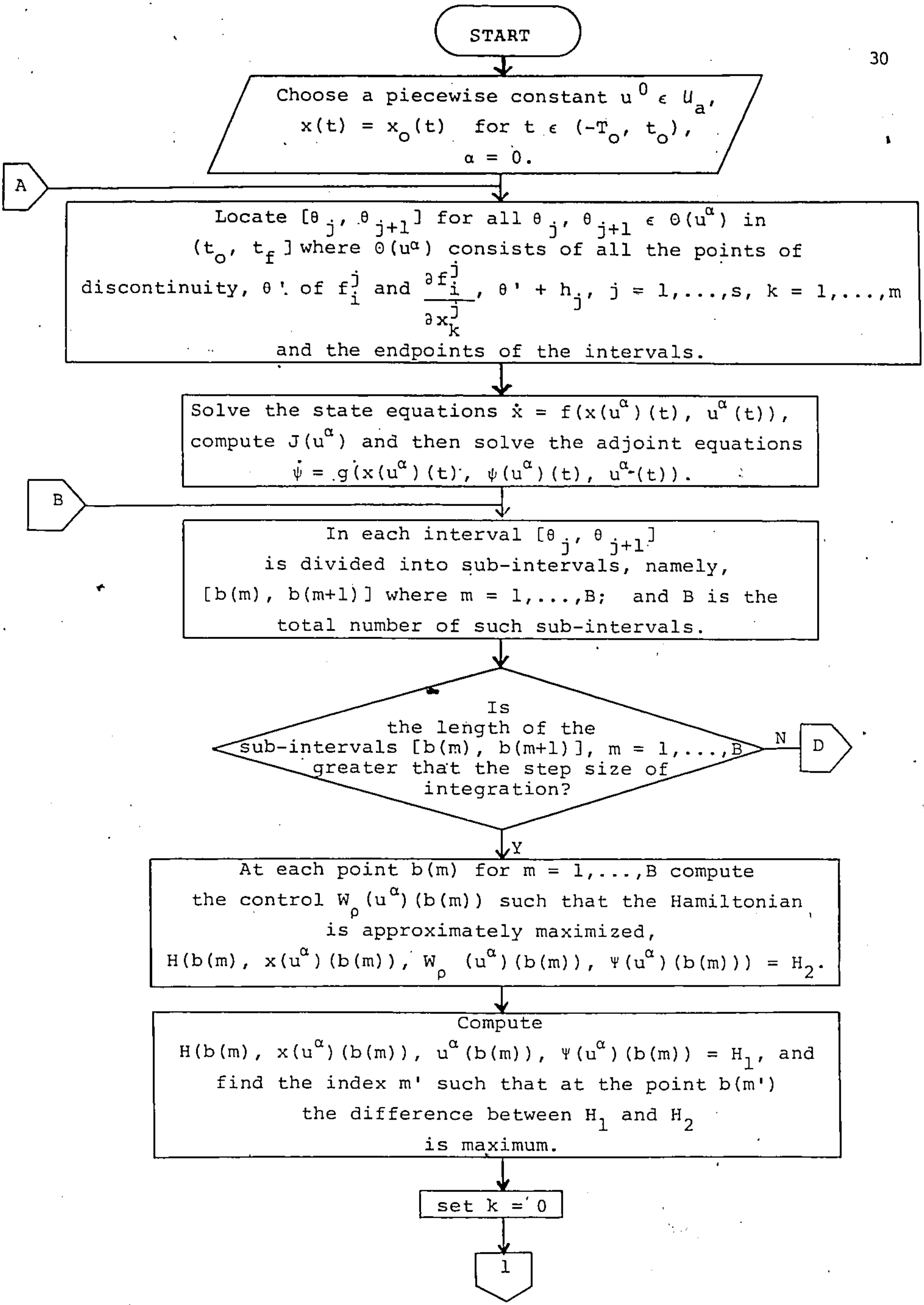
for $t \varepsilon (t_o, t_f]$, $x \varepsilon R^m$, and $\psi \varepsilon R^m$. Let u be any admissible control and suppose $x(u)$ and $\psi(u)$ are the corresponding responses of the state equation (2.1) and the adjoint equation (2.4) respectively. Using a function maximization algorithm for a fixed but arbitrary $t \varepsilon (t_o, t_f]$, an approximate maximizing control $w_\rho \in \mathcal{U}_a$, which may depend on u, can be chosen at the ρ th stage of iteration such that

$$\left| H_{max}(t, x(u)(t), \Psi(u)(t)) - H(t, x(u)(t), w_\rho(u)(t), \Psi(u)) \right| < k\Gamma^\rho$$

for a fixed integer k and $0 < \Gamma < 1$.

3.5.1 The (Murray and Teo's) Algorithm

The strategy of the algorithm used to perturb the control is based on the idea of locating a point t_1, in the time interval $t \in (t_o, t_f]$ such that there is a maximum change in the value of the hamiltonian function corresponding to the previous control $u^i(t_1)$, and the most recently computed control $w_\rho(u^i)(t_1)$. The point where the largest change in the hamiltonian function occurs is taken as the focal point around which the change of control u^i to $w_\rho(u^i)$ takes place. However, if the perturbation interval around the most sensitive point is not properly chosen, then the resulting trajectories of the states may be quite different from those corresponding to the extremal control. So, one may perturb the control around the sensitive point with a larger interval initially; and if there is no cost improvement, then one may reduce the interval by half and so on until there is an improvement in the cost functional. The flow chart describing the algorithm is presented in Figure 6 [5,14]. The notation [a] means the greatest integer less than or equal to a (see Figure 6).


START
Choose a piecewise constant $u^0 \epsilon U_a$, $x(t) = x_o(t)$ for $t \epsilon (-T_o, t_o)$, $\alpha = 0$.
A
Locate $[\theta_j, \theta_{j+1}]$ for all $\theta_j, \theta_{j+1} \epsilon \Theta(u^\alpha)$ in $(t_o, t_f]$ where $\Theta(u^\alpha)$ consists of all the points of discontinuity, θ' of f_i^j and $\frac{\partial f_i^j}{\partial x_k^j}$, $\theta' + h_j$, $j = 1,\ldots,s$, $k = 1,\ldots,m$ and the endpoints of the intervals.
Solve the state equations $\dot{x} = f(x(u^\alpha)(t), u^\alpha(t))$, compute $J(u^\alpha)$ and then solve the adjoint equations $\dot{\psi} = g(x(u^\alpha)(t), \psi(u^\alpha)(t), u^\alpha(t))$.
B
In each interval $[\theta_j, \theta_{j+1}]$ is divided into sub-intervals, namely, $[b(m), b(m+1)]$ where $m = 1,\ldots,B$; and B is the total number of such sub-intervals.
Is the length of the sub-intervals $[b(m), b(m+1)]$, $m = 1,\ldots,B$ greater that the step size of integration?
N
D
Y
At each point $b(m)$ for $m = 1,\ldots,B$ compute the control $W_\rho(u^\alpha)(b(m))$ such that the Hamiltonian is approximately maximized, $H(b(m), x(u^\alpha)(b(m)), W_\rho(u^\alpha)(b(m)), \Psi(u^\alpha)(b(m))) = H_2$.
Compute $H(b(m), x(u^\alpha)(b(m)), u^\alpha(b(m)), \Psi(u^\alpha)(b(m))) = H_1$, and find the index m' such that at the point b(m') the difference between H_1 and H_2 is maximum.
set k = 0
1

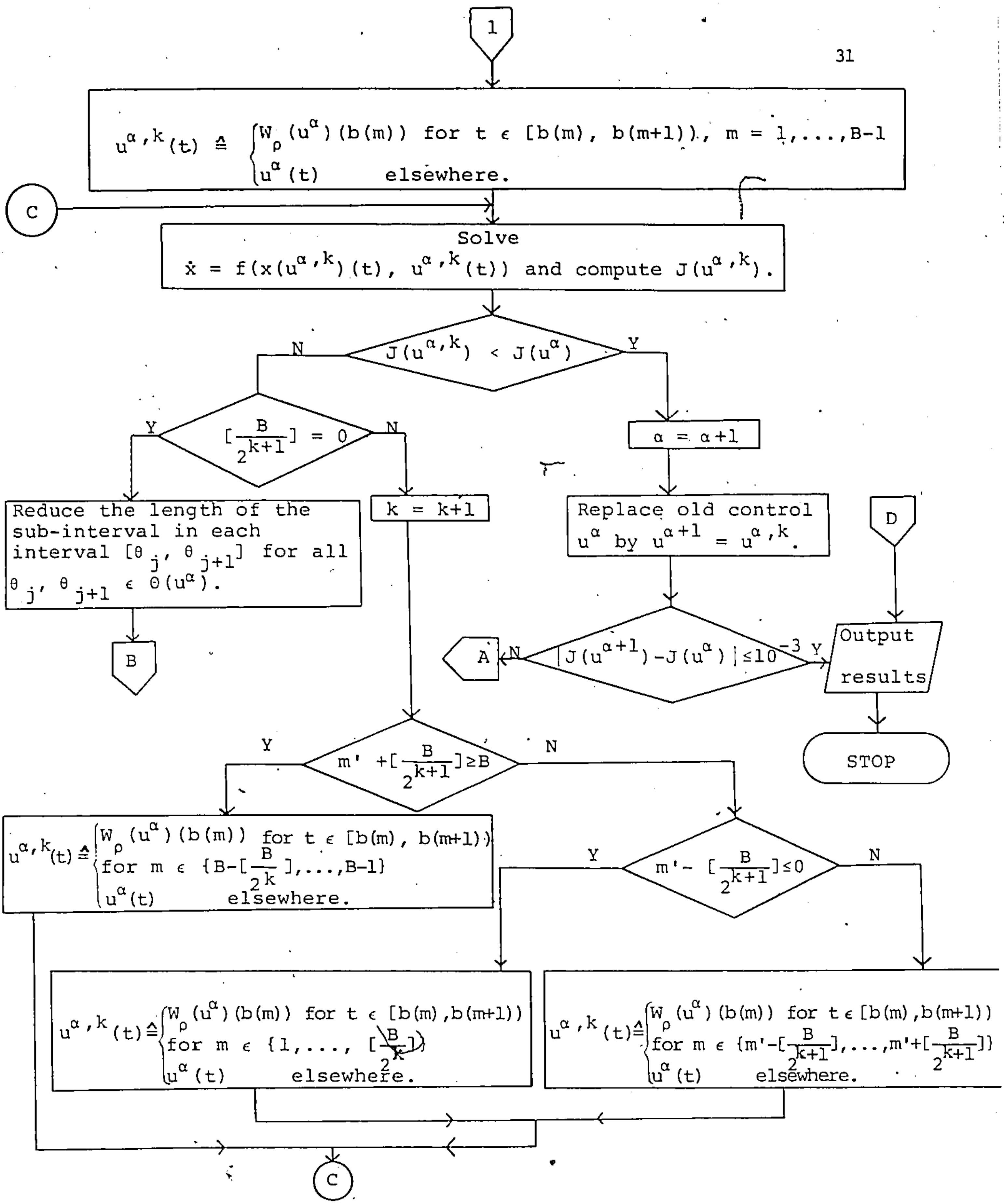


Figure 6: Flow Chart for Murray and Teo's Algorithm

3.6 SIMULATION OF THE TWO-DIMENSIONAL NINE-REGION MODEL

As an illustration, a Two-Dimensional Nine-Region model as shown in Figure 7 is considered. The dynamics equations of the model are given by equation (3.26). This equation follows from the general equation (3.24) by appropriate choice of the parameters α_i, β_i, P_{ij} and T_{ij} $(i,j = 1,\ldots,m)$ which depend on the topology of the cell arrangements and target dynamics.

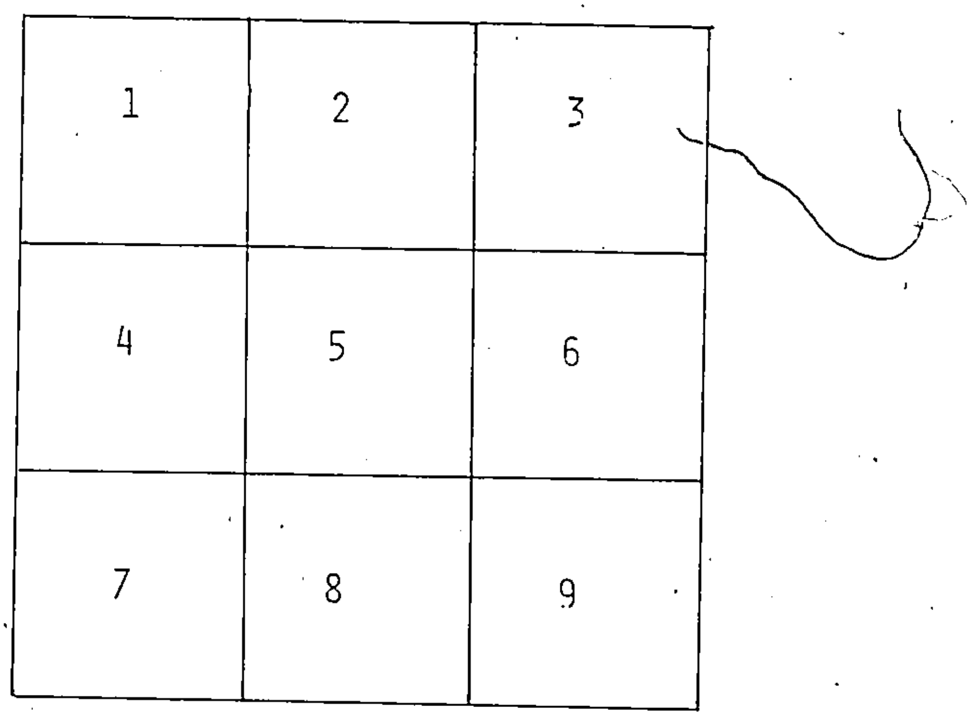


Figure 7: Geometric Representation of the Two-dimensional Nine-Region Model

$$\dot{x}_1 = \beta_1 u_1 (1-x_1) - \alpha_1 (1-u_1) x_1$$

$$\dot{x}_2 = \beta_2 u_2 (1-x_2) - \alpha_2 (1-u_2) x_2$$

$$\dot{x}_3 = \beta_3 u_3 (1-x_3) - \alpha_3 (1-u_3) x_3$$

$$\dot{x}_4 = -\beta_4 u_4 (x_4-1) - \alpha_4 (1-u_4)[x_4 - P_4(a)x_1(t-T_o)-0.5P_4(\bar{a})x_2(t-T_o)]$$

$$\dot{x}_5 = -\beta_5 u_5(x_5-1)-\alpha_5(1-u_5)[x_5-P_5(a)x_2(t-T_o)$$
$$-0.5P_5(\bar{a})x_1(t-T_o)-0.5P_5(\bar{a})x_3(t-T_o)]$$

$$\dot{x}_6 = -\beta_6 u_6(x_6-1)-\alpha_6(1-u_6)[x_6-P_6(a)x_3(t-T_o)-0.5P_6(\bar{a})x_2(t-T_o)]$$

$$\dot{x}_7 = -\beta_7 u_7(x_7-1)-\alpha_7(1-u_7)[x_7-P_7(a)x_4(t-T_o)-0.5P_7(\bar{a})x_5(t-T_o)]$$

$$\dot{x}_8 = -\beta_8 u_8(x_8-1)-\alpha_8(1-u_8)[x_8-P_8(a)x_5(t-T_o)$$
$$-0.5P_8(\bar{a})x_4(t-T_o)-0.5P_8(\bar{a})x_6(t-T_o)]$$

$$\dot{x}_9 = -\beta_9 u_9(x_9-1)-\alpha_9(1-u_9)[x_9-P_9(a)x_6(t-T_o)-0.5P_9(\bar{a})x_5(t-T_o)] \quad (3.26)$$

As an example, the initial conditions and parameter values are assumed to be

$$x_i(t) = x_{io} \qquad (i = 1,\ldots,9) \quad \text{for } t \in [-T_o, t_o]$$

and,

$P_4(a) = P_{14}$, $\quad P_{15} = 0.5(1-P_{14})$, $\quad P_{ij} = 0 \quad (j \neq 4,5)$;

$P_5(a) = P_{25}$, $\quad P_{24} = P_{26} = 0.5(1-P_{25})$, $\quad P_{2j} = 0 \quad (j \neq 4,5,6)$;

$P_6(a) = P_{36}$, $\quad P_{35} = 0.5(1-P_{36})$, $\quad P_{3j} = 0 \quad (j \neq 5,6)$;

$P_7(a) = P_{47}$, $\quad P_{48} = 0.5(1-P_{47})$, $\quad P_{4j} = 0 \quad (j \neq 7,8)$;

$P_8(a) = P_{58}$, $\quad P_{57} = P_{59} = 0.5(1-P_{58})$, $\quad P_{5j} = 0 \quad (j \neq 7,8,9)$;

$P_9(a) = P_{69}$, $\quad P_{68} = 0.5(1-P_{69})$, $\quad P_{6j} = 0 \quad (j \neq 8,9)$;

$P_i(\bar{a}) = 1-P_i(a) \quad (i = 4,\ldots,9)$, $\quad P_{7j} = P_{8j} = P_{9j} = 0 \quad (j = 1,\ldots,9)$, and,

$$T_{14} = T_o, \quad T_{1j} = 0 \quad (j \neq 4);$$
$$T_{25} = T_o, \quad T_{2j} = 0 \quad (j \neq 5);$$
$$T_{36} = T_o, \quad T_{3j} = 0 \quad (j \neq 6);$$
$$T_{47} = T_o, \quad T_{4j} = 0 \quad (j \neq 7);$$
$$T_{58} = T_o, \quad T_{5j} = 0 \quad (j \neq 8);$$
$$T_{69} = T_o, \quad T_{6j} = 0 \quad (j \neq 9);$$
$$T_{7j} = T_{8j} = T_{9j} = 0 \quad (j = 1,\ldots,9).$$

It is assumed that target can enter the protected area only through the first three regions (fence regons) and that the time parameters are the same for all regions.

The cost functional to be minimized is given by equation (3.25) with m = 9.

The adjoint (costate) differential equations and the hamiltonian function corresponding to the system equations (3.26) are previously desrcibed in equations (2.4)-(2.5) and (2.6) respectively. The explicit expressions for the adjoint equations and the hamiltonian function are given in Appendix A.

The optimal control problem was simulated in an AMDAHL 470/V7A computer at the University of Ottawa. The following parameters were used in the simulations:

Case 1

$$\beta_i = 2 \text{ sec}^{-1}$$
$$\alpha_i = 0.05 \text{ sec}^{-1} \quad (i = 1,\ldots,9)$$

Case 2

$$\beta_i = 0.5 \text{ sec}^{-1}$$
$$\alpha_i = 0.05 \text{ sec}^{-1} \quad (i = 1,\ldots,9).$$

In both cases, the time delay T_0 is taken to be 10 seconds, and the weighting factors of the cost functional are taken as

$$\sigma_1 = \sigma_2 = \sigma_3 = \ldots = \sigma_9 = 1,$$
$$\gamma_1 = \gamma_2 = \gamma_3 = \ldots = \gamma_9 = 10,$$
$$\lambda_1 = \lambda_2 = \lambda_3 = \ldots = \lambda_9 = 20.$$

The initial conditions are

$x_1(0) = x_2(0) = \ldots = x_9(0) = 0.5$, and

the probabilities of entering one region from another are

$P_4(a) = P_6(a) = P_7(a) = P_9(a) = 0.9$, and

$P_5(a) = P_8(a) = 0.8.$

3.7 COMPUTATIONAL RESULTS AND PERFORMANCE ANALYSIS

The optimization program was written in Fortran IV requiring 208K bytes of storage and an average CPU time of about 95 seconds. The results obtained are presented in Figures 8- 17. Figures 8, 9, 10, and 13, 14, 15 show the variation of certainties in the nine regions corresponding to case 1 and case 2 respectively. From the simulation results, it was observed that the certainties fluctuate within the range of 0.5 to 1 in the first case, and 0.4 to 0.92 in the second case. Also, the certainties at the terminal time t_f are maintained at high values: 0.73 to 0.94 in the first case, and 0.52 to 0.98 in the second case. The optimal scanning scheme as shown in Figure 11 follows the sequence

4, 5, 1, 6, 8, 3, 2, ..., 1, 3, 1, and 2,

i.e. region 4 is scanned first and then region 5 and so on. The duration of scanning of the i th region (i = 1, ..., 9) is indicated by the width of the control u_i. The results shown in Figures 11 and 16 show that the radar visits the fence regions frequently and switches to regions 4,5,6,7,8 and 9 occasionally. The scanning scheme is intuitively correct since frequent scanning of the fence regions produces an increased certainty in other regions at later times. Regions 4, 5, 6, 7, 8, 9 are occasionally scanned to detect any target having penetrated the protected area while scanning one of the fence regions. However, the preference depends on the relative values of the weighting factors. Convergences of the cost functional J(u) for the two cases are shown in Figures 12 and 17.

From the results given in Figures 8, 9, 10, and 13, 14, 15 it is observed that the distribution of certainties depends on the time constants τ_i and $\hat{\tau}_i$ which in turn depend on target dynamics and radar detection statistics. The ratio of time constants used in Case 1 (Figures 8, 9, 10) is $\hat{\tau}_i : \tau_i = 40 : 1$ ($i = 1,\ldots,9$), whereas for Case 2 (Figure 13, 14, 15) is $\hat{\tau}_i : \tau_i = 10 : 1$. From the facts that the overall distribution of certainties in the first case is better than the second case, one may expect more improvement in certainties by choosing the ratio of the two time constants larger. This choice, however, depends on the power of the radar used as well as region size and target dynamics.

Regular raster scan was performed and the corresponding performance indices were found to be 73.81 for case 1 and 77.08 for case 2; whereas, optimal scanning produces performance indices 11.29 for case 1 and 24.82 for case 2 (see Figures 11, 12, 16, 17). These results clearly indicate the desirability of optimal scanning against conventional raster scanning.

In this chapter, a Two-Dimensional Nine-Region model was considered. A protected area of different partitioning configurations were considered in [15].

A linear three-region model was one of the different partitioning configurations considered in [15]. This is a special case of the Two-Dimensional Nine-Region model. It can be obtained easily from the dynamics of the Two-Dimensional Nine-Region model by appropriate choices of time constants and probabilities of entry of the regions. In this case, the protected area is divided into three regions and it is assumed that the target enters the protected area only through the fence region. From simulation results, it was observed that the certainties in the three regions were maintained reasonable high throughout the whole scanning interval. Moreover, it is interesting to observe that there were some time slots that the radar was idle, and the certainties can still be maintained at a acceptable high level.

Further discussions of these models will not be considered here, and interested readers are refered to [15]. The Fortran code used to solve the Two-Dimensional Nine-Region model is ~~listed~~ in [14:38-84].

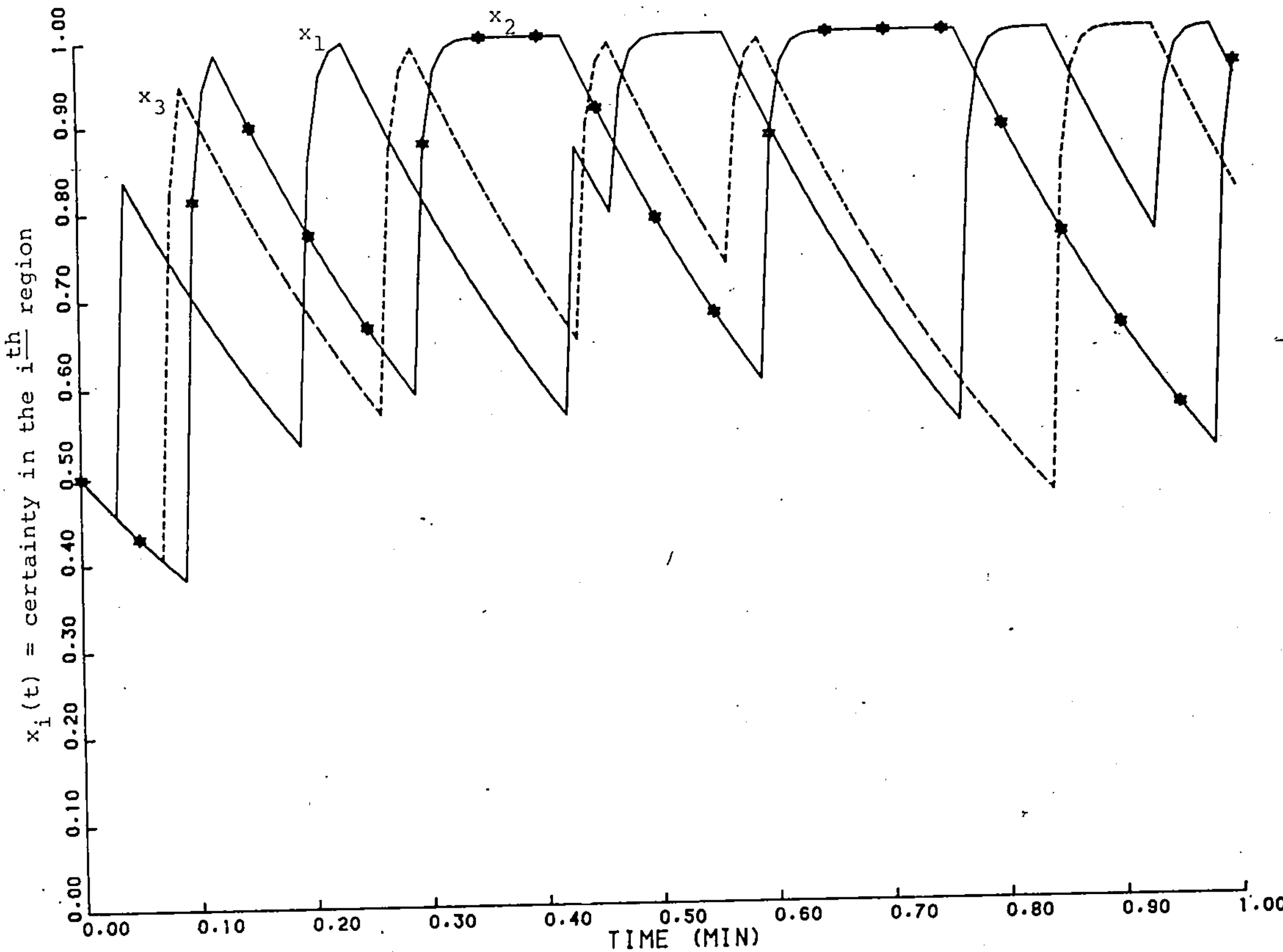


Figure 8: Case 1: Distribution of Certainties in Region 1 to 3

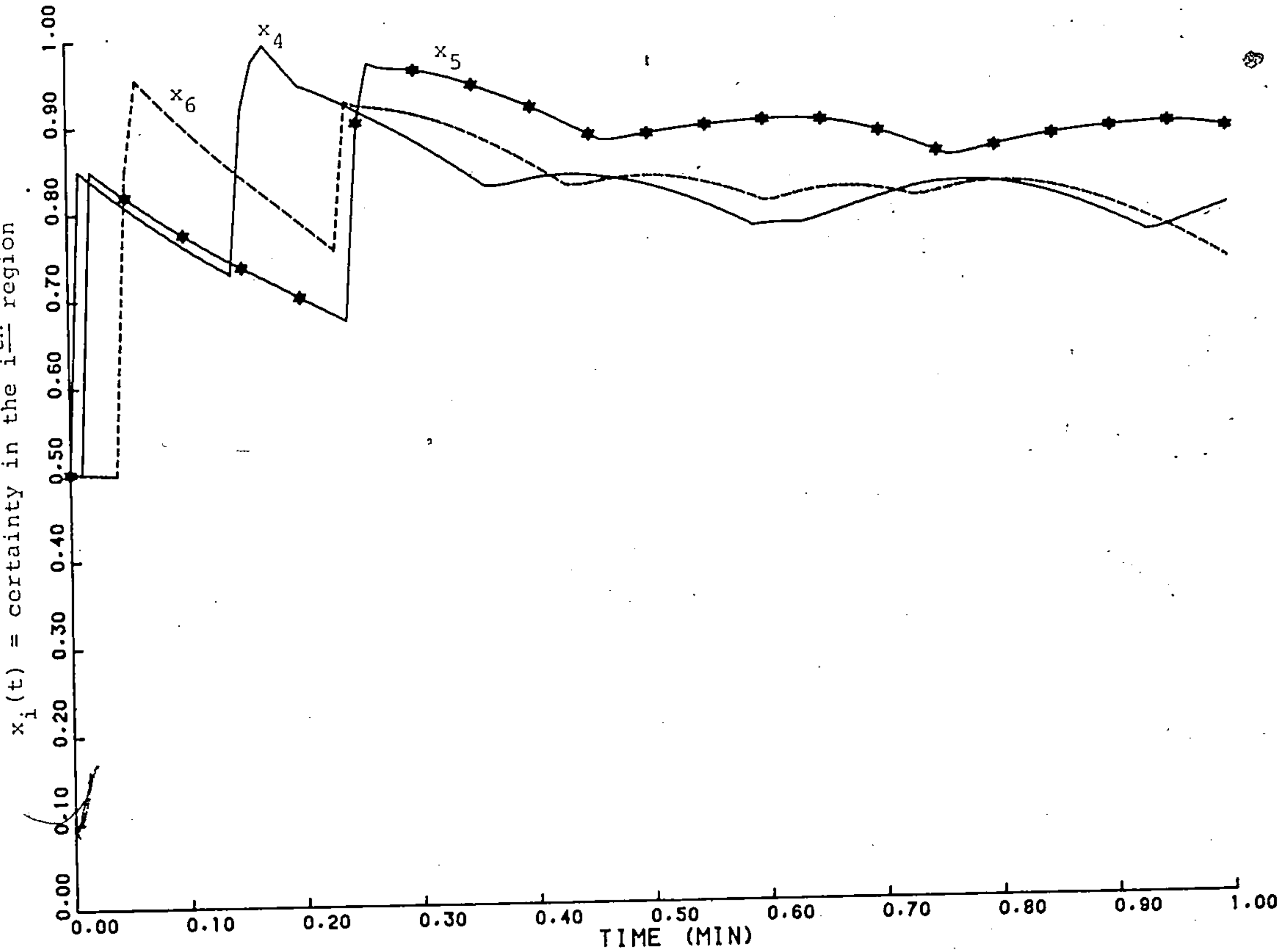


Figure 9: Case 1: Distribution of Certainties in Region 4 to 6

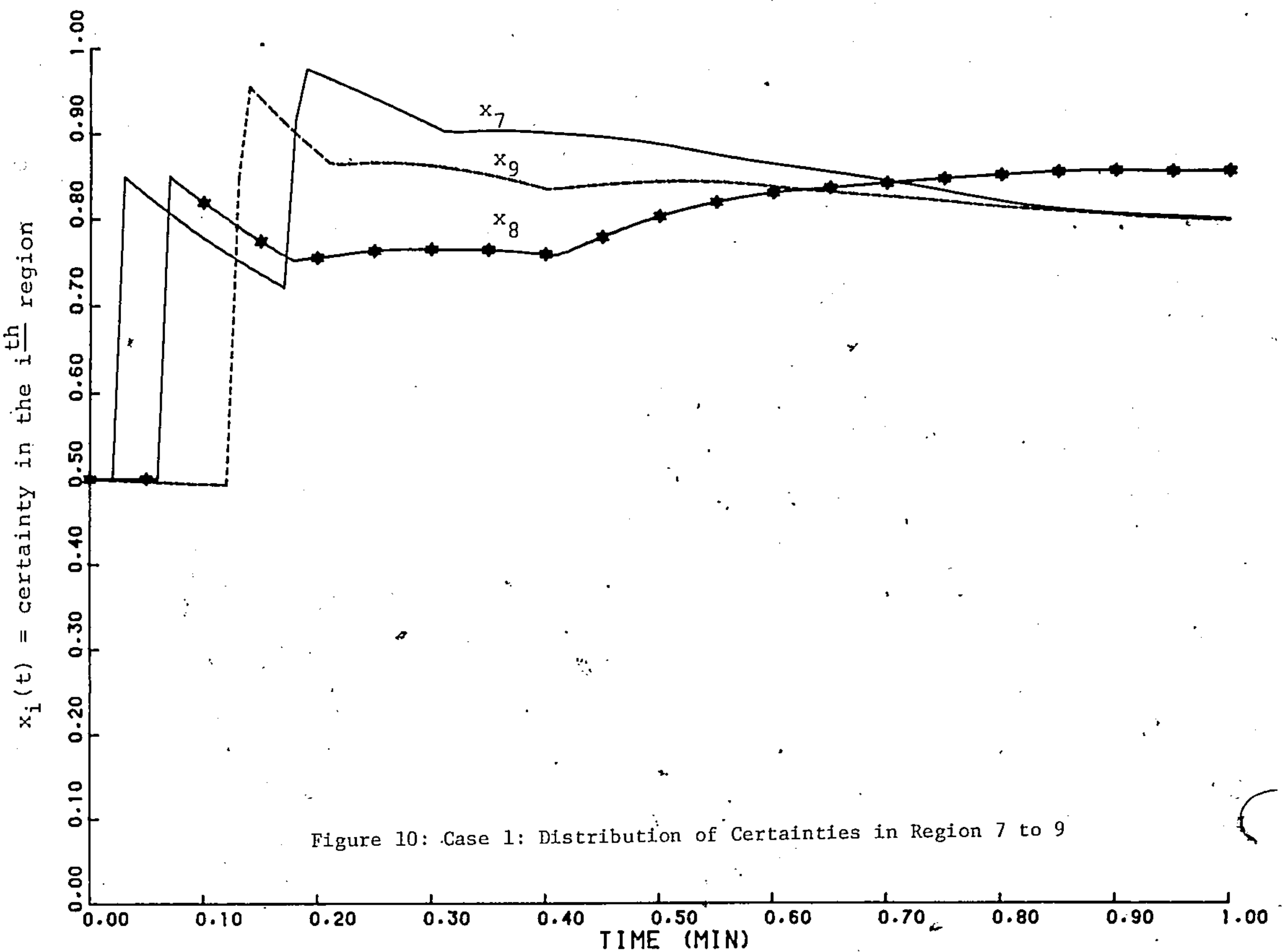


Figure 10: Case 1: Distribution of Certainties in Region 7 to 9

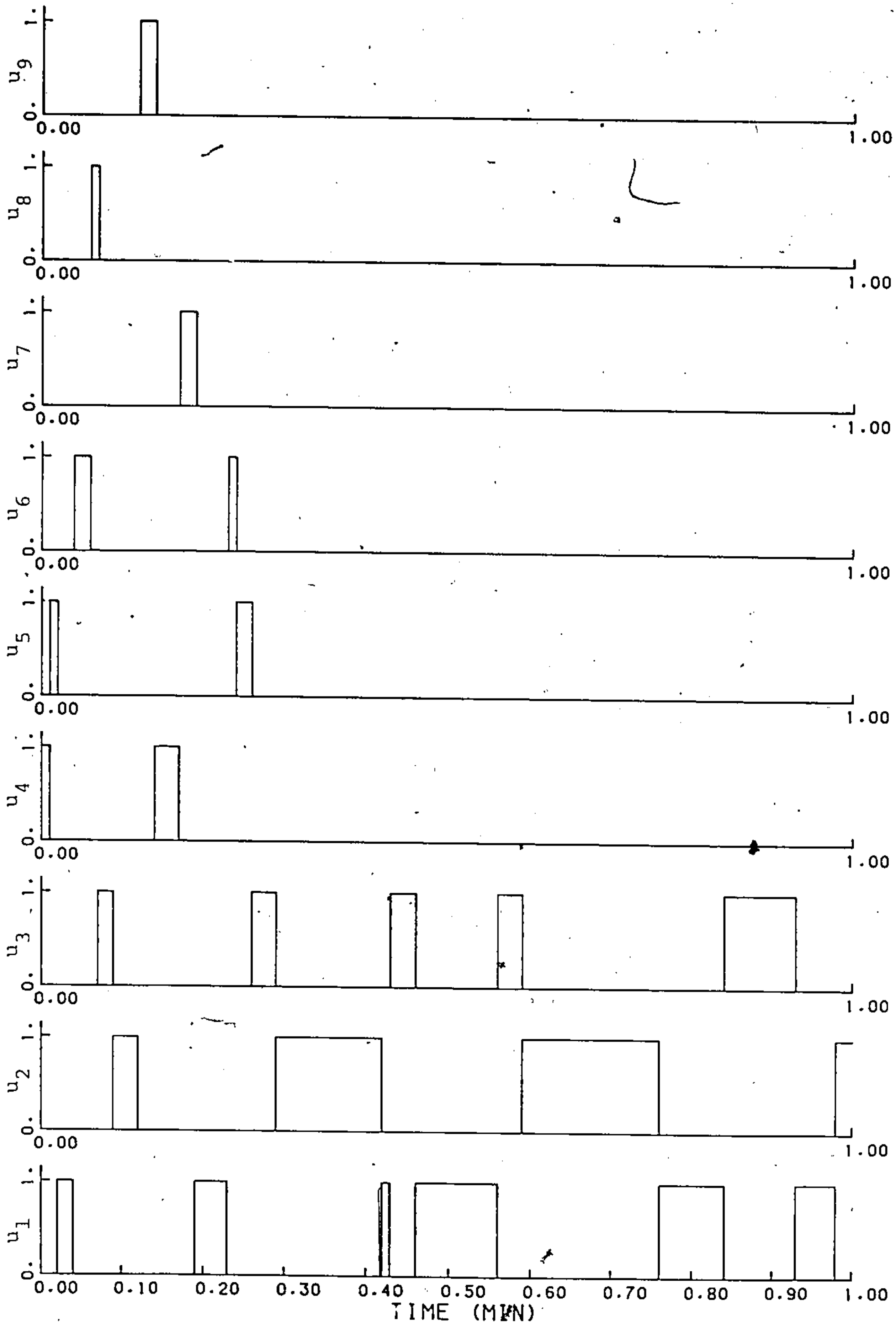


Figure 11: Case 1: Optimal Control Policy

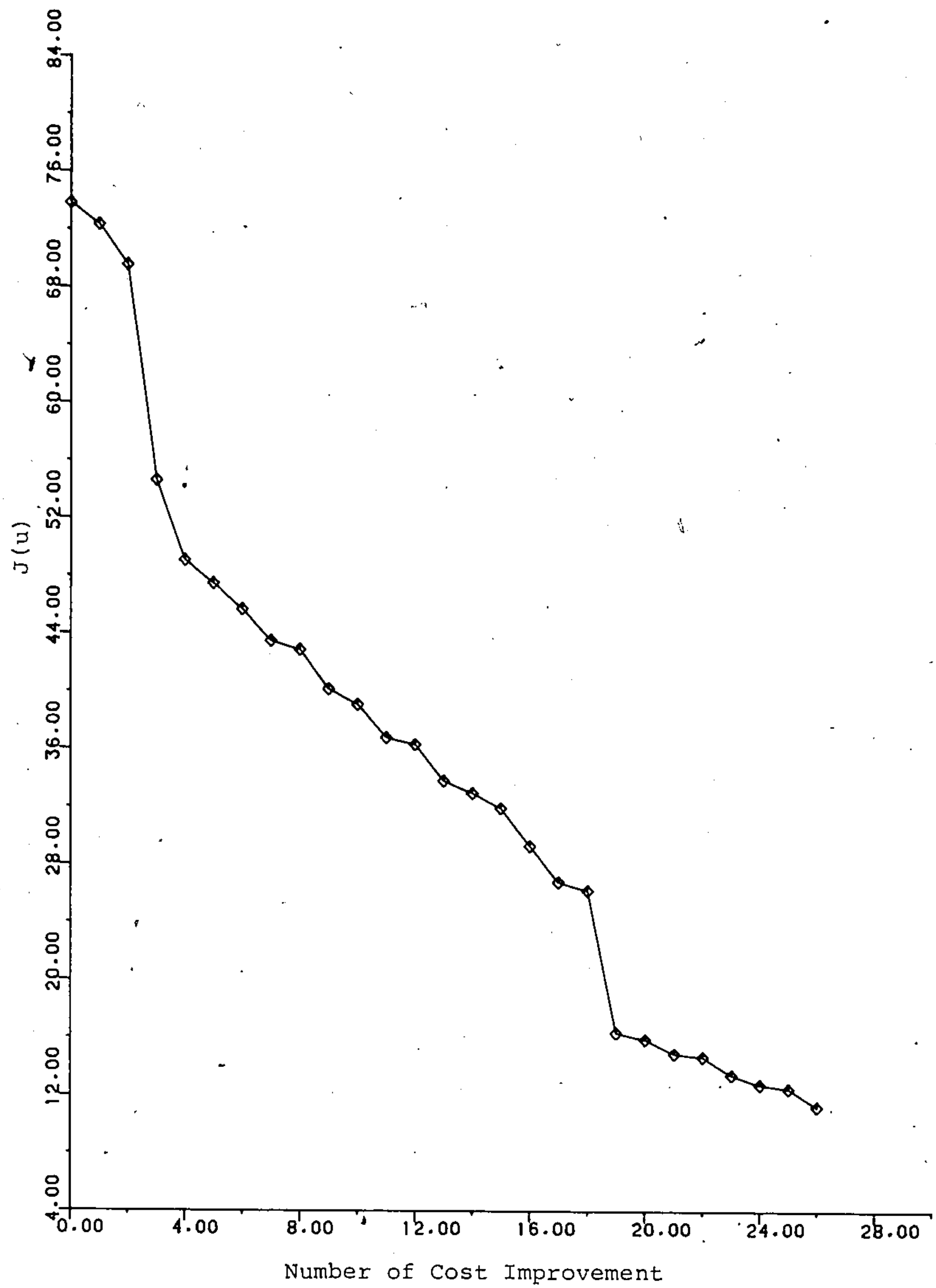


Figure 12: Case 1: Convergence of Solution

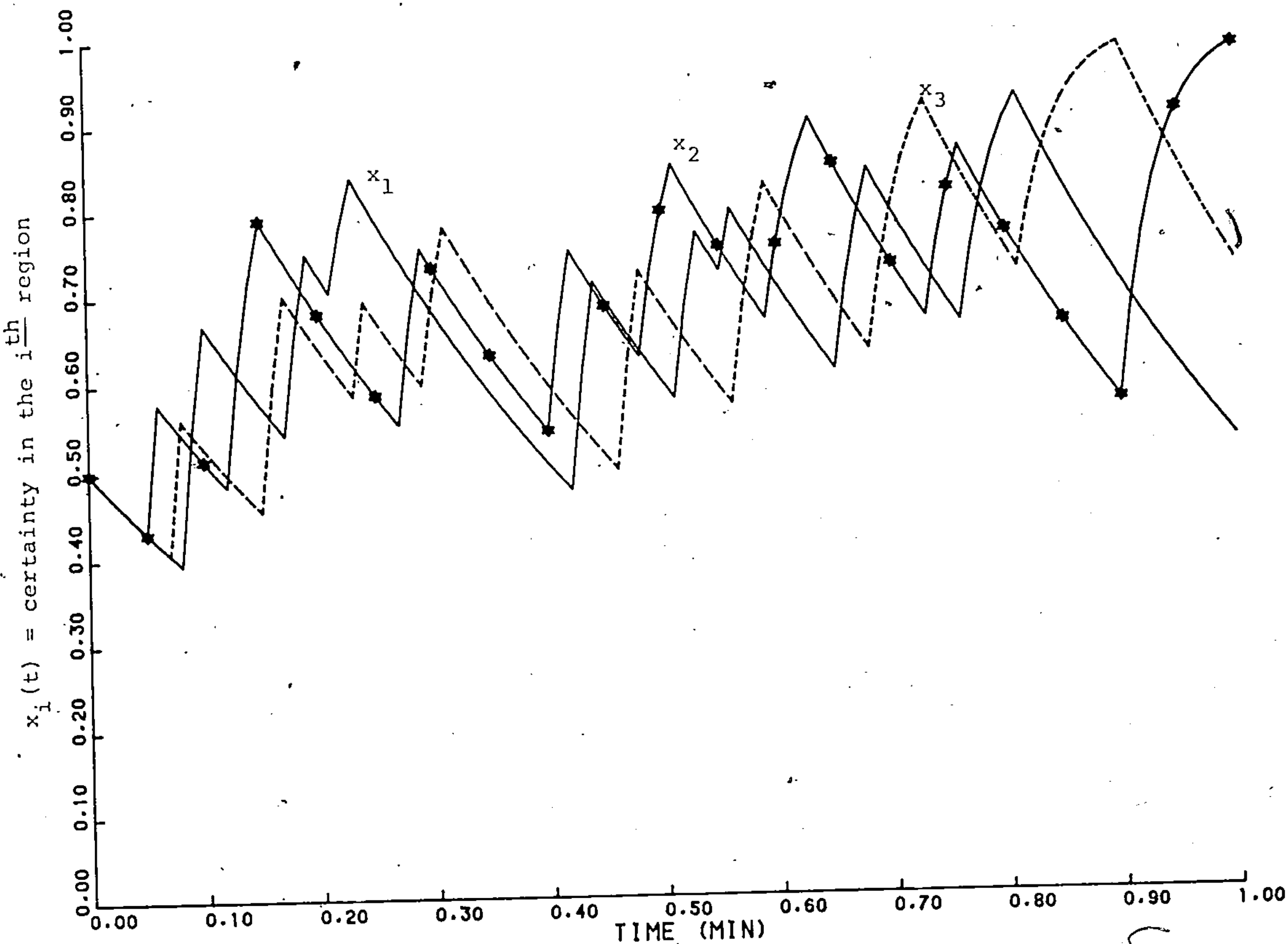


Figure 13: Case 2: Distribution of Certainties in Region 1 to 3

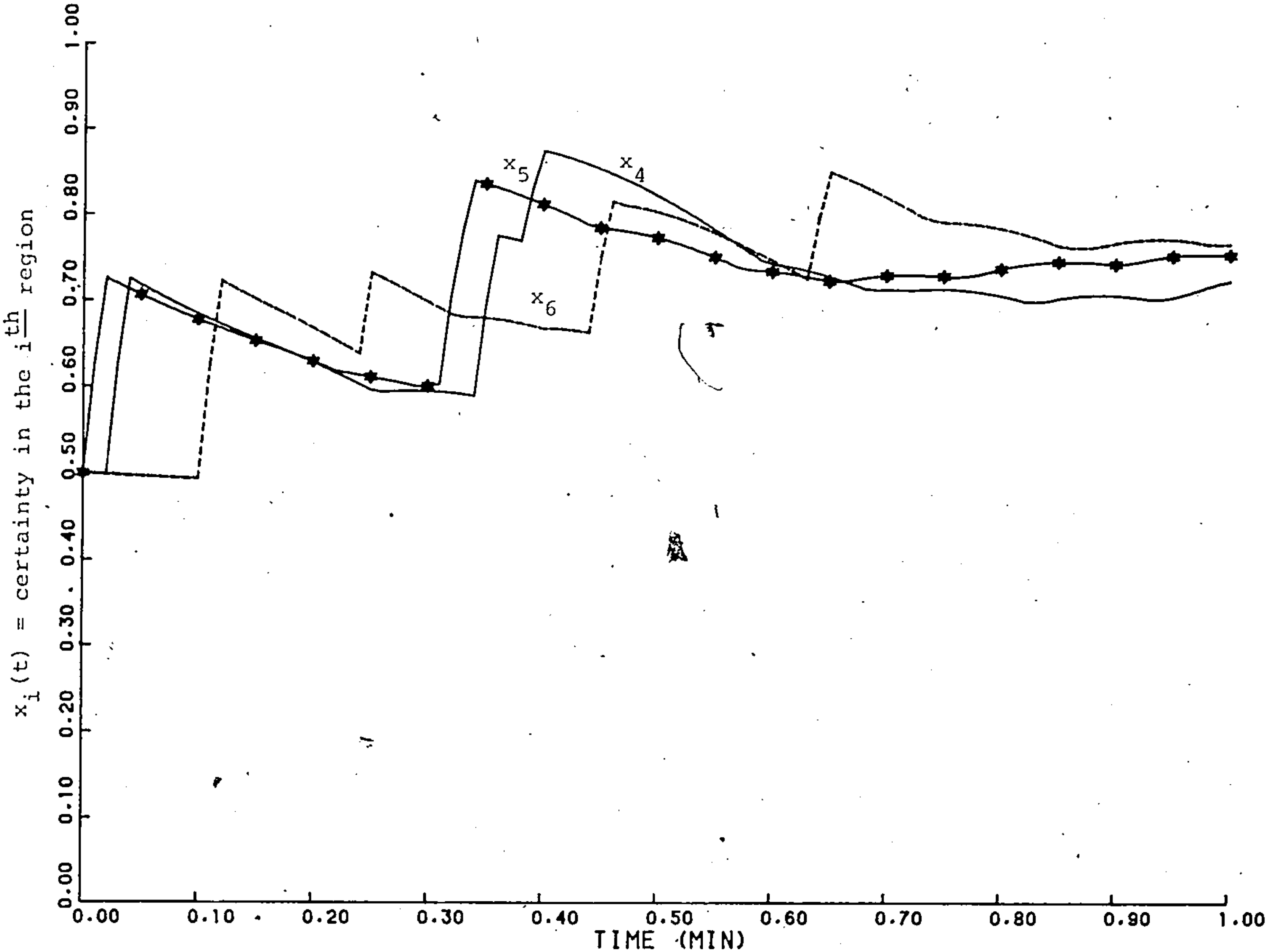


Figure 14: Case 2: Distribution of Certainties in Region 4 to 6

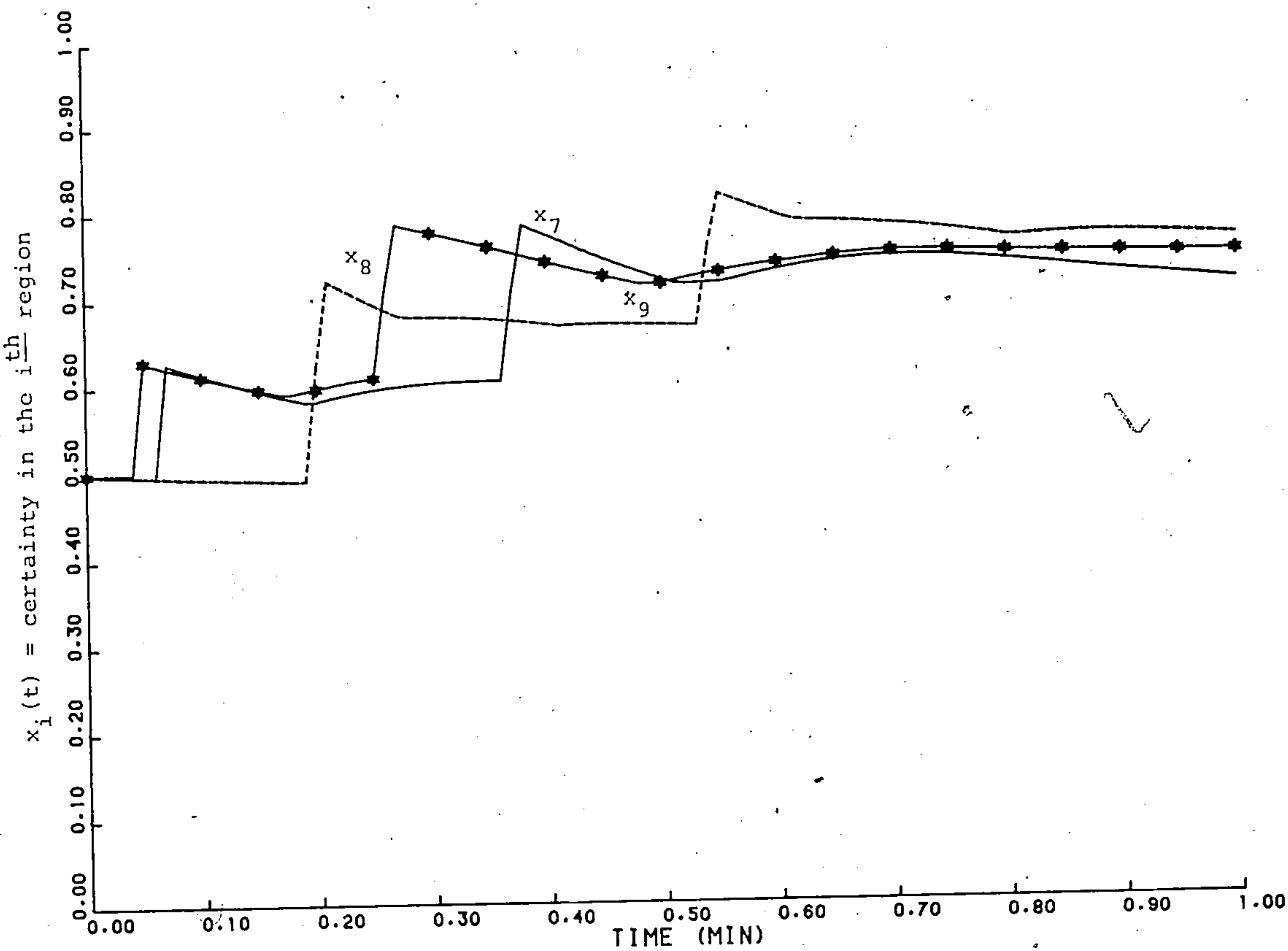


Figure 15: Case 2: Distribution of Certainties in Region 7 to 9

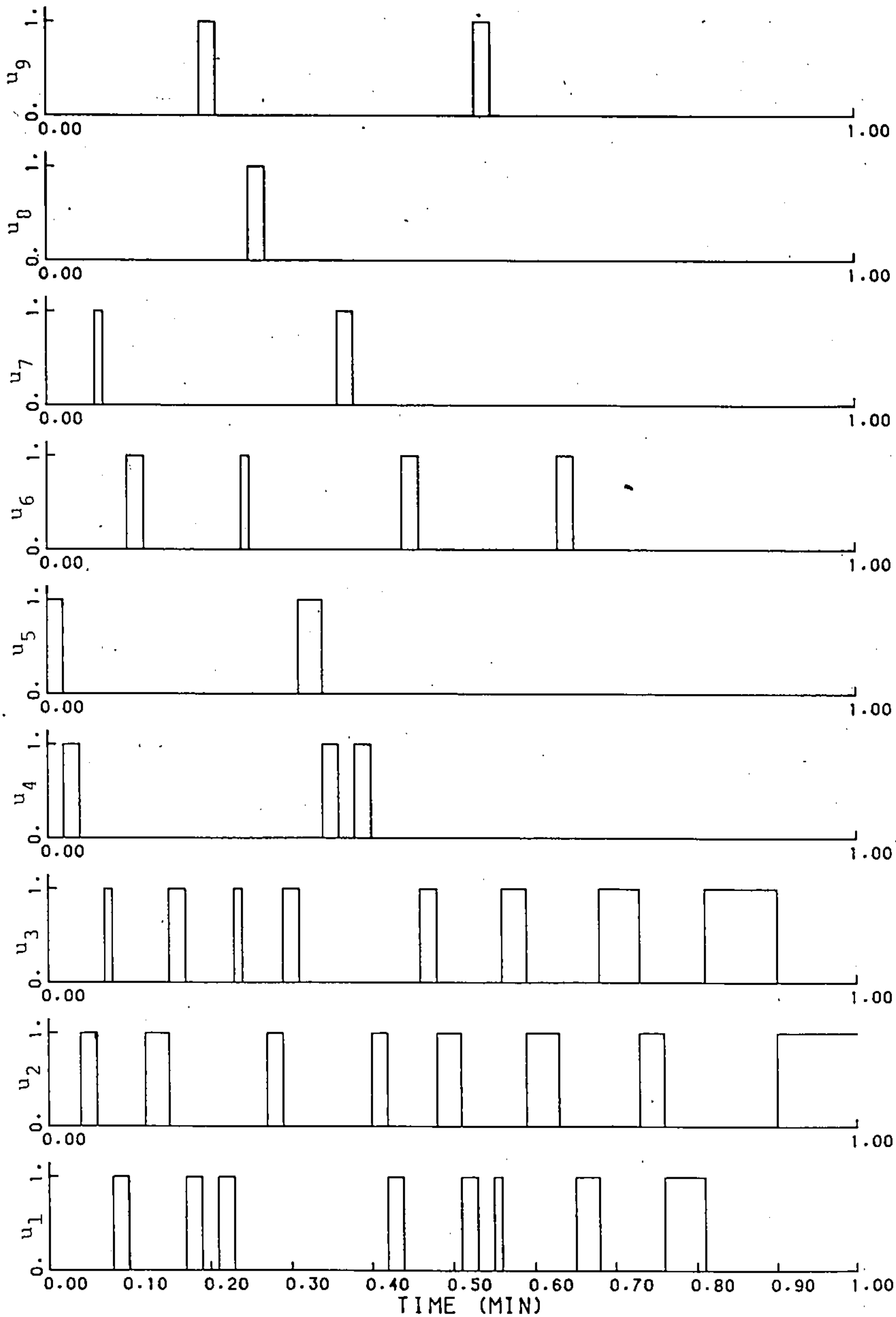


Figure 16: Case 2: Optimal Control Policy

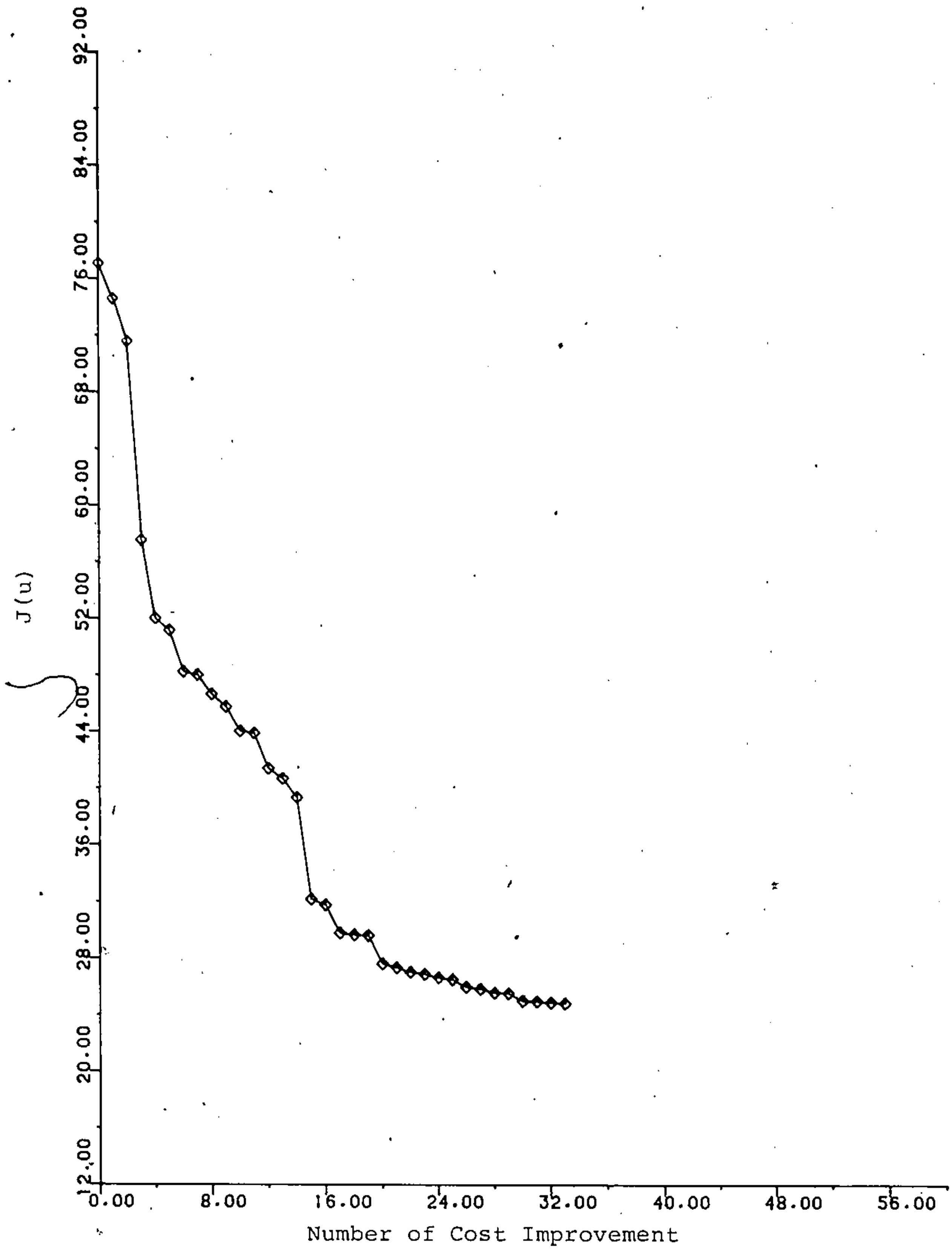


Figure 17: Case 2: Convergence of Solution

3.8 CONCLUSIONS

In this chapter a dynamic model for radar detection process has been proposed in which the protected area is partitioned into a number of regions. The time evolution of the certainties of the protected regions is governed by a system of linear delay differential equations that include scanning efforts as the control variables. Based on Optimal Control Theory, a computer program was developed to obtain the optimal scanning policy. For the purpose of illustration, an example of a Two-Dimensional Nine-Region model described by a system of nine first order differential equations was considered. The results are shown in Figures 8- 17. These results indicate that the overall certainty of the protected area can be substantially improved by optimal distribution of scanning efforts. On the basis of the results of this paper, the author strongly believes that radar detection process can be significantly improved by using optimal scanning.

3.9 SUGGESTIONS FOR FURTHER STUDIES

The model considered in this thesis is totally dependent on the time constants. The time constants are chosen so that the linear first order differential equation best approximates the actual statistical properties of a particular detection process used by the radar and of the movements of intruders. Once the time constants are fixed, the system becomes deterministic; so it may happen that the model does not describe the actual situation to a greater extent.

Owing to these reasons, it is desirable to model the detection problem in a stochastic processes approach. The process should be modelled by considering the electromagnetic waves transmitted and received. The model can be formulated by considering the variation of the received signal that is proportional to the sum of transmitted signal and the noise due to the background (thermal, sky, intentional noise-type jammers and clutter).

The author would like to suggest interested readers to look into the modelling of the detection process. A possible approach to optimize the detection of intruders is to compute optimal threshold level for each cell to declare an alarm. A preliminary formulation of the detection process mentioned above was included in [15:Sec.9].

Chapter IV

OPTIMAL ORBITAL CONTROL SCHEME FOR SYNCHRONOUS SATELLITE

A periodic orbital manoeuvre scheme is developed to maintain a desirable orbit. An optimal control is desired to keep the orbital elements within tolerable errors and to minimize the fuel consumption. The orbital dynamics of a synchronous satellite is known. An optimal control problem is formulated and solved by Conjugate Gradient Descent method. The optimal control policy is compared with the Sequential Control Scheme proposed in [16]. The results obtained can serve as guidelines in the design of orbital control systems [17].

4.1 INTRODUCTION

A terrestrial satellite is said to be synchronous if it travels at the same angular velocity as the Earth and thus remains constantly at the same meridian. Most communication satellites are in synchronous orbit since they can illuminate a large part of the Earth's surface at one time. For a satellite to be synchronous, it must be placed in an orbit at an altitude of about 22,300 miles. In order for a satellite to be geosynchronous, it should be placed in a synchronous circular orbit with a period of 23 hr and 56 min and its plane should be in perfect alignment with the plane containing the Earth's equator. 6

Due to launch imperfections and under the effect of external perturbations, the orbit tends to deform. This affects the geostationary (synchronous) character of the satellite thereby causing problems in communications.

The perturbation in the orbit of a stationary satellite can be classified into two components; (i) east-west drift caused mainly by deviation of the attraction of the Earth from the central force, and (ii) north-south drift caused mainly by lunar and solar attraction.

Frequent corrections of the orbital elements are required to keep the satellite within acceptable limits so that a direct line of sight (LOS) is maintained at all times between the satellite and the ground stations. On board thrusters placed at different angles are used to provide the necessary corrections. The life time of a satellite is limited by the fuel carried on board; so it is desirable to have a control scheme such that the fuel consumption is minimal and yet the orbital parameters are kept within the admissible limit.

The operation of an orbit control system (OCS) must be accomplished within the restrictions of economy and state-of-art technology. The system can be ground controlled or entirely satellite-borne and self contained. Both types can be related to a basic OCS with the essential elements shown in Figure 18.

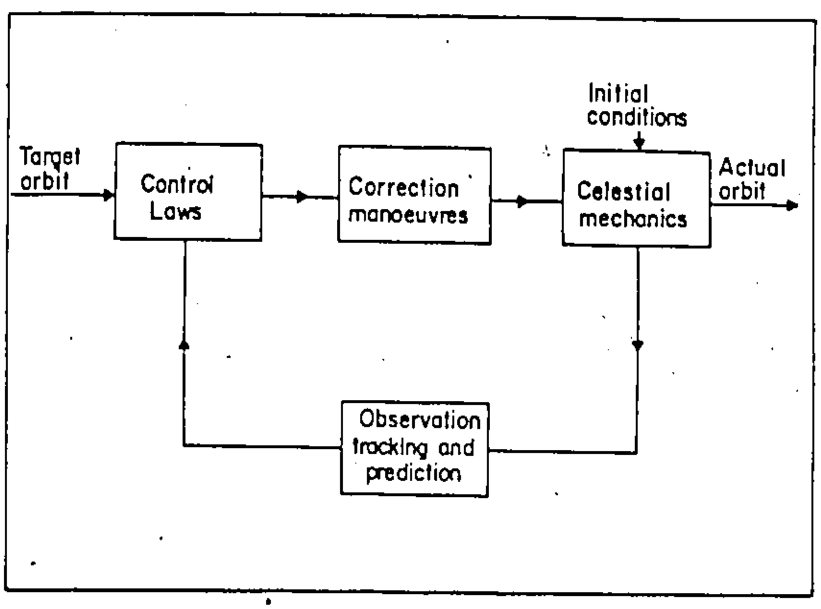


Figure 18: Basic Orbit Control System

With a few exceptions, most OCS are likely to be ground controlled due to the cost of incorporating comprehensive on-board computer facilities. The observations, computations, comparisons and optimizations are performed on the ground. Command signal is transmitted to the satellite by a suitable communication link. The command signal is decoded and stored in a timer and programmer. On reaching the time or position for execution of the correction manoeuvre, the timer initiates a programmed sequence for the manoeuvre which is executed by the guidance, control, and propulsion subsystem of the satellite. The satellite reacts according to the laws of celestial mechanics and moves into a corrected orbit. A closed loop control system therefore exists between the satellite and the ground station. The communication link between various subsystems on the ground and the satellite are shown in Figure 19 [18].

Before going into the optimal control problem, it is necesary to introduce the coordinate systems employed. A brief description of various orbital elements is also provided. The development of the Sequential Control Scheme will be highlighted.

In a recent paper [16], the authors proposed a state-dependent control law using the Lyapunov Stability Theory. A Sequential Control Scheme (non-optimal) was presented to correct the orbital parameters.

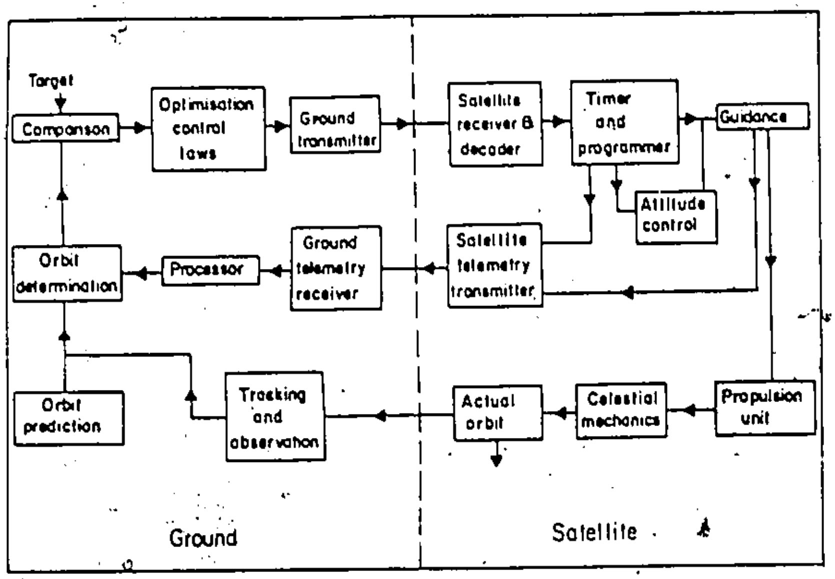


Figure 19: Satellite Orbit Control By Ground Based Operation

In this thesis, the interval between the orbit manoeuvre is determined, and an optimal control problem is formulated to minimize the fuel consumption and orbital deviation. A study is made to compare the optimal control policy presented in this thesis with the Sequential Control Scheme proposed in [16].

4.2 COORDINATE SYSTEMS

4.2.1 The Geocentric-Equatorial Coordinate System

The first requirement for describing an orbit is to define a suitable inertial reference frame. For satellites of the Earth, the Geocentric-Equatorial system is used. This coordinate system has its origin at the Earth's center. The fundamental plane is the Earth's equator and the positive X-axis points in the vernal equinox direction (this direction is defined by a line joining the center of the Earth on the first day of Autumn to the center of the Sun). The Z-axis points in the direction of the North Pole. The coordinate system is shown in Figure 20. The XYZ system is not fixed to the Earth and turning with it; rather, the Geocentric-Equatorial frame is non-rotating with respect to the stars (except for precession of the equinoxes) and the Earth turns relative to it. The unit vectors of the XYZ axes are represented by $\bar{I}\bar{J}\bar{K}$ respectively.

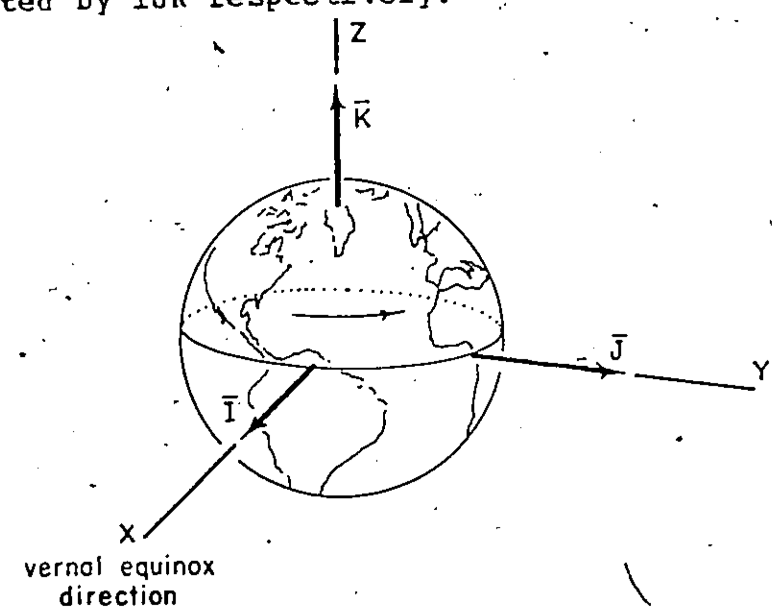


Figure 20: The Geocentric-Equatorial Coordinate System

4.2.2 The Perifocal Coordinate System

Perifocal coordinate system is one of the most convenient frames for describing the motion of a satellite. In this coordinate system, the fundamental plane is the satellite's orbital plane. The coordinate axes are named x_W, y_W, and z_W. The x_W axis points toward the periapsis; the y_W axis is rotated 90° in the direction of orbital motion and lies in the orbital plane; the z_W axis along $\overline{h}$ (the angular momentum vector) completes the right-handed perifocal system (Figure 21). Unit vectors in the direction of x_W, y_W, and z_W are denoted by $\overline{P}$, $\overline{Q}$, and $\overline{W}$ respectively.

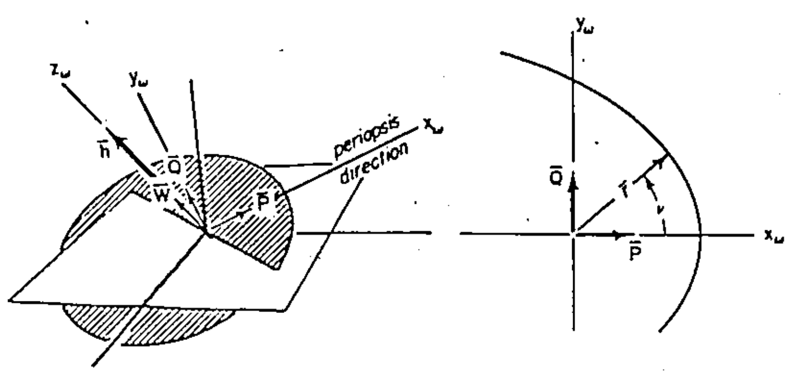


Figure 21: The Perifocal Coordinate System

4.2.3 The RTN Coordinate System

The reference coordinate system RTN (Figure 22) is a moving reference frame displaced from the ordinary perifocal system through an angle γ representing the true anomaly at time t. The frame XYZ represents the inertial coordinate system with its origin O located at the center of the Earth.

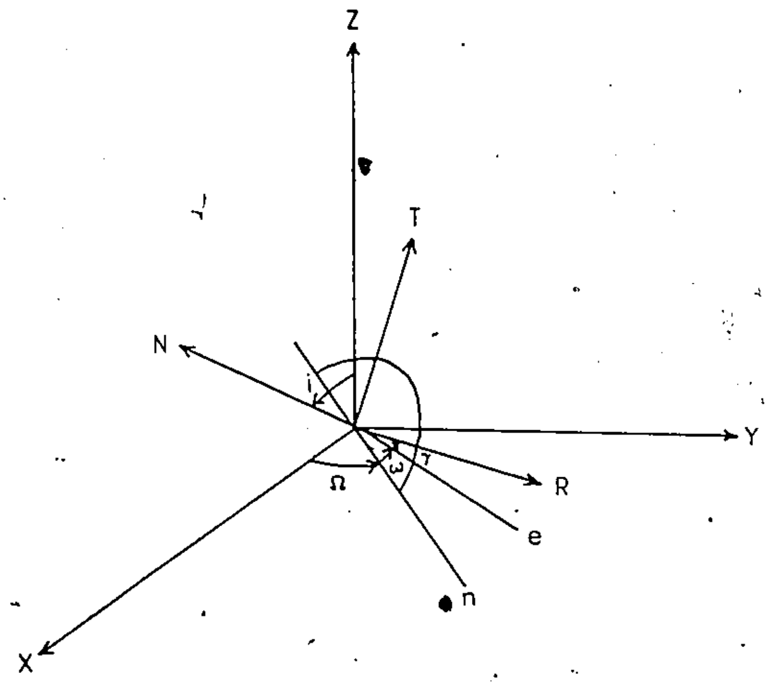


Figure 22: The RTN Coordinate System

4.3 CLASSICAL ORBITAL ELEMENTS

In order to describe the size, shape and orientation of an orbit it is necesary to define five independent quantities called 'orbital elements' or 'orbital parameters'. An additional elements, however is required to pinpoint the position of the satellite along its orbit at any particular time. The classical set of six orbital elements are defined with the help of Figure 23 as follow :

(i) a, semi-major axis - a constant defining the size of the conic orbit.

(ii) e, eccentricity - a constant defining the shape of the conic orbit

(iii) i, inclination - the angle between $\overline{K}$ unit vector and the angular momentum vector, $\overline{h}$.

(iv) Ω, longitude of the ascending node - the angle, in the fundumental plane, between the $\overline{I}$ unit vector and the point where the satellite crosses through the fundumental plane in a northerly direction (ascending node) measured counterclockwise when viewed from the north side of the fundumental plane.

(v) ω, argument of periapsis - the angle, in the plane of the satellite's orbit, between the asecending node and the periapsis point, measured in the direction of the satellite's motion. The extreme end-points of the major axis of an orbit are referred to as 'apses'. The point nearest to the prime focus is called 'periapsis'. Depending on what is the central attracting body in an orbital situation, 'periapsis' may also be called 'perigee'.

(vi) T, time of periapsis passage - the time when the satellite was at periapsis.

Instead of argument of periapsis, sometimes longitude of periapsis, Π is used. Anyone of the following parameters; namely, true anomaly at epoch, γ_0 ; argument of latitude at epoch, u_0 may be substituted for time of periapsis to locate the satellite at time t_0.

The three vectors shown in Figure 23 have the following definitions :

(i) $\overline{h}$, angular momentum vector

$$\overline{h} = \overline{r} \times \overline{v}.$$

where $\overline{r}$ and $\overline{v}$ are the position and velocity vectors of the satellite.

(ii) $\overline{n}$, node vector

$$\overline{n} = \overline{K} \times \overline{h}.$$

(iii) $\overline{e}$, eccentricity vector which points from the center of the Earth (focus of the orbit) toward perigee with a magnitude exactly equal to the eccentricity of the orbit.

4.4 PERTURBED ORBITAL EQUATIONS

In this section, the perturbed orbital equations for the satellite in the reference coordinate system RTN are presented [6:397-407].

Consider a satellite which is orbiting the Earth in a direct elliptic orbit (with inclination between 0° and 90°), and let its motion be perturbed by a force of constant acceleration having components f_R, f_T and f_N in a reference coordinate system RTN.

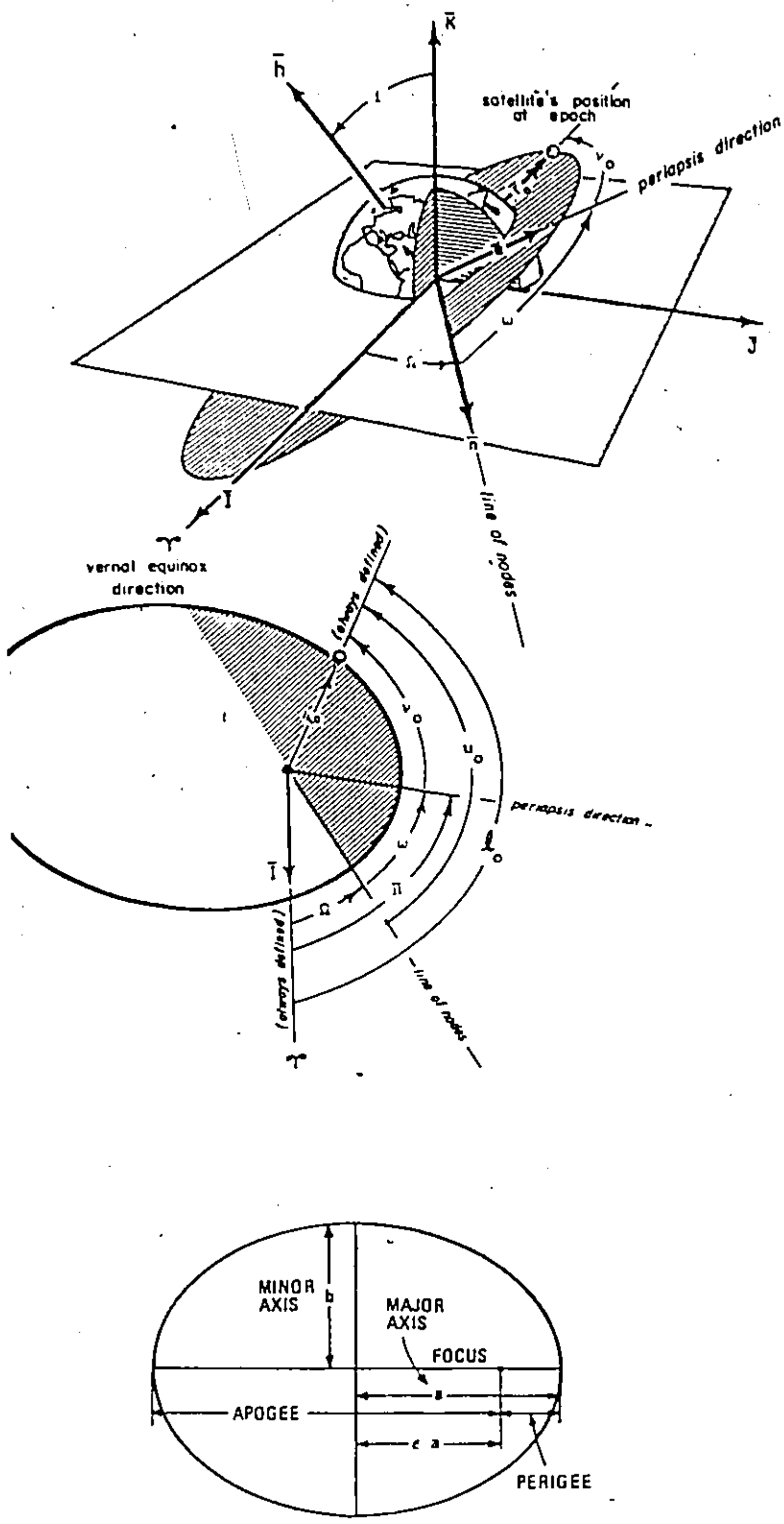


Figure 23: Illustration of the Orbital Elements

The perturbed orbital equations for the satellite in the reference coordinate system RTN can be written as

$$\dot{\tilde{a}} = \frac{2a^2}{\sqrt{\mu a(1-e^2)}} [(1+e \cos \gamma) f_T + e \sin \gamma f_R] \tag{4.1}$$

$$\dot{\tilde{e}} = \sqrt{\frac{a(1-e^2)}{\mu}} [\frac{2 \cos \gamma + e(1+\cos^2 \gamma)}{1 + e \cos \gamma} f_T + \sin \gamma f_R] \tag{4.2}$$

$$\dot{\tilde{i}} = \sqrt{\frac{a(1-e^2)}{\mu}} \frac{\cos(\omega+\gamma)}{1+e \cos\gamma} f_N \tag{4.3}$$

$$\dot{\tilde{\Omega}} = \sqrt{\frac{a(1-e^2)}{\mu}} \frac{\sin(\omega+\gamma)}{\sin i \ (1+e \cos\gamma)} f_N \tag{4.4}$$

$$\dot{\tilde{\omega}} = \frac{1}{e}\sqrt{\frac{a(1-e^2)}{\mu}} [\frac{2+e \cos\gamma}{1+e \cos\gamma} \sin \gamma f_T - \cos \gamma f_R - \frac{e}{1+e \cos\gamma} \cot i \sin(\omega+\gamma) f_N] \tag{4.5}$$

$$\dot{\tilde{\gamma}} = \frac{1}{e}\sqrt{\frac{a(1-e^2)}{\mu}} [- \frac{2+e \cos\gamma}{1+e \cos\gamma} \sin \gamma f_T + \cos \gamma f_R] \tag{4.6}$$

where,

$$a = a_0 + \tilde{a} \qquad \text{(semi-major axis)}$$
$$e = e_0 + \tilde{e} \qquad \text{(eccentricity)}$$
$$i = i_0 + \tilde{i} \qquad \text{(inclination)}$$
$$\Omega = \Omega_0 + \tilde{\Omega} \qquad \text{(longitude of ascending node)}$$
$$\omega = \omega_0 + \tilde{\omega} \qquad \text{(argument of periapsis)}$$
$$\gamma = \gamma_0 + \tilde{\gamma} \qquad \text{(true anomaly at epoch)}$$
$$\mu = 1.41 \times 10^{16} \ ft^3/s^2 \ \text{(gravitational constant)}$$

and where $\tilde{a}$, $\tilde{e}$, $\tilde{i}$, $\tilde{\Omega}$, $\tilde{\omega}$, $\tilde{\gamma}$ denote the deviation of the orbital parameters from their respective nominal values a_0, e_0, i_0, Ω_0, ω_0, γ_0 and f_R, f_T, f_N represent the components of the perturbing acceleration (both natural and control forces) acting along the radial, tangential, and normal axes of the reference coordinate system RTN. Throughout this thesis, the following nominal orbital paramters are used [16]:

$$a_0 = 1.4 \times 10^8 \ ft$$
$$e_0 = 1.5 \times 10^{-4}$$
$$i_0 = 1.67 \times 10^{-2} \ rad$$
$$\Omega_0 = -1.20 \ rad$$
$$\omega_0 = -1.60 \ rad$$
$$\gamma_0 = 0.90 \ rad. \qquad (4.7)$$

A simplified version of the orbital equations is given in [18]. The orbital elements a, e, i and Ω can be changed independently by the controls f_T, f_T, f_N and f_N respectively. However, the orbital element ω can be varied by f_R and f_N. The variation of true anomaly at epoch, γ is not considered in this simplified orbital equations.

The change in semi-major axis (a) is given by

$$\dot{a} = 2a \sqrt{\frac{a(1-e^2)}{\mu}}\ f_T \tag{4.8}$$

Generally, a change in eccentricity (e) is described by

$$\dot{e} = -\frac{3}{2}\ e \sqrt{\frac{a(1-e^2)}{\mu}}\ f_T \tag{4.9}$$

However, in order to change e independently of a, it is necessary for the thrust direction to be reversed at each crossing of the minor axis, and in this case the change in e is given by

$$\dot{e} = \frac{4}{\pi} \sqrt{\frac{a(1-e^2)}{\mu}}\ f_T \tag{4.10}$$

The change in inclination (i), longitude of ascending node (Ω) and argument of periapsis (ω) are respectively

$$\dot{i} = \frac{2}{\pi} \sqrt{\frac{a}{\mu}}\ (\mathrm{sgn}\ f_N)_{\Theta=0^\circ} |f_N| \tag{4.11}$$

$$\dot{\Omega} = -\frac{3\ a\ e \sin\omega}{2\sqrt{\mu a(1-e^2)}\ \sin i}\ f_N \tag{4.12}$$

$$\dot{\omega} = \sqrt{\frac{a(1-e^2)}{\mu}}\ f_R + \frac{3}{2}\ \frac{a\ e \cot i \sin\omega}{\sqrt{\mu a(1-e^2)}}\ f_N \tag{4.13}$$

where Θ is the argument of latitude [18].

4.5 REVIEW OF THE DEVELOPMENT OF SEQUENTIAL CONTROL SCHEME

In the subsequent sections, a study will be made to compare the Optimal Control Scheme developed in this thesis with the Sequential Control Scheme proposed by Ahmed and Sen [16]. At this stage, it would be appropriate to highlight the development of the Sequential Control Scheme.

A state-dependent control law is derived using the Lyapunov Stability Theory. A more practical means of orbit control based on the state-dependent control law is proposed and is known as the Sequential Control Scheme.

4.5.1 Basis of Control Using Lyapunov Stability Theory

The vector differential equation (4.1)-(4.6) can be expressed in the following form

$$\dot{x} = A(x_0+x)(u_n+u_c) \tag{4.14}$$

where u_n are the natural forces acting on the satellite, u_c is the control applied, x_o is the desired state of the satellite, and $x=(\tilde{a}, \tilde{e}, \tilde{i}, \tilde{\Omega}, \tilde{\omega}, \tilde{\gamma})^T$, the deviation of the state from x_o. $A(.)$ is an $(n \times m)$ matrix valued function and u_c is an m-vector.

The system will remain in the perturbed state $x_o + \xi$ (where $x(0) = \xi$) for all t when the natural forces, $u_n(t)$ are nullified by the control, $u_c(t)$ for all $t \geq 0$.

It is desirable to select control actions, preferably state feedback, so that the perturbation is eliminated in a short time after the occurence of the disturbance.

A state feedback control law $u_c(t)=u_c[x(t)]$ that stablizes the system around the nominal operating state x_0 will be developed by use of Lyapunov Stability Theory.

Consider the system given in (4.14). Let Q be a positive symmetric matrix (nxn) and let the Lyapunov function given by

$$V(x) = 0.5\ (\ x,\ Qx\ ). \tag{4.15}$$

The time derivative of V along any trajectory x is given by

$$\dot{V}(x) = [\ (u_n+u_c),\ A'(x_0+x)Qx\ ]. \tag{4.16}$$

Based on (4.16), two control laws are formulated.

Case 1 : Continuous Control Law

In (4.16), if one choose

$$u_n + u_c = -A'(x_0+x)\,Qx \tag{4.17}$$

or,

$$u_c = -u_n - A'(x_0+x)Qx, \text{ then} \tag{4.18}$$

$$\dot{V}(x) = -\,|A_0'\ (x\ +x)\,Qx\,|^2. \tag{4.19}$$

Hence,

$$V[x(t)] = V[x(0)] - 2\int_0^t |A'(x_0+x(\tau))Qx(\tau))|^2 \, d\tau. \qquad (4.20)$$

This cleary show that

$$V[x(t)] \leq V[x(0)] \text{ for all } t \geq 0$$

which means that the system is stable in the Lyapunov sense with control law (4.18). This control law, however, requires a control system with continuously adjustable thrusters.

If $A'(x_0+\xi)Q\xi \neq 0$ for all $\xi \neq 0$, then $\lim_{t\to\infty} V[x(t)]=0$ and $\lim_{t\to\infty} x(t)=0$, i.e. the system returns to the nominal operating state x_0 as $t \to \infty$.

Case 2 : On-Off Control Law

In (4.16), if one choose

$$u_c = - u_n - K \,\mathrm{sgn}[A'(x_0+x)Qx] \qquad (4.21)$$

where sgn is a signum function, and K is a nonnegative diagonal matrix (nxm), then

$$V[x(t)] = V[x(0)] - 2\int_0^t \{K\mathrm{sgn}(A'(x_0+x)Qx), A'(x_0+x)Qx\} d\tau$$

$$= V[x(0)] - 2\int_0^t \left(\sum_{i=1}^m K_i |(A'(x_0+x)Qx)_i|\right) d\tau. \qquad (4.22)$$

Again, by the same argument, the system is asymptotically stable, and the system returns to the nominal state x_0 asymptotically.

4.5.2 Development of Sequential Control Scheme

Since thrusters are generally operated as on-off devices, so only the on-off control law is considered here. From (4.21), the three componentes of the control law (assuming $u_n=(0,0,0)'$) are as follows :

$$f_R = u_{c1} = - K_1 \operatorname{sgn} (V_R) \tag{4.23}$$

$$f_T = u_{c2} = - K_2 \operatorname{sgn} (V_T) \tag{4.24}$$

$$f_N = u_{c3} = - K_3 \operatorname{sgn} (V_N) \tag{4.25}$$

where

$$V_R = \{2a^2\tilde{a}e\sin\gamma + \tilde{e}p\sin\gamma + (\tilde{\gamma}-\tilde{\omega})p\cos\gamma/e\}/h$$

$$V_T = \{2a^2\tilde{a}p/r) + [p(ap-r^2)\tilde{e}/aer] - (\tilde{\gamma}-\tilde{\omega})(p+r)\sin\gamma/e\}/h$$

$$V_N = \{\tilde{i}r\cos(\omega+\gamma) + [(\tilde{\Omega}/\sin i)-\tilde{\omega}\cot i]r\sin(\omega+\gamma)\}/h$$

$$p = a(1-e^2)$$

$$r = p/(1+e\cos\ )$$

$$h = (\mu p)^{0.5}, \text{ and}$$

Q was taken to be an identity matrix.

Based on the simulation results using control laws (4.23)-(4.25), it is observed that the parameter K_2 controls $\tilde{a}$; K_1 controls $\tilde{e}$ provided K_2 is zero; and K_3 controls $\tilde{i}$ exclusively.

A practical control scheme must drive the orbital errors to some target set regardless of their initial states. This control scheme is consist of the following sequential steps:

<u>Step 1 (predominantly controls $\tilde{i}$)</u>

For $0 \leq t \leq t_1$,

$K_3 \neq 0 \quad (K_1 = K_2 = 0),$

$q_{33} \neq 0 \quad (q_{ii} = 0, i \neq 3).$

<u>Step 2 (predominantly controls $\tilde{e}$)</u>

For $t_1 \leq t \leq t_2$,

$K_1 \neq 0 \quad (K_2 = K_3 = 0),$

$q_{11} \neq 0 \quad (q_{ii} = 0, i \neq 1).$

<u>Step 3 (predominantly controls $\tilde{a}$)</u>

For $t_2 \leq t \leq t_3$,

$K_2 \neq 0 \quad (K_1 = K_3 = 0),$

$q_{22} \neq 0 \quad (q_{ii} = 0, i \neq 2).$

In these three steps, q_{ii} $(i=1,2,...,6)$ represents the diagonal elements of the matrix Q. The above control scheme is known as the <u>Sequential Control Scheme</u> [16].

With all the background provided in the previous sections, one is ready to develop an optimal periodic control scheme for station keeping.

4.6 PERIODIC ORBITAL MANOEUVER

The prime objective in controlling the orbit of a satellite is to confine the orbital parameters within a set of target values. In general, it is more desirable to maintain the semi-major axis (altitude), eccentricity and inclination of the orbit within the desired set. In order to determine the interval between orbital manoeuvre, the following admissible errors are defined :

$$|a| \leqslant 6.2 \times 10^{3} \text{ ft} \tag{4.26}$$

$$|e| \leqslant 3.0 \times 10^{-5} \tag{4.27}$$

$$|i| \leqslant 1.6 \times 10^{-4} \text{ rad.} \tag{4.28}$$

A satellite drifts from its nominal orbit under perturbation of natural forces. The orbital parameters are determined by tracking and observing the satellite on ground. Whenever the satellite drifts out of the target set, orbital manoevure is performed. The satellite is permitted to drift naturally and the control cycle is repeated. The interval between each manoeuvre depends on the initial states of the orbital parameters.

As an illustration, the solar radiation torque is considered as the major natural disturbing torque acting on the satellite [19]. The components of the natural disturbances f_R, f_T, f_N along the RTN coordinate system as shown in Figure 24 are assumed. The control forces (non-optimal) used in obtaining the periodic manoeuvre are :

$$|f_R| = 1 \text{ ft/s}^2, \quad |f_T| = 25 \text{ ft/s}^2, \quad |f_N| = 50 \text{ ft/s}^2.$$

Based on these informations, a periodic orbital manoeuvre scheme is developed.

The interval between orbital manoeuvres is determined when any one of the constraints (4.26)-(4.28) is violated. Orbital manoeuvre is performed to bring the states within the target set. Next manoeuvre (time) can then be estimated by integrating the orbital equations (4.1)-(4.6), with the final state of the last manoeuvre taken as the initial state and the perturbing force present during the period in question. From simulation results as illustrated in Figure 25, it is observed that a complete manoeuvre cycle of the orbital parameters a, e and i takes about 2.41 days. During this period, manoeuvres of eccentricity (e), altitude (a) and inclination (i) were performed 1.81 days, 2.36 days and 2.41 days respectively after the initial control period. However, the manoeuvre period and frequency are dependent on the initial values of the states and the admissible error.

Fig.24 : Variation of Solar radiation torque

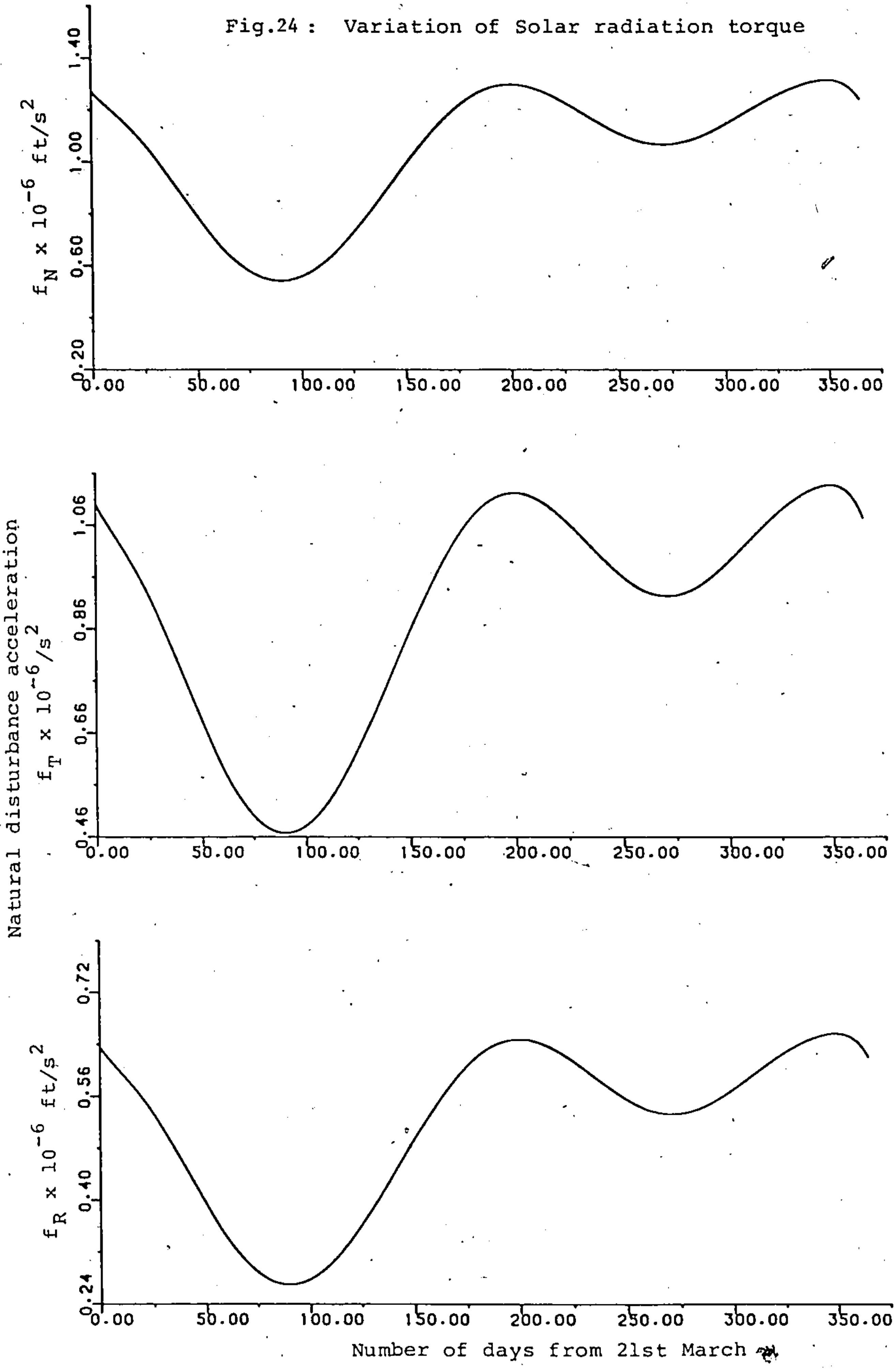
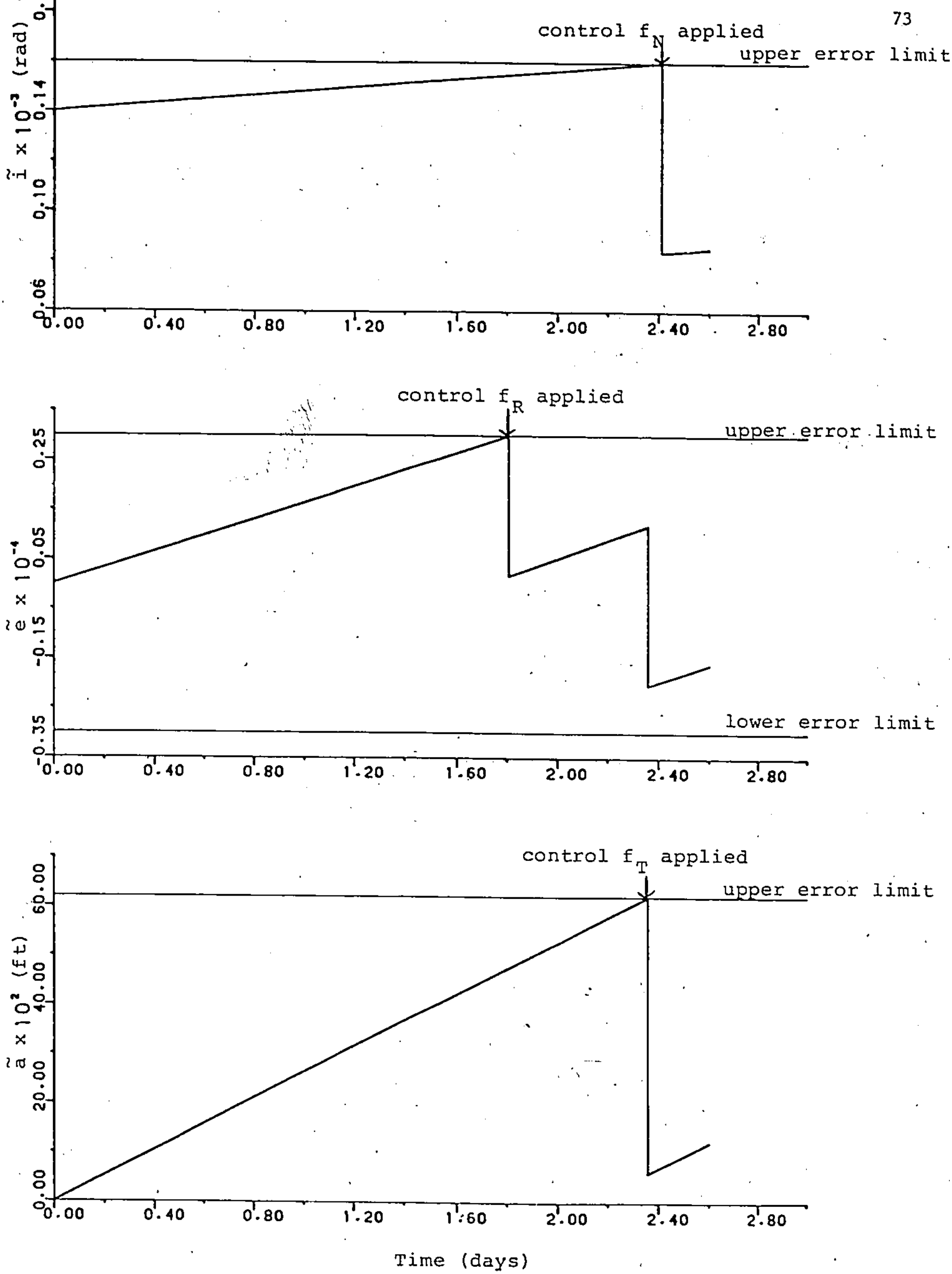


Fig 25 : Determination of period of orbital manoeuvre



4.7 FORMULATION OF OPTIMAL CONTROL PROBLEM

It is desirable to have a control policy for an orbiting satellite such that the orbital parameters are driven to the target set with minimum fuel consumption. This objective can be achieved by minimizing the following cost functional :

$$J(u) = \sum_{i=1}^{6} \beta_i |\tilde{x}_i(t_f)|^2 + \int_0^{t_f} \{ \sum_{i=1}^{6} \sigma_i |\tilde{x}_i(t)|^2 + \sum_{i=1}^{3} \lambda_i u_i^2 \} dt \quad (4.29)$$

where

$\tilde{x} = (\tilde{a}, \tilde{e}, \tilde{i}, \tilde{\Omega}, \tilde{\omega}, \tilde{\gamma})^T$,

$u = (f_R, f_T, f_N)^T$,

β_i ($i=1,\ldots,6$) is the weighting factor of the orbital parameter error at terminal time t_f,

σ_i ($i=1,\ldots,6$) is the weighting factor of the orbital parameter error in the interval $(0,t_f]$,

λ_i ($i=1,2,3$) is the weighting factor of the control, and

$\mathcal{U}_a$, the set of admissible controls, is the set of all measurable and bounded real-valued functions.

It is required to choose a control policy u so as to drive the system from its perturbed states to the desirable states in the fixed time interval $t\varepsilon(0,t_f]$ such that the cost functional J(u) described in (4.29) is minimized.

The explicit forms of the adjoint (costate) equations and the hamiltonian function corresponding to the orbital dynamics (4.1)-(4.6) are given in Appendix B.

4.8 METHOD OF SOLUTION

To solve the optimal control problem, it is required to introduce another system of differential equations called the adjoint (costate) equations $\psi(t)$, and the hamiltonian function, H (see Chapter 2). The problem described above was solved by using Conjugate Gradient Descent (CGD) method [8,9,10].

4.8.1 The Conjugate Gradient Descent Algorithm

The CGD algorithm requires the computation of the gradient trajectory which is defined as $g(u) = \partial H/\partial u$. Let $u^i(t)$ be the i th approximation to the optimal control $u^*(t)$. The corresponding gradient $g(u^i)$ is computed by solving the dynamics (state) equations forward in time, and the adjoint equations backward in time with the control u^i. A sequence of control ploicies $u^1, u^2, \ldots, u^i, u^{i+1}, \ldots;$ $u \in \mathcal{U}_a$ is constructed by the CGD method, and it can be shown that the resulting sequence of u 's converges to the optimal control u^* [8]. The algorithm can be described as follows :

Guess an initial control u^0, compute $J(u^0)$, $g_o = g(u^0)$, and the search direction $s_o = - g_o$ and perform the following steps in the i th iteration :

Step 1

Compute α_i by performing a one-dimensional minimization of the cost functional $J(u^i + \alpha_i s_i)$ with respect to α.

Step 2

Compute new control $u^{i+1} = u^i + \alpha_i s_i$. Stop the iteration if $| J(u^{i+1}) - J(u^i)| < \delta_1$ (δ_1 is a small number, usually of the order 10^{-5}).

Step 3

Compute the gradient $g_{i+1} = g(u^{i+1})$, and calculate

$$\beta_i = \frac{(g_{i+1}, g_{i+1})}{(g_i, g_i)}$$

where, $(g_i, g_j) = \int_0^{t_f} g_i(t)\, g_j(t)\, dt$.

If $|| g_{i+1} - g_i|| < \delta_2$ (δ_2 is a small number), iteration stops.

Step 4

Compute the search direction $s_{i+1} = - g_{i+1} + \beta_i s_i$, set $i=i+1$ and go to Step 1.

Flow chart of the Conjugate Gradient Descent procedure is summarized in Figure 26.

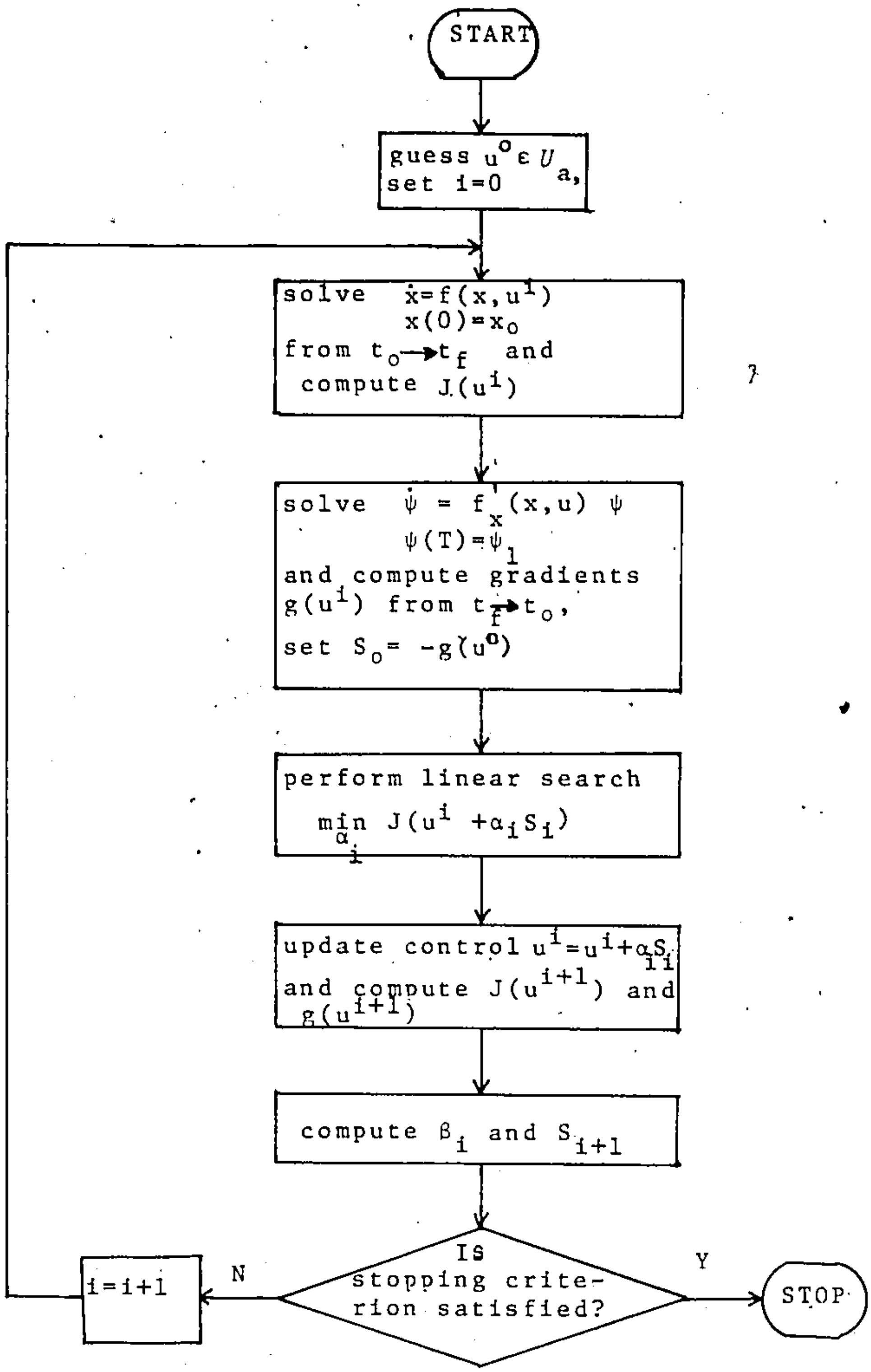


Figure 26: Flow Chart for Conjugate Gradient Descent Method

4.9 COMPUTATIONAL RESULTS AND PERFORMANCE ANALYSIS

The satellite orbital control problem (4.1)-(4.6), (4.29) is solved by using the algorithm presented in Section 4.8.

The weighting factors of the cost functional (4.29) are taken as

Case 1

$\sigma_1 = 10$	$\sigma_2 = 6.7\times10^6$	$\sigma_3 = 1.1\times10^6$	$\sigma_4 = \sigma_5 = \sigma_6 = 0$
$\beta_1 = 1.0\times10^{12}$	$\beta_2 = 6.7\times10^{17}$	$\beta_3 = 1.1\times10^{17}$	$\beta_4 = \beta_5 = \beta_6 = 0$
$\lambda_1 = 6.5\times10^2$	$\lambda_2 = 2.3\times10^{16}$	$\lambda_3 = 83.3$,	

Case 2

$\sigma_1 = 10$	$\sigma_2 = 3.0\times10^7$	$\sigma_3 = 1.1\times10^6$	$\sigma_4 = \sigma_5 = \sigma_6 = 0$
$\beta_1 = 1.0\times10^{12}$	$\beta_2 = 3.0\times10^{18}$	$\beta_3 = 1.1\times10^{17}$	$\beta_4 = \beta_5 = \beta_6 = 0$
$\lambda_1 = 2.1\times10^3$	$\lambda_2 = 1.5\times10^{11}$	$\lambda_3 = 83.3$.	

The weighting factors indicate the relative importance of the orbital elements and the control effort. In order to ensure that the system is driven to the desired states, the values of terminal weightings β_i should be much larger than the intermediate weightings σ_i $(i=1,\ldots,6)$.

Note that the final conditions obtained in the periodic orbital manoeuvres (Figure 25) can be used as the initial conditions in the optimization problem; however, in order to save computer time and storage, the initial conditions shown in Figures 27, 28, 30 and 31 are chosen.

From simulation results (Figures 27, 28, 30 and 31), it is observed that the orbital errors $\tilde{a}$, $\tilde{e}$, $\tilde{i}$ and $\tilde{\Omega}$ decay asymptotically from their initial values in the time interval $t \in (0, t_f]$. However, $\tilde{\omega}$ and $\tilde{\gamma}$ have different behaviour in both cases. Due to the facts that the controls f_T, f_R, f_N have more pronounced effects on the orbital elements a, e and i than Ω, ω and γ, it is difficult to drive all the orbital parameters to their desirable states.

The extremal controls obtained in both cases (Figures 29 and 32) show that the controls f_R and f_N are operating at constant magnitudes throughout the whole interval while f_T is variable.

From the results of the comparative studies of the Extremal Control and the Sequential Control Scheme, it is shown that the extremal controls drive the system to the same states in a much shorter time with less fuel than the Sequential Control Scheme. A fuel saving of about 11 % can be achieved by using the Extremal Control Scheme. Results of the studies indicated that the Extremal Control Scheme is capable of correcting the orbital parameters a, e and i simultaneously, thus enables the satellite to restore its orbit faster than the Sequential Control Scheme. As a comparison, the results of the two control schemes are summarised as follows:

	Optimal Control Scheme	Sequential Control Scheme
Case 1		
$J(u)$	0.217945×10^{13}	0.258779×10^{13}
$\int_0^{t_f} u^2 dt$	0.217930×10^{13}	0.243810×10^{13}
Control Period	0.85 s	2.395 s
Case 2		
$J(u)$	0.206028×10^{8}	0.127342×10^{12}
$\int_0^{t_f} u^2 dt$	0.201497×10^{8}	0.223509×10^{8}
Control Period	0.85 s	2.305 s

From the results shown in Figure 25 it is observed that the orbital errors $\tilde{a}$, $\tilde{e}$ and $\tilde{i}$ drift out of their admissible states in a linear fashion. Based on these observations, it is possible, by linear extrapolation, to predict the moment at which the satellite will drift out of its desirable limit. An extremal control scheme can then be computed off-line by using the extrapolated state errors as initial conditions for the next orbital manoeuvre. This procedure yields an optimal periodic control scheme for station keeping. The results presented in this chapter can serve as guidelines in the design of the orbital control systems.

The Fortran code used to solve this problem is listed in Appendix C.

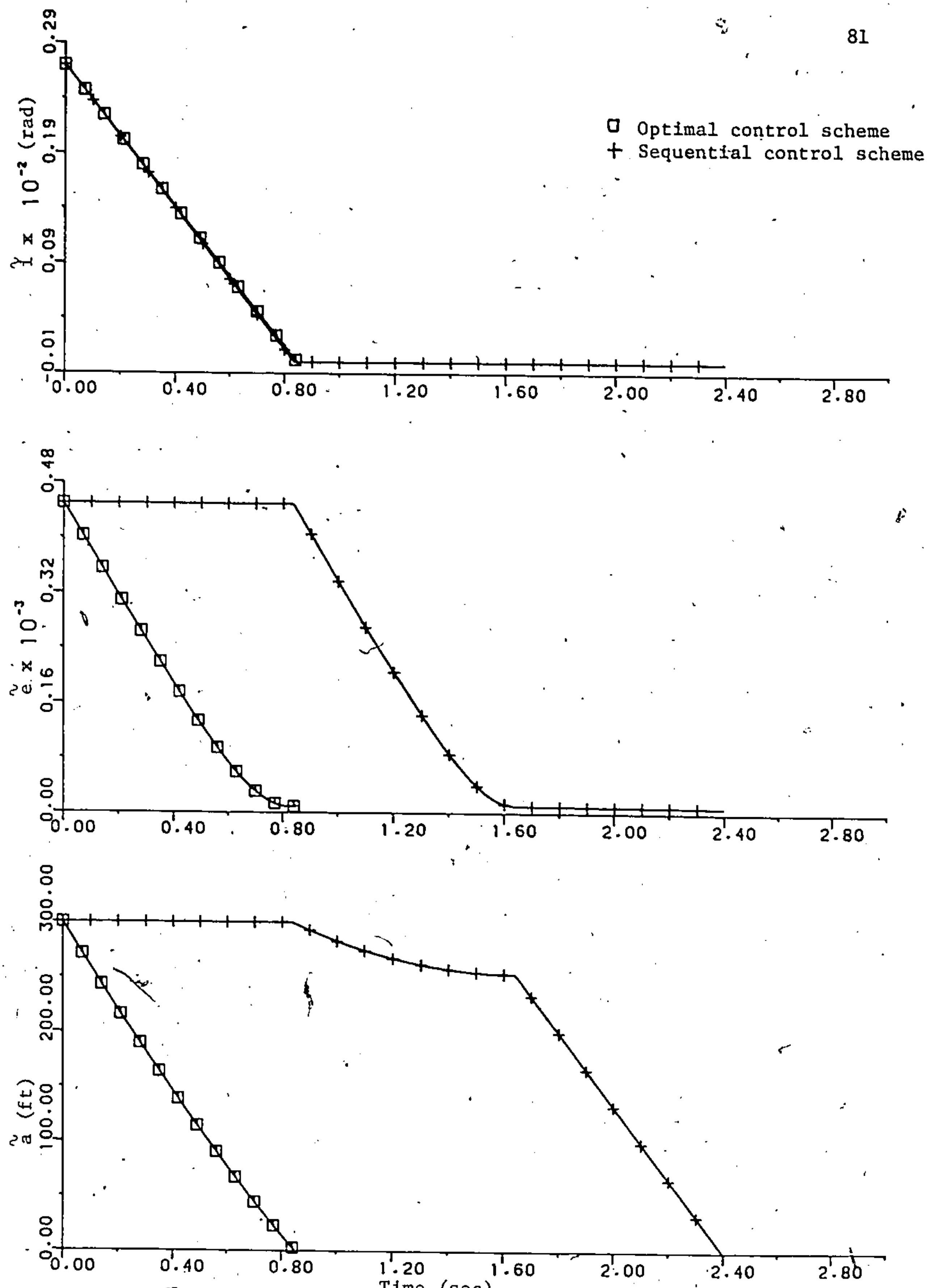


Figure 27: Case 1: Error response of optimal and sequential schemes
$\tilde{a}(o)=300\text{ft}$, $\tilde{e}(o)=0.45\times10^{-3}$, $\tilde{i}(o)=0.27\times10^{-2}$ rad

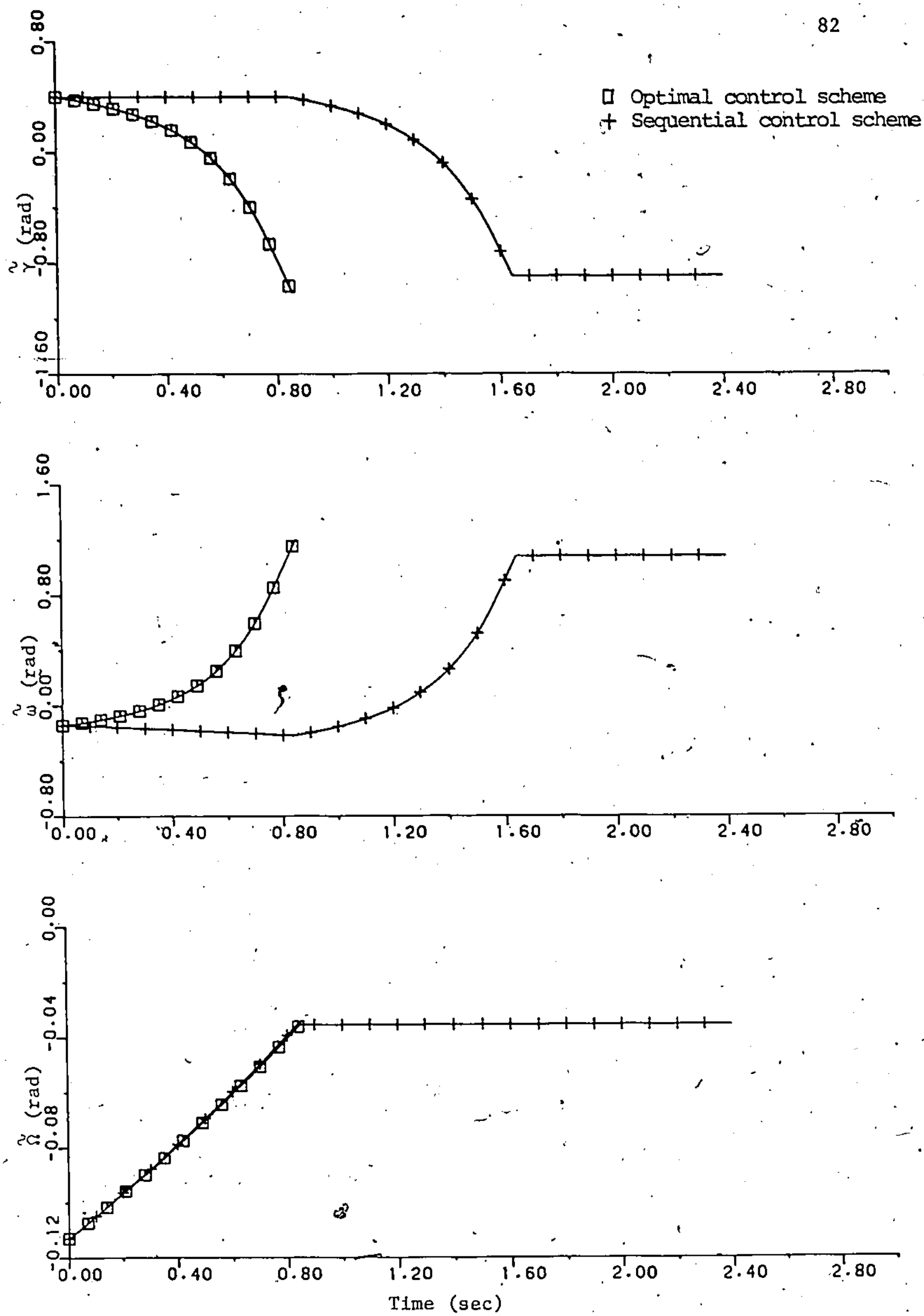


Figure 28: Case 1: Error response of optimal and sequential schemes
$\tilde{\Omega}(o)=-0.113$ rad, $\tilde{\omega}(o)=-o.14$ rad, $\tilde{\gamma}(o)=o.4$ rad.

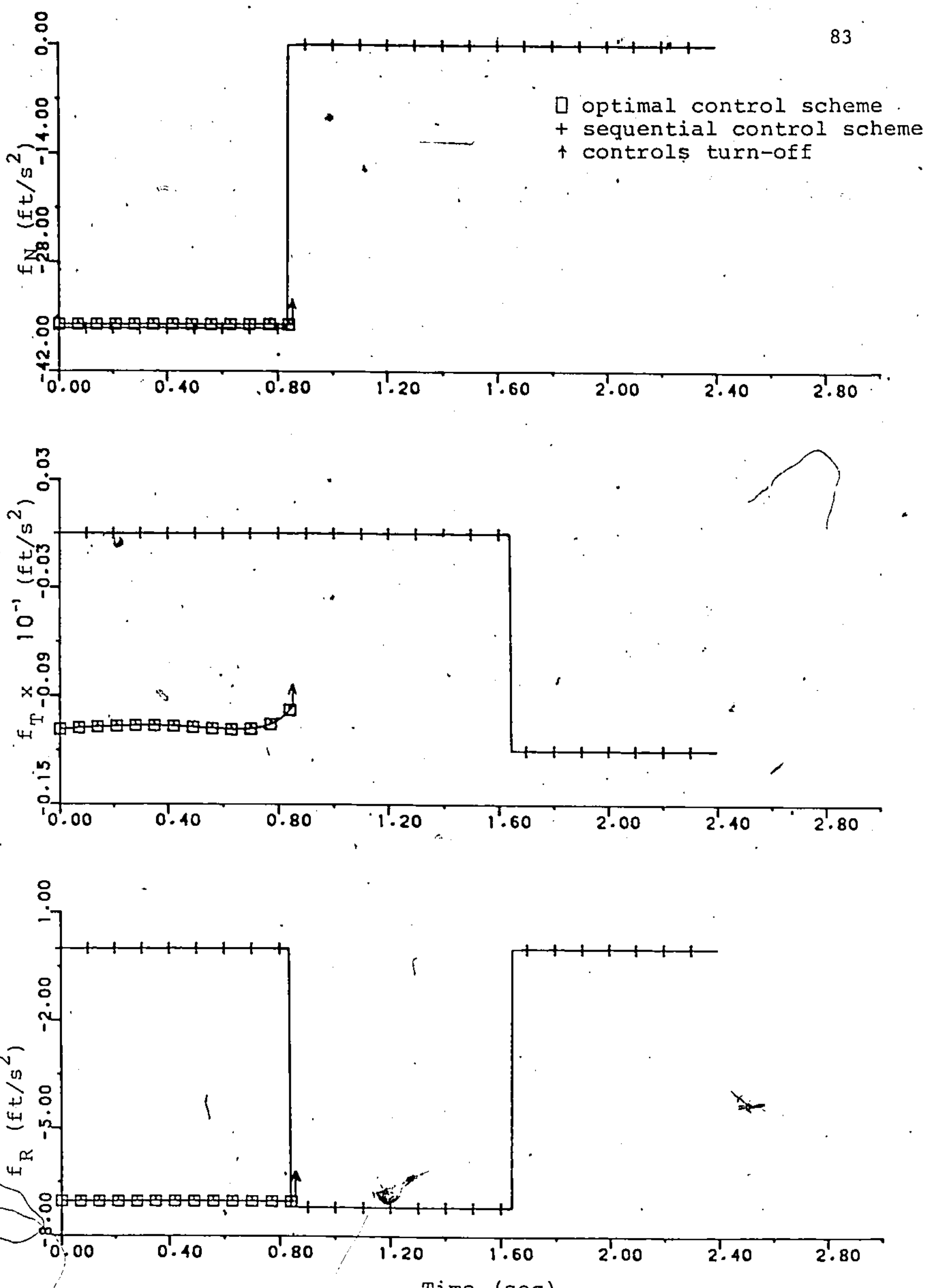


Fig.29 : Case 1: Optimal and sequential control policies

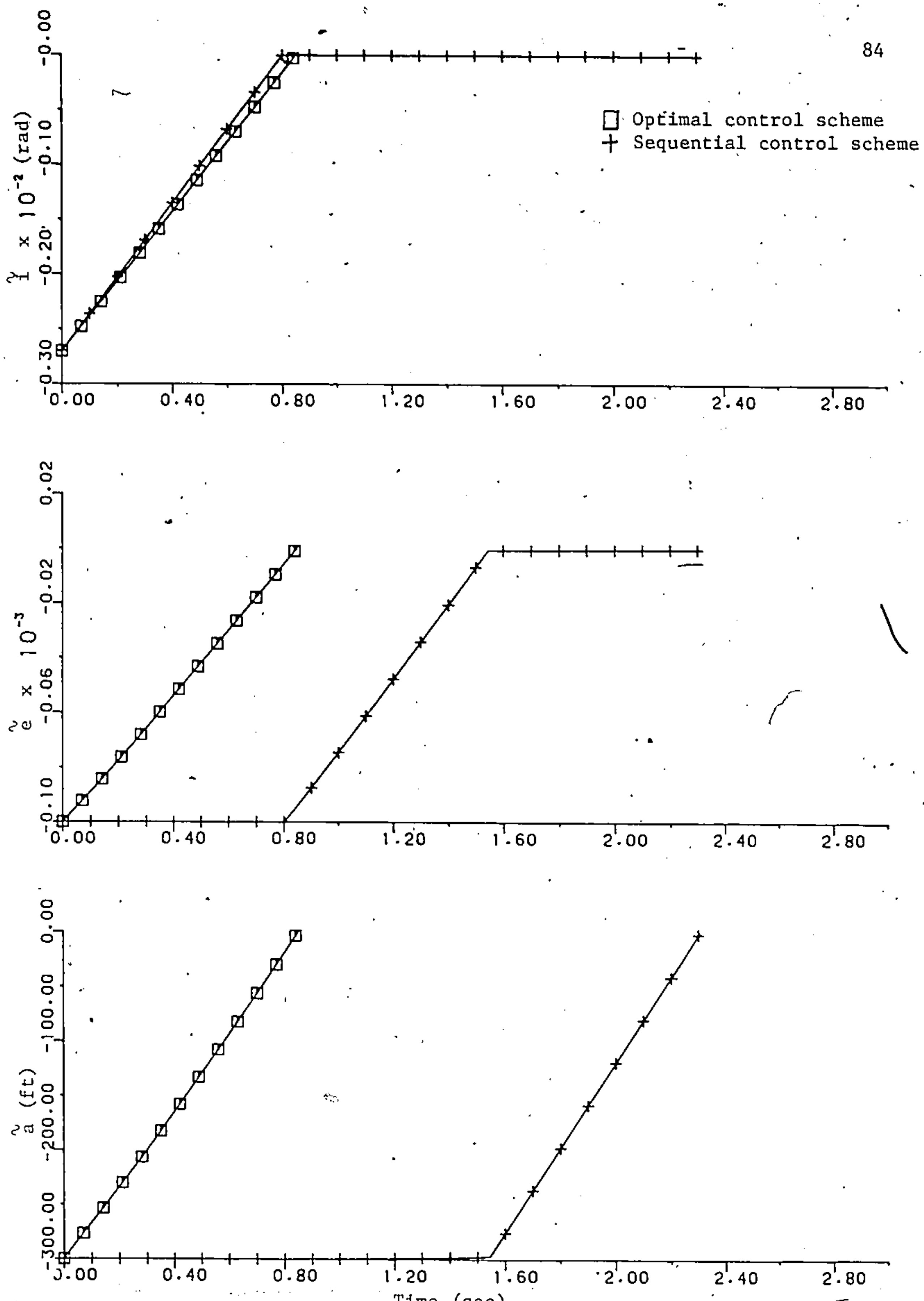


Figure 30: Case 2: Error response of optimal and sequential control schemes
$\tilde{a}(o)=-300ft$, $\tilde{e}(o)=-o.1x10^{-3}$, $\tilde{i}(o)=-o.27xlo^{-2}$ rad.

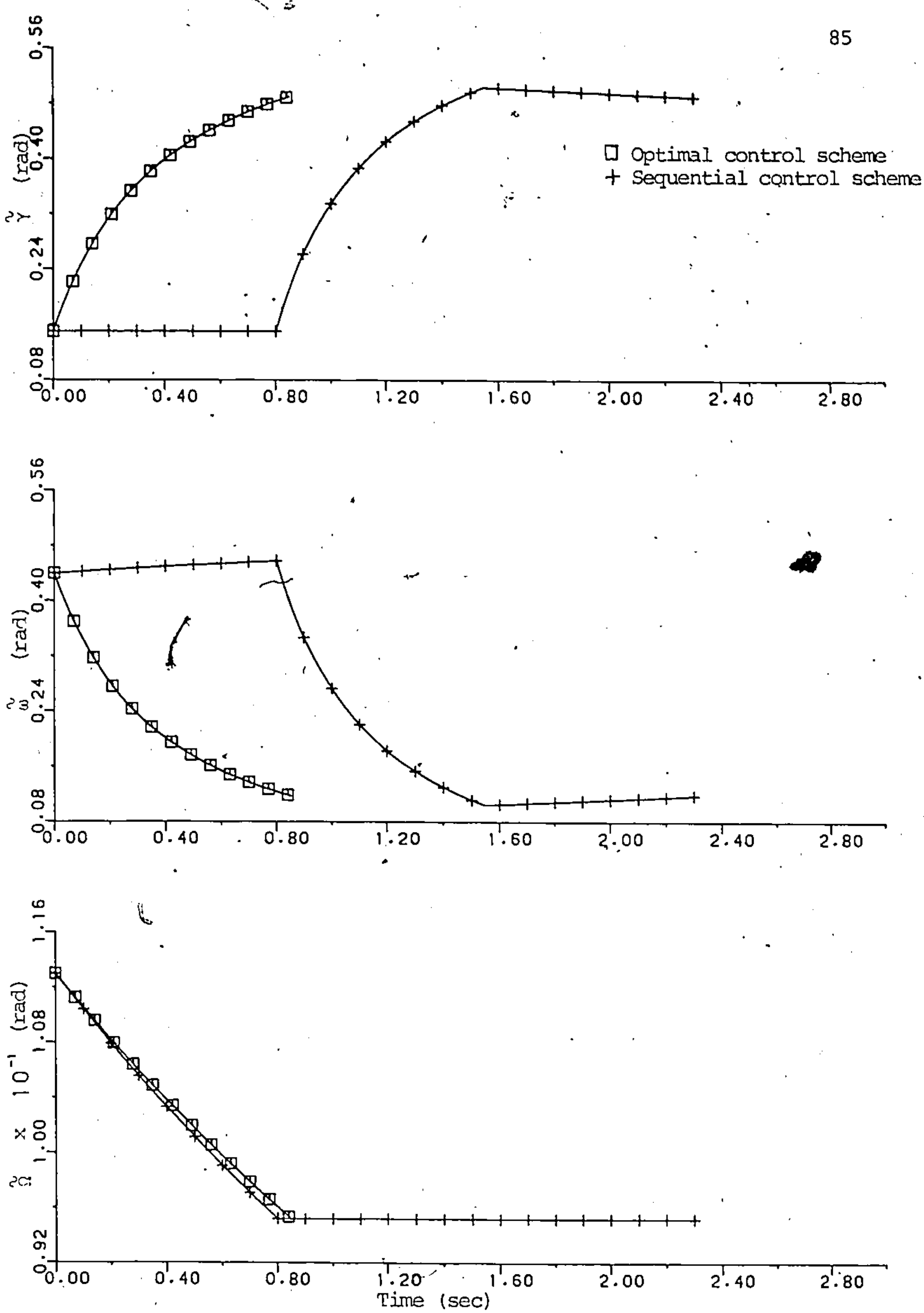


Figure 31: Case 2: Error response of optimal and sequential schemes
$\tilde{\Omega}(o)$=o.113 rad, $\tilde{\omega}(o)$=0.44 rad, $\tilde{\gamma}(o)$=o.15 rad.

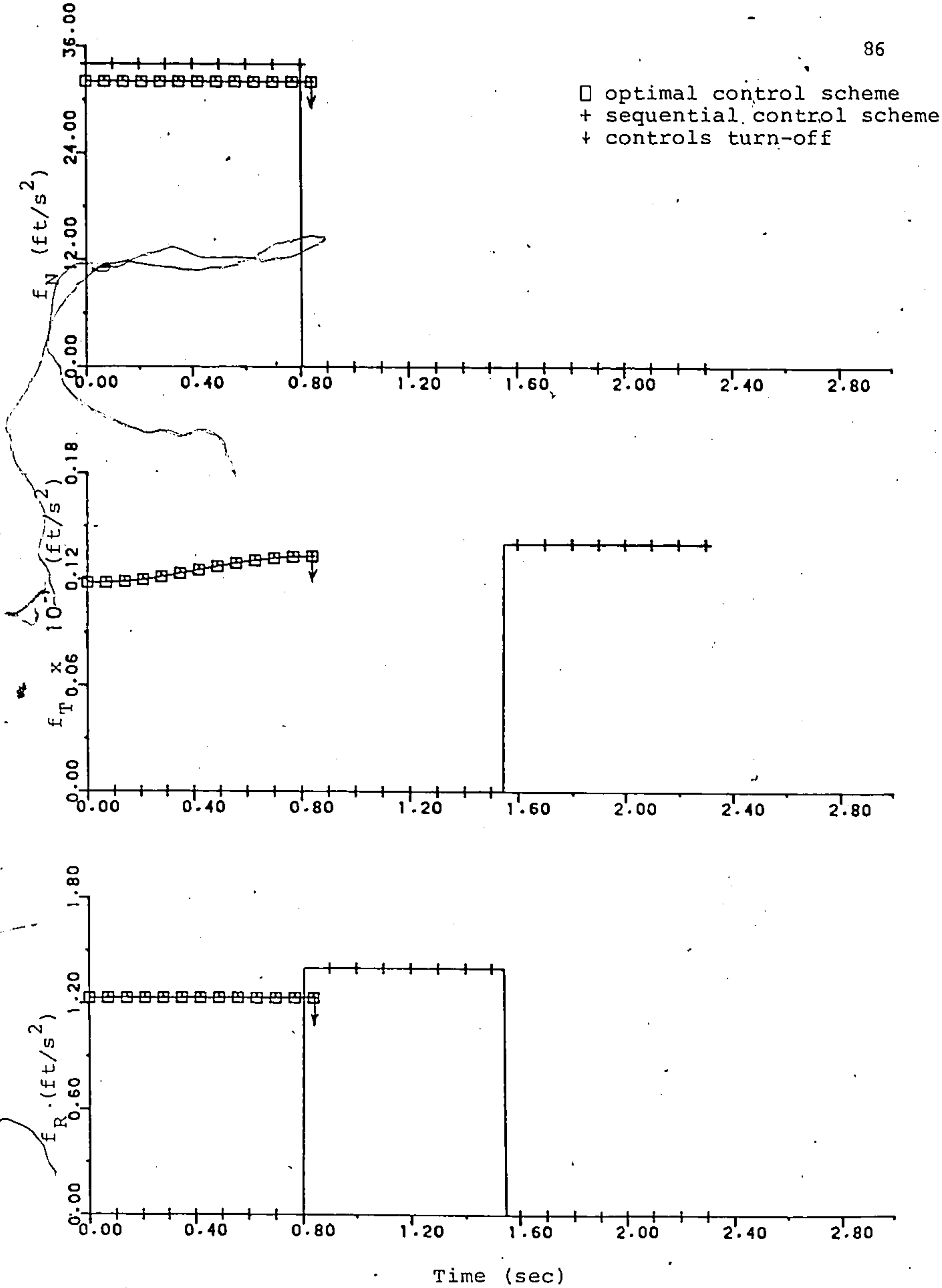


Fig.32: Case 2: Optimal and sequential control policies

4.10 CONCLUSIONS

In this chapter, the periodic orbital manoeuvre scheme is determined by integrating the dynamics system with solar radiation disturbance as the natural perturbation acting on the orbiting satellite.

An optimal orbital control problem is formulated and is solved by Conjugate Gradient Descent method to bring the system to the desired states with minimum fuel consumption. A study is made to compare the Extremal Control Scheme presented in this thesis with the Sequential Control Scheme proposed by Ahmed and Sen [16], the results show that the former control scheme is superior than the latter one.

A careful examination of the results shows that the response of the system is extremely sluggish, this suggests that the dynamic system can be approximated by a system of difference equations.

The results obtained in this thesis and [16] could be usefully applied to the design ~~of~~ the orbital control systems for the synchronous satellite.

4.11 SUGGESTIONS FOR FURTHER STUDIES

Stability and the existence of trap mode of a synchronous satellite were investigated by Ahmed and Sen [16,20]. In this thesis, the optimal orbital control of a synchronous satellite was considered.

One of the challenging problems for further investigation is to reduce the computational task. One of the reasons for the lengthy computation may be due to the nonlinearity of the system dynamics. So, in order to reduce the computational task; it is advisable to obtain an optimal control for the linearized system and then use this control policy as an initial guess to the original (nonlinear) system to obtain the optimal control policy. It would be interesting to study the difference between the solutions of the linearized and the original system.

As mentioned earlier, it is difficult to drive all the orbital parameters to their desirable states, so it would be an interesting problem to study the controllability of the system. However, it is difficult to study the controllability of a nonlinear system and it is advisable to start with the linearised system.

It appears to the author that most of the satellite control problems were concentrated on attitude control and few works were done on the optimal orbital control problems;

so it would be a contribution to consider the time optimal control and other control problems in this area.

Chapter V

CONCLUSIONS

In this thesis, Optimal Control Theory has been extensively applied to the studies of two aerospace problems : radar scanning problem and orbital control problem. Two computational algorithms are employed for the solution of the problems.

Conclusions of the control problems were presented at the end of Chapters 3 and 4 and will not be elaborated here. In this chapter, a discussion of the two computational algorithms used in this thesis and their comparisons will be presented.

The radar scanning problem composed of linear differential equations with time delay arguments, while the orbital control problem is described by a set of nonlinear differential equations without delayed terms.

Most commonly used computational methods are not suitable to compute the optimal control for systems with time delayed arguments, so a strong variational (Murray and Teo's) algorithm was chosen to solve the problem.. The radar scanning problem was successfully solved by this algorithm. This algorithm was also adopted to solve the orbital control problem; however this technique was not satisfactory since the convergence was extremely slow thus require enormous computing time. Alternate techniques were sought and the Conjugate Gradient Descent (CGD) method was decided for its outstanding fast convergence and simplicity.

The approach of the two algorithms are quite different. The Murray and Teo's algorithm modifies the control in the interval where the change in the hamiltonian function is maximum. The CGD algorithm changes the controls on the basis of the past and present gradients (which in fact are the derivatives of the hamiltonian with respect to the control variables).

As a comparison of the two algorithms, the CGD algorithm has the advantage of requiring less computer storage than the Murray and Teo's algorithm. The latter algorithm requires considerably larger storage memory to store the states ($x \in R^m$), costates ($\psi \in R^m$), old controls ($u^i \in R^r$) and updated controls ($u^{i+1} \in R^r$) for the time interval $[t_0, t_f]$. However, in the CGD algorithm, it is required to store the states ($x \in R^m$), gradients ($g(u) \in R^r$), old control ($u^i \in R^r$), and updated controls ($u^{i+1} \in R^r$). In CGD algorithm, it is not required to store the costate variables since the gradients can be computed directly after solving $\dot{\psi}$ and the values of the costate variables are not required in future calculations. Unlike the CGD algorithm, in Murray and Teo's algorithm, the costate variables are required for the maximization of the hamiltonian over the entire interval. Owing to this difference, considerable amount of memory can be saved by using CGD algorithm especially when the dimension of the states is much larger than that of the controls. Besides, the CGD algorithm is less complex and hence it is easier to implement.

The pitfall of the CGD algorithm is that it cannot guarantee global minima. As pointed out in [8], like most other iterative procedures, this method cannot distinguish between local and global minima. However, in general, 'the best that can be achieved is efficient convergence to the bottom of whatever valley it starts in. As for the Murray and Teo's algorithm, the weakness lies in one of the stopping criteria. The computation of the optimal control will be terminated when the length of the sub-interval, [b(m),b(m+1)] is less than the integration step size, h (see algorithm flow chart in Figure 6). Owing to this criterion, global minima cannot be ensured. One way of avoiding the occurrence of this undesired stopping criterion is to reduce the integration step size. However, this will inevitably increase the storage and computing time.

The convergence of the CGD algorithm is well known. For a linear quadratic problem, it converges in a finite number of iterations. However, the convergence of a nonlinear problem is quite slow, despite large improvement of cost in the first few iterations. On the other hand, the convergence of Murray and Teo's algorithm is acceptably fast for the linear case; however informations on the convergence of nonlinear problems are not available. It is the feeling of the author that a well defined convergence criterion is an important aspect of the algorithm.

Based on the experience accumulated from the computation of optimal control problems with different algorithms, it is the feeling of the author that the computational aspect of the optimal control problems is still in its infancy. There is no sure way of choosing an algorithm to solve a problem and there is no algorithm that suits all problems. The computational aspects is also a good area to pursue further investigations.

Afterall, Optimal Control Theory had been sucessfully applied to two aerospace problems. The author believes that the theory can be applied to numerous problems.

Appendix A

ADJOINT EQUATIONS AND HAMILTONIAN FUNCTION FOR THE RADAR PROBLEM

By following equations (2.4) and (2.5), the adjoint (costate) equation corresponding to the system dynamics (3.26) is given by the following system of equations.

$$\dot{\psi}_1(t) = -2\gamma_1(1-x_1)+\beta_1 u_1 \psi_1 + \alpha_1(1-u_1)\psi_1$$
$$-\alpha_4(1-u_4)P_4(a)\psi_4(t+T_o)-\alpha_5(1-u_5)0.5P_5(\bar{a})\psi_5(t+T_o)$$

$$\dot{\psi}_2(t) = -2\gamma_2(1-x_2)+\beta_2 u_2 \psi_2 + \alpha_2(1-u_2)\psi_2$$
$$-\alpha_4(1-u_4)0.5P_4(\bar{a})\psi_4(t+T_o)-\alpha_5(1-u_5)P_5(a)\psi_5(t+T_o)$$
$$-\alpha_6(1-u_6)0.5P_6(\bar{a})\psi_6(t+T_o)$$

$$\dot{\psi}_3(t) = -2\gamma_3(1-x_3)+\beta_3 u_3 \psi_3 + \alpha_3(1-u_3)\psi_3$$
$$-\alpha_5(1-u_5)0.5P_5(\bar{a})\psi_5(t+T_o)-\alpha_6(1-u_6)P_6(a)\psi_6(t+T_o)$$

$$\dot{\psi}_4(t) = -2\gamma_4(1-x_4)+[\beta_4 u_4+\alpha_4(1-u_4)]\psi_4$$
$$-\alpha_7(1-u_7)P_7(a)\psi_7(t+T_o)$$
$$-\alpha_8(1-u_8)0.5P_8(\bar{a})\psi_8(t+T_o)$$

$$\dot{\psi}_5(t) = -2\,\gamma_5(1-x_5)+[\beta_5 u_5+\alpha_5(1-u_5)]\,\psi_5$$

$$-\,\alpha_7(1-u_7)\,0.5P_7(\bar{a})\,\psi_7(t+T_o)$$

$$-\,\alpha_8(1-u_8)\,P_8(a)\,\psi_8(t+T_o)$$

$$-\,\alpha_9(1-u_9)\,0.5P_9(\bar{a})\,\psi_9(t+T_o)$$

$$\dot{\psi}_6(t) = -2\,\gamma_6(1-x_6)+[\beta_6 u_6+\alpha_6(1-u_6)]\,\psi_6$$

$$-\,\alpha_8(1-u_8)\,0.5P_8(\bar{a})\,\psi_8(t+T_o)$$

$$-\,\alpha_9(1-u_9)\,P_9(a)\,\psi_9(t+T_o)$$

$$\dot{\psi}_7(t) = -2\,\gamma_7(1-x_7)+[\beta_7 u_7+\alpha_7(1-u_7)]\,\psi_7$$

$$\dot{\psi}_8(t) = -2\,\gamma_8(1-x_8)+[\beta_8 u_8+\alpha_8(1-u_8)]\,\psi_8$$

$$\dot{\psi}_9(t) = -2\,\gamma_9(1-x_9)+[\beta_9 u_9+\alpha_9(1-u_9)]\,\psi_9$$

with final conditions

$$\psi_i(t) = -\frac{\partial}{\partial x_i}\left[\,\lambda_i(1-x_i(t_f))^2\,\right] \quad \text{for } t \geq t_f$$

$$\psi_1(t) = 2\,\lambda_1\,(1-x_1(t_f))$$

$$\psi_2(t) = 2\,\lambda_2\,(1-x_2(t_f))$$

$$\psi_3(t) = 2\,\lambda_3\,(1-x_3(t_f))$$

$$\vdots$$

$$\psi_9(t) = 2\,\lambda_9\,(1-x_9(t_f)) \qquad \text{for } t \geq t_f.$$

Hamiltonian Function for the Radar Problem

The following hamiltonian function follows from equation (2.6).

$$
\begin{aligned}
H(t,x,v,\Psi) = {} & u_1[-\sigma_1-\beta_1 x_1\psi_1+\beta_1\psi_1+\alpha_1 x_1\psi_1] \\
& + u_2[-\sigma_2-\beta_2 x_2\psi_2+\beta_2\psi_2+\alpha_2 x_2\psi_2] \\
& + u_3[-\sigma_3-\beta_3 x_3\psi_3+\beta_3\psi_3+\alpha_3 x_3\psi_3] \\
& + u_4[-\sigma_4-\beta_4(x_4-1)\psi_4+\alpha_4 x_4\psi_4 \\
& \quad -\alpha_4\{P_4(a)x_1(t-T_o)+0.5P_4(\bar{a})x_2(t-T_o)\}\psi_4(t+T_o)] \\
& + u_5[-\sigma_5-\beta_5(x_5-1)\psi_5+\alpha_5 x_5\psi_5 \\
& \quad -\alpha_5\{P_5(a)x_2(t-T_o)+0.5P_5(\bar{a})x_1(t-T_o)+0.5P_5(\bar{a})x_3(t-T_o)\} \\
& \quad \psi_5(t+T_o)] \\
& + u_6[-\sigma_6-\beta_6(x_6-1)\psi_6+\alpha_6 x_6\psi_6 \\
& \quad -\alpha_6\{P_6(a)x_3(t-T_o)+0.5P_6(\bar{a})x_2(t-T_o)\}\psi_6(t+T_o)] \\
& + u_7[-\sigma_7-\beta_7(x_7-1)\psi_7+\alpha_7 x_7\psi_7 \\
& \quad -\alpha_7\{P_7(a)x_4(t-T_o)+0.5P_7(\bar{a})x_5(t-T_o)\}\psi_7(t+T_o)] \\
& + u_8[-\sigma_8-\beta_8(x_8-1)\psi_8+\alpha_8 x_8\psi_8 \\
& \quad -\alpha_8\{P_8(a)x_5(t-T_o)+0.5P_8(\bar{a})x_4(t-T_o) \\
& \quad +0.5P_8(\bar{a})x_6(t-T_o)\}\psi_8(t+T_o)]
\end{aligned}
$$

$$+ u_9[-\sigma_9-\beta_9(x_9-1)\psi_9+\alpha_9 x_9\psi_9$$

$$-\alpha_9\{P_9(a)x_6(t-T_o)+0.5P_9(\bar{a})x_5(t-T_o)\}\psi_9(t+T_o)]$$

$$+ [-\gamma_1(1-x_1)^2-\gamma_2(1-x_2)^2-\gamma_3(1-x_3)^2-\gamma_4(1-x_4)^2$$

$$-\gamma_5(1-x_5)^2-\gamma_6(1-x_6)^2-\gamma_7(1-x_7)^2-\gamma_8(1-x_8)^2-\gamma_9(1-x_9)^2]$$

$$+ [-\alpha_1 x_1\psi_1-\alpha_2 x_2\psi_2-\alpha_3 x_3\psi_3-\alpha_4 x_4\psi_4$$

$$-\alpha_5 x_5\psi_5-\alpha_6 x_6\psi_6-\alpha_7 x_7\psi_7-\alpha_8 x_8\psi_8-\alpha_9 x_9\psi_9]$$

$$+ \alpha_4[P_4(a)x_1(t-T_o)+0.5P_4(\bar{a})x_2(t-T_o)]\psi_4(t+T_o)$$

$$+\alpha_5[P_5(a)x_2(t-T_o)+0.5P_5(\bar{a})x_1(t-T_o)+0.5P_5(\bar{a})x_3(t-T_o)]\psi_5(t+T_o)$$

$$+\alpha_6[P_6(a)x_3(t-T_o)+0.5P_6(\bar{a})x_2(t-T_o)]\psi_6(t+T_o)$$

$$+\alpha_7[P_7(a)x_4(t-T_o)+0.5P_7(\bar{a})x_5(t-T_o)]\psi_7(t+T_o)$$

$$+\alpha_8[P_8(a)x_5(t-T_o)+0.5P_8(\bar{a})x_4(t-T_o)+0.5P_8(\bar{a})x_6(t-T_o)]\psi_8(t+T_o)$$

$$+\alpha_9[P_9(a)x_6(t-T_o)+0.5P_9(\bar{a})x_5(t-T_o)]\psi_9(t+T_o).$$

Appendix B

ADJOINT EQUATIONS AND HAMILTONIAN FUNCTION FOR THE SATELLITE PROBLEM

The adjoint equations and the hamiltonian function for the dynamics system without delay terms can be obtained from equations (2.4) and (2.5) by setting s=0 (i.e. ignoring the terms due to the delays). Due to the nonlinearity of the dynamics equations, the adjoint equations are quite lengthy.

$$\dot{\psi}_1(t) = -\frac{\partial f_o}{\partial \tilde{a}}\psi_o - \frac{\partial f_1}{\partial \tilde{a}}\psi_1 - \frac{\partial f_2}{\partial \tilde{a}}\psi_2 - \frac{\partial f_3}{\partial \tilde{a}}\psi_3 - \frac{\partial f_4}{\partial \tilde{a}}\psi_4 - \frac{\partial f_5}{\partial \tilde{a}}\psi_5 - \frac{\partial f_6}{\partial \tilde{a}}\psi_6$$

where,

$$\frac{\partial f_o}{\partial \tilde{a}} = 2\sigma_1\, \tilde{a}$$

$$\frac{\partial f_1}{\partial \tilde{a}} = \frac{(1+e\cos\gamma)f_T + e\sin\gamma f_R}{\mu a(1-e^2)}\left[4a\sqrt{\mu a(1-e^2)} - \frac{\mu a^2(1-e^2)}{\sqrt{\mu a(1-e^2)}}\right]$$

$$\frac{\partial f_2}{\partial \tilde{a}} = \frac{1}{2}\sqrt{\frac{\mu}{a(1-e^2)}}\,\frac{1-e^2}{\mu}\left[\frac{2\cos\gamma+e(1+\cos^2\gamma)}{1+e\cos\gamma} f_T + \sin\gamma f_R\right]$$

$$\frac{\partial f_3}{\partial \tilde{a}} = \frac{1}{2}\sqrt{\frac{\mu}{a(1-e^2)}}\,\frac{1-e^2}{\mu}\,\frac{\cos(\omega+\gamma)}{1+e\cos\gamma} f_N$$

$$\frac{\partial f_4}{\partial \tilde{a}} = \frac{1}{2}\sqrt{\frac{\mu}{a(1-e^2)}}\,\frac{1-e^2}{\mu}\,\frac{\sin(\omega+\gamma)}{\sin i(1+e\cos\gamma)} f_N$$

$$\frac{\partial f_5}{\partial \tilde{a}} = \frac{1}{2e}\sqrt{\frac{\mu}{a(1-e^2)}}\,\frac{1-e^2}{\mu}\left[\frac{2+e\cos\gamma}{1+e\cos\gamma}\sin\gamma f_T - \cos\gamma f_R - \frac{e\cot i\sin(\omega+\gamma)f_N}{1+e\cos\gamma}\right]$$

$$\frac{\partial f_6}{\partial \tilde{a}} = \frac{1}{2e}\sqrt{\frac{\mu}{a(1-e^2)}}\,\frac{1-e^2}{\mu}\left[-\frac{2+e\cos\gamma}{1+e\cos\gamma}\sin\gamma f_T + \cos\gamma f_R\right]$$

$$\dot{\psi}_2(t) = -\frac{\partial f_o}{\partial \tilde{e}}\psi_o - \frac{\partial f_1}{\partial \tilde{e}}\psi_1 - \dots - \frac{\partial f_6}{\partial \tilde{e}}\psi_6$$

where,

$$\frac{\partial f_o}{\partial \tilde{e}} = 2\,\sigma_2\,\tilde{e}$$

$$\frac{\partial f_1}{\partial \tilde{e}} = [(1+e\cos\gamma)f_T + e\sin\gamma f_R]\frac{2\mu a^3 e}{[\mu a(1-e^2)]^{3/2}} + \frac{2a^2(\cos\gamma f_T+\sin\gamma f_R)}{\sqrt{\mu a(1-e^2)}}$$

$$\frac{\partial f_2}{\partial \tilde{e}} = -\frac{ae}{\sqrt{\mu a(1-e^2)}}\left[\frac{2\cos\gamma + e(1-\cos^2\gamma)}{1+e\cos\gamma}f_T + \sin\gamma\, f_R\right]$$

$$+ f_T\sqrt{\frac{a(1-e^2)}{\mu}}\left[\frac{(1+e\cos\gamma)(1+\cos^2\gamma) - [2\cos\gamma+e(1+\cos^2\gamma)]\cos\gamma}{(1+e\cos\gamma)^2}\right]$$

$$\frac{\partial f_3}{\partial \tilde{e}} = -f_N\left[\frac{\cos(\omega+\gamma)}{1+e\cos\gamma}\sqrt{\frac{\mu}{a(1-e^2)}}\frac{ae}{\mu} + \sqrt{\frac{a(1-e^2)}{\mu}}\frac{\cos(\omega+\gamma)\cos\gamma}{(1+e\cos\gamma)^2}\right]$$

$$\frac{\partial f_4}{\partial \tilde{e}} = -f_N\left[\frac{\sin(\omega+\gamma)}{\sin i(1+e\cos\gamma}\sqrt{\frac{\mu}{a(1-e^2)}}\frac{ae}{\mu} + \sqrt{\frac{a(1-e^2)}{\mu}}\frac{\sin(\omega+\gamma)\cos\gamma}{\sin i(1+e\cos\gamma)^2}\right]$$

$$\frac{\partial f_5}{\partial \tilde{e}} = -\left[\frac{2+e\cos\gamma}{1+e\cos\gamma}\sin\gamma f_T - \cos\gamma f_R - \frac{e\cot i\sin(\omega+\gamma)f_N}{1+e\cos\gamma}\sqrt{\frac{\mu e^2}{a(1-e^2)}}\right]\cdot\frac{e^2 a + a(1-e^2)}{e^3\mu}$$

$$+ \frac{1}{e}\sqrt{\frac{a(1-e^2)}{\mu}}\left[\sin\gamma f_T\frac{(1+e\cos\gamma)\cos\gamma-(2+e\cos\gamma)\cos\gamma}{(1+e\cos\gamma)^2} - \frac{\cot i\sin(\omega+\gamma)f_N}{(1+e\cos\gamma)^2}\right]$$

$$\frac{\partial f_6}{\partial \tilde{e}} = \left[\frac{2+e\cos\gamma}{1+e\cos\gamma}\sin\gamma f_T + \cos\gamma f_R\right]\sqrt{\frac{e^2\mu}{a(1-e^2)}}\frac{a(e^2 + (1-e^2))}{e^3\mu}$$

$$-\frac{1}{e}\sqrt{\frac{a(1-e^2)}{\mu}}\sin\gamma f_T\frac{(1+e\cos\gamma)\cos\gamma - (2+e\cos\gamma)\cos\gamma}{(1+e\cos\gamma)^2}$$

$$\dot{\psi}_3(t) = -\frac{\partial f_o}{\partial \tilde{i}}\psi_o - \frac{\partial f_1}{\partial \tilde{i}}\psi_1 - \ldots - \frac{\partial f_6}{\partial \tilde{i}}\psi_6$$

where

$$\frac{\partial f_o}{\partial \tilde{i}} = 2\ \sigma_3\ \tilde{i}$$

$$\frac{\partial f_4}{\partial \tilde{i}} = -f_N\sqrt{\frac{a(1-e^2)}{\mu}}\ \frac{\sin(\omega+\gamma)}{1+e\ \cos\gamma}\ \frac{\cos i}{\sin^2 i}$$

$$\frac{\partial f_5}{\partial \tilde{i}} = f_N\sqrt{\frac{a(1-e^2)}{\mu}}\ \frac{\sin(\omega+\gamma)\mathrm{cosec}^2 i}{1+e\ \cos\gamma}$$

$$\frac{\partial f_1}{\partial \tilde{i}} = \frac{\partial f_2}{\partial \tilde{i}} = \frac{\partial f_3}{\partial \tilde{i}} = \frac{\partial f_6}{\partial \tilde{i}} = 0$$

$$\dot{\psi}_4(t) = \frac{\partial f_o}{\partial \tilde{\Omega}}\psi_o - \frac{\partial f_1}{\partial \tilde{\Omega}}\psi_1 - \ldots - \frac{\partial f_6}{\partial \tilde{\Omega}}\psi_6$$

where

$$\frac{\partial f_o}{\partial \tilde{\Omega}} = 2\ \sigma_4\ \tilde{\Omega}$$

$$\frac{\partial f_1}{\partial \tilde{\Omega}} = \frac{\partial f_2}{\partial \tilde{\Omega}} = \frac{\partial f_3}{\partial \tilde{\Omega}} = \frac{\partial f_4}{\partial \tilde{\Omega}} = \frac{\partial f_5}{\partial \tilde{\Omega}} = \frac{\partial f_6}{\partial \tilde{\Omega}} = 0.$$

$$\dot{\psi}_5(t) = -\frac{\partial f_0}{\partial\tilde{\omega}}\psi_0 - \frac{\partial f_1}{\partial\tilde{\omega}}\psi_1 - \cdots - \frac{\partial f_6}{\partial\tilde{\omega}}\psi_6$$

where,

$$\frac{\partial f_0}{\partial\tilde{\omega}} = 2\,\sigma_5\,\tilde{\omega}$$

$$\frac{\partial f_3}{\partial\tilde{\omega}} = -f_N\sqrt{\frac{a(1-e^2)}{\mu}}\,\frac{\sin(\omega+\gamma)}{1+e\cos\gamma}$$

$$\frac{\partial f_4}{\partial\tilde{\omega}} = f_N\sqrt{\frac{a(1-e^2)}{\mu}}\,\frac{\cos(\omega+\gamma)}{\sin i(1+e\cos\gamma)}$$

$$\frac{\partial f_5}{\partial\tilde{\omega}} = -f_N\sqrt{\frac{a(1-e^2)}{\mu}}\,\frac{\cot i\cos(\omega+\gamma)}{1+e\cos\gamma}$$

$$\frac{\partial f_1}{\partial\tilde{\omega}} = \frac{\partial f_2}{\partial\tilde{\omega}} = \frac{\partial f_6}{\partial\tilde{\omega}} = 0$$

$$\dot{\psi}_6(t) = -\frac{\partial f_0}{\partial\tilde{\gamma}}\psi_0 - \frac{\partial f_1}{\partial\tilde{\gamma}}\psi_1 - \cdots - \frac{\partial f_6}{\partial\tilde{\gamma}}\psi_6$$

where,

$$\frac{\partial f_0}{\partial\tilde{\gamma}} = 2\,\sigma_6\,\tilde{\gamma}$$

$$\frac{\partial f_1}{\partial\tilde{\gamma}} = \frac{2a^2e}{\sqrt{\mu a(1-e^2)}}\,(\cos\gamma f_R - \sin\gamma f_T)$$

$$\frac{\partial f_2}{\partial\tilde{\gamma}} = \sqrt{\frac{a(1-e^2)}{\mu}}\,[f_T\cdot\frac{-(1+e\cos\gamma)(2\sin\gamma-e\sin2\gamma)+e\sin2\gamma+e^2\sin\gamma(1+\cos^2\gamma)}{(1+e\cos\gamma)^2}$$

$$+\cos\gamma f_R]$$

$$\frac{\partial f_3}{\partial\tilde{\gamma}} = f_N\sqrt{\frac{a(1-e^2)}{\mu}}\,\frac{e\sin\gamma\cos(\omega+\gamma) - (1+e\cos\gamma)\sin(\omega+\gamma)}{(1+e\cos\gamma)^2}$$

POOR COPY
COPIE DE QUALITEE INFERIEURE

$$\frac{\partial f_4}{\partial \tilde{\gamma}} = f_N \sqrt{\frac{a(1-e^2)}{\mu}} \frac{(1+e\cos\gamma)\cos(\omega+\gamma) + e\sin\gamma\sin(\omega+\gamma)}{\sin i\,(1+e\cos\gamma)^2}$$

$$\frac{\partial f_5}{\partial \tilde{\gamma}} = \frac{1}{e} \sqrt{\frac{a(1-e^2)}{\mu}} [f_T \cdot \frac{(1+e\cos\gamma)(2\cos\gamma + e\cos 2\gamma) + e(2+e\cos\gamma)\sin^2\gamma}{(1+e\cos\gamma)^2}$$

$$+ f_R \sin\gamma - ef_N \cot i \frac{(1+e\cos\gamma)\cos(\omega+\gamma) + e\sin(\omega+\gamma)\sin\gamma}{(1+e\cos\gamma)^2}]$$

$$\frac{\partial f_o}{\partial \tilde{\gamma}} = \frac{1}{e} \sqrt{\frac{a(1-e^2)}{\mu}} [-f_T \frac{(1+e\cos\gamma)(2\cos\gamma + e\cos 2\gamma) + e(2+e\cos\gamma)\sin^2\gamma}{(1+e\cos\gamma)^2}$$

$$- f_R \sin\gamma]$$

The explicit expression of the Hamiltonian function is

$$H(t,x,v,\Psi) = \sigma_1|\tilde{a}|^2 + \ldots + \sigma_6|\tilde{\gamma}|^2 + \lambda_1 f_R^2 + \lambda_2 f_T^2 + \lambda_3 f_N^2$$

$$+ \frac{2a^2}{\sqrt{\mu a(1-e^2)}} [(1+e\cos\gamma) f_T + ef_R \sin\gamma]\psi_1$$

$$+ \sqrt{\frac{a(1-e^2)}{\mu}} [\frac{2\cos\gamma + e(1+\cos^2\gamma)}{1+e\cos\gamma} f_T + f_R \sin\gamma]\ \psi_2$$

$$+ \sqrt{\frac{a(1-e^2)}{\mu}} \frac{\cos(\omega+\gamma)}{1+e\cos\gamma} f_N\ \psi_3$$

$$+ \sqrt{\frac{a(1-e^2)}{\mu}} \frac{\sin(\omega+\gamma)}{\sin i(1+e\cos\gamma)} f_N\ \psi_4$$

$$+ \frac{1}{e}\sqrt{\frac{a(1-e^2)}{\mu}} [\frac{2+e\cos\gamma}{1+e\cos\gamma} f_T \sin\gamma - f_R\cos\gamma - \frac{ef_N \cot i \sin(\omega+\gamma)}{1+e\cos\gamma}]\ \psi_5$$

$$+ \frac{1}{e}\sqrt{\frac{a(1-e^2)}{\mu}} [-\frac{2+e\cos\gamma}{1+e\cos\gamma} f_T \sin\gamma + f_R\cos\gamma]\ \psi_6 .$$

Appendix C

COMPUTER PROGRAM FOR THE SATELLITE PROBLEM

```
C     FIRST EDITION : AUGUST 17, 1981
C     REVISED       : DEC.12,1981
C              OPTIMIZATION BY CONGUGATE GRADIENT DESCENT METHOD
C                  ( *** SATELLITE PROBLEM *** )
C     PROGRAM NAME  : CGD
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL *8 MU,IO,ID,IDO,J1,J2,J3,J4,J5,J6,L1,L2,L3
      INTEGER EXIT
      DIMENSION CTL(1002,3),SRH(1002,3),GRDO(1002,3),GRDN(1002,3),
     &     TT(1002),X(1002,6),BTA(3),G(1002,3)
      COMMON / SIZE / H
      COMMON / XSTATE / ADO,EDO,IDO,ODO,WDO,RDO
      COMMON / SCBLK / MU,AO,EO,IO,OO,WO,RO
      COMMON / WC / J1,J2,J3,J4,J5,J6,L1,L2,L3
      COMMON / SS / S1,S2,S3,S4,S5,S6
      COMMON / FWC / B1,B2,B3,B4,B5,B6
      COMMON / BBPF / BB1,BB2,BB3,BB4,BB5,BB6
      COMMON / DCTL / FRO,FTO,FNO
      COMMON / ALPHA / AAO
      CALL INPUT ( TT,CTL,KOUNT )
C*****INITIALIZATION
      EXIT = 0
      NEXIT = 190
      WRITE (6,11) NEXIT
11    FORMAT (T7,'MAX  OF ITERATION :',I5)
C*****JI:1 FOR BBPF, JI:0 FOR NO BBPF (BASIC BIASED PENALTY FUNCTION)
      JI = 0
      BB1 = 0.
      BB2 = 0.
      BB3 = 0.
      BB4 = 0.
      BB5 = 0.
      BB6 = 0.
C*****SOLVE STATE EQUATION FORWARD IN TIME
      CALL STATE ( CTL,KOUNT, X )
      CALL COST ( X,CTL,H,KOUNT, DUMMY,COSTOD )
C*****SOLVE COSTATE EQN
      CALL ADJ ( X,CTL,KOUNT, G)
C*****STORE GRADIENT
      CALL STOGRD ( G,KOUNT, GRDO )
C*****INITIALIZE SEARCH DIRECTION
      DO 80 J = 1,3
      DO 80 I = 1,KOUNT
      SRH(I,J) = - G(I,J)
```

```
80     CONTINUE
210    WRITE (6,510) EXIT,COSTOD
       WRITE (8,510) EXIT,COSTOD
510    FORMAT (T5,'EXIT=',I3,3X,'COST = ',E19.12)
       EXIT = EXIT + 1
C*****FIND OPTIMAL STEP SIZE THAT IMPROVES THE COST FUNCTIONAL
       CALL MINMIZ ( X,CTL,SRH,COSTOD,H,KOUNT, STEP )
C*****COMPUTE U(I+1)
C      DO 24 I=1,KOUNT,30
C      WRITE (6,27) CTL(I,1),CTL(I,2),CTL(I,3)
C27     FORMAT (T7,'CTL BEFORE=',3(E19.12,1X))
C24     CONTINUE
       CALL NEWCTL ( KOUNT,SRH,STEP, CTL )
C      DO 33 I=1,KOUNT,30
C      WRITE (6,34) CTL(I,1),CTL(I,2),CTL(I,3)
C34     FORMAT (T7,'CTL AFTER=',3(E19.12,1X))
C33     CONTINUE
C*****SOLVE STATE EQN
       CALL STATE ( CTL,KOUNT, X )
       CALL COST ( X,CTL,H,KOUNT, SUM,COST$ )
C*****SOLVE COSTATE EQN
       CALL ADJ ( X,CTL,KOUNT, G )
C      DO 31 I = 1,KOUNT,30
C      WRITE (6,32) TT(I),G(I,1),G(I,2),G(I,3)
C32     FORMAT (F7.4,1X,3(E19.12,1X))
C31     CONTINUE
C*****STORE GRADIENT G(U)
       CALL STOGRD ( G,KOUNT, GRDN )
C*****COMPUTE THE UNIT VECTOR
       CALL UNIVEC ( GRDO,GRDN,H,KOUNT, BTA )
       WRITE (6,71) BTA(1),BTA(2),BTA(3)
71     FORMAT (T7,'BETA =',3(E19.12,1X))
C*****COMPUTE SEARCH DIRECTION S(I+1)
       CALL SEARCH ( GRDN,BTA,KOUNT, SRH )
C      DO 75 I=1,KOUNT,30
C      WRITE (6,76) SRH(I,1),SRH(I,2),SRH(I,3)
C76     FORMAT (T7,'SEARCH =',3(E19.12,1X))
C75     CONTINUE
C*****STORE U(I+1) AS U(I)
       CALL EXGRD ( KOUNT,GRDN, GRDO )
C*****CHECK STOPPING CRITERION
       BALL = ( X(KOUNT,1)**2 + X(KOUNT,2)**2 + X(KOUNT,3)**2 )**0.5
       WRITE (6,1729) BALL
1729   FORMAT ( T7,'STATE SPACE BALL :',E19.12 )
       IF ( BALL.LE.1.0E-6 ) GO TO 9999
       COSDIF = DABS(COST$-COSTOD)
       WRITE (6,1738) COSDIF
1738   FORMAT ( T7,'COST DIFFERENCE :',E19.12 )
       IF ( COSDIF.LE.1E-5 )  GO TO 9999
       IF (EXIT.GT.NEXIT) GO TO 9999
C*****UPDATE PENALTY WEIGHT BY BASIC BIASED PENALTY FUNCTION APPROACH
       BB1 = JI*( BB1 + DABS(X(KOUNT,1)) )
       BB2 = JI*( BB2 + DABS(X(KOUNT,2)) )
       BB3 = JI*( BB3 + DABS(X(KOUNT,3)) )
```

```
      BB4 = JI*( BB4 + DABS(X(KOUNT,4)) )
      BB5 = JI*( BB5 + DABS(X(KOUNT,5)) )
      BB6 = JI*( BB6 + DABS(X(KOUNT,6)) )
      COSTOD = SUM+B1*( X(KOUNT,1)+BB1 )**2 + B2*( X(KOUNT,2)+BB2 )**2
     &   B3*( X(KOUNT,3)+BB3 )**2 + B4*( X(KOUNT,4)+BB4 )**2
     &   B4*( X(KOUNT,5)+BB5 )**2 + B6*( X(KOUNT,6)+BB6 )**2
      WRITE (6,606) X(KOUNT,1),X(KOUNT,2),X(KOUNT,3)
606   FORMAT ( T7,'X AT TF : ',3(T8,E19.12,/) )
      GO TO 210
9999  CALL OUTPUT ( TT,X,CTL,COSTOD,EXIT,G,KOUNT )
      WRITE (6,234) BALL, COSDIF
234   FORMAT (T7,'BALL =',E19.12,/,T7,'COST DIFF =',E19.12)
      WRITE (6,77) BB1,BB2,BB3,BB4,BB5,BB6
77    FORMAT ( T7,'BB WEIGHTING : ',/,6(T8,E16.8,/) )
      STOP
      END
      SUBROUTINE STATE ( CTL,KOUNT, X )
C
C     THIS SUBROUTINE COMPUTES THE STATE EQNS FORWARD IN TIME
C     X    : MATRIX STORING THE STATES
C     U'S  : CONTROLS
C     RKGT: SUBROUTINE TO SOLVE D.E.
C     COST: COMPUTE J(U)
C
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 IDO
      DIMENSION X(1002,6),CTL(1002,3),TT(1002)
      COMMON / XSTATE / ADO,EDO,IDO,ODO,WDO,RDO
      COMMON / SIZE / H
      INC=1
C*****STORE ALL STATES VARIABLES IN MATRIX X
      X(INC,1) = ADO
      X(INC,2) = EDO
      X(INC,3) = IDO
      X(INC,4) = ODO
      X(INC,5) = WDO
      X(INC,6) = RDO
      X1 = ADO
      X2 = EDO
      X3 = IDO
      X4 = ODO
      X5 = WDO
      X6 = RDO
100   U1 = CTL(INC,1)
      U2 = CTL(INC,2)
      U3 = CTL(INC,3)
      CALL RKGT (X1,X2,X3,X4,X5,X6,U1,U2,U3,H)
      IF ( INC.EQ.KOUNT ) GO TO 999
      INC=INC+1
      X(INC,1) = X1
      X(INC,2) = X2
      X(INC,3) = X3
      X(INC,4) = X4
      X(INC,5) = X5
```

```
      X(INC,6) = X6
      GO TO 100
999   RETURN
      END
      SUBROUTINE ADJ ( X,CTL,KOUNT, G )
C
C     THIS SUBROUTINE COMPUTES THE COSTATE EQN BACKWARD IN TIME, AND
C     IT COMPUTES THE GRADIENT
C     S'S     : COSTATE VARIABLES
C     B'S     : WEIGHTING AT FINAL TIME
C     GRAD    : COMPUTE GRADIENT OF J(U)
C     COSTAT  : SOLVE COSTATE D.E.
C
      IMPLICIT REAL *8  (A-H,O-Z)
      DIMENSION X(1002,6),CTL(1002,3),G(1002,3)
      COMMON / FWC / B1,B2,B3,B4,B5,B6
      COMMON / SS / S1,S2,S3,S4,S5,S6
      COMMON / BBPF / BB1,BB2,BB3,BB4,BB5,BB6
C*****COMPUTE THE FINAL CONDITIONS FOR THE COSTATE EQN
      S1 = +2.*B1*  ( X(KOUNT,1)+BB1 )
      S2 = +2.*B2*  ( X(KOUNT,2)+BB2 )
      S3 = +2.*B3*  ( X(KOUNT,3)+BB3 )
      S4 = +2.*B4*  ( X(KOUNT,4)+BB4 )
      S5 = +2.*B5*  ( X(KOUNT,5)+BB5 )
      S6 = +2.*B6*  ( X(KOUNT,6)+BB6 )
      IDECP=KOUNT
120   U1 = CTL(IDECR,1)
      U2 = CTL(IDECR,2)
      U3 = CTL(IDECR,3)
      X1 = X(IDECR,1)
      X2 = X(IDECR,2)
      X3 = X(IDECR,3)
      X4 = X(IDECR,4)
      X5 = X(IDECR,5)
      X6 = X(IDECR,6)
      CALL GRAD ( X1,X2,X3,X4,X5,X6,U1,U2,U3,IDECR, G )
      CALL COSTAT (X1,X2,X3,X4,X5,X6,U1,U2,U3 )
      IDECR = IDECR - 1
      IF ( IDECR.EQ.0 ) GO TO 999
      GO TO 120
999   RETURN
      END
      SUBROUTINE STOGRD ( G,KOUNT, GRD )
C
C     IT STORE THE GRADIENT
C     GRD : VECTOR WHICH IS EITHER GRDO(U) OR GRDN(U)
C     G     : VECTOR STORING GRAD(U(I+1))
C
      IMPLICIT REAL *8  (A-H,O-Z)
      DIMENSION G(1002,3),GRD(1002,3)
      DO 10 I = 1,KOUNT
      GRD(I,1) = G(I,1)
      GRD(I,2) = G(I,2)
      GRD(I,3) = G(I,3)
```

```
10    CONTINUE
      RETURN
      END
      SUBROUTINE EXGRD ( KOUNT,GRDN, GRDO)
C
C     IT EXCHANGES THE OLD AND NEW GRADIENT
C     GRDO : VECTOR STORING GRAD(U(I))
C     GRDN : VECTOR STORING GRAD(U(I+1))
C
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION GRDO(1002,3),GRDN(1002,3)
      DO 10 I = 1,KOUNT
      GRDO(I,1)  = GRDN(I,1)
      GRDO(I,2)  = GRDN(I,2)
      GRDO(I,3)  = GRDN(I,3)
10    CONTINUE
      RETURN
      END
      SUBROUTINE UNIVEC (GRDO,GRDN,H,KOUNT, BTA)
C
C     IT COMPUTES THE UNIT STEP SEARCH VECTOR, BETA(I)
C     GRADO : GRAD(U(I))
C     GRADN : GRAD(U(I+1))
C     VALO  : VALUE OF THE INTEGRAL G(U(I))
C     VALN  : VALUE OF THE INTEGRAL G(U(I+1))
C
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION GRDO(1002,3),GRDN(1002,3),BTA(3),VALO(3),VALN(3)
      CALL INTGRL ( GRDO,H,KOUNT, VALO )
      CALL INTGRL ( GRDN,H,KOUNT, VALN )
      DO 10 I = 1,3
      BTA(I)=( VALN(1)+VALN(2)+VALN(3)  )/( VALO(1)+VALO(2)+VALO(3)  )
10    CONTINUE
      RETURN
      END
      SUBROUTINE INTGRL ( GG,H,KOUNT, VAL )
C
C     IT COMPUTES THE INTEGRAL OF THE GRADIENT
C     VAL : VALUE OF THE INTEGRAL
C
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION GG(1002,3),VAL(3),SUM(3)
      DO 10 I = 1,3
10    VAL(I)  = 0.
      KOUNT1 = KOUNT - 1
      DO 30 J = 1,3
      DO 20 K = 2,KOUNT1
      SUM(J)  =  GG(K,J)*GG(K,J)
      VAL(J)  = VAL(J)  + SUM(J)
20    CONTINUE
      VAL(J)  = VAL(J)  + 0.5*GG(1,J)*GG(1,J)
      VAL(J)  = VAL(J)  + 0.5*GG(KOUNT,J)*GG(KOUNT,J)
      VAL(J)  = VAL(J)  * H
30    CONTINUE
```

```
      RETURN
      END
      SUBROUTINE SEARCH ( GRDN,BTA,KOUNT, SRH )
C
C     IT COMPUTES THE SEARCH DIRECTION
C     SRH : VECTOR OF THE SEARCH DIRECTION
C     BTA : UNIT VECTOR OF THE SEARCH VECTOR
C
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION GRDN(1002,3),BTA(3),SRH(1002,3)
      DO 10 J = 1,3
      DO 10 I = 1,KOUNT
      SRH(I,J) = - GRDN(I,J)  + BTA(J)*SRH(I,J)
10    CONTINUE
      RETURN
      END
      SUBROUTINE MINMIZ ( X,CTL,SRH,COSTOD,H,KOUNT, STEP )
C
C     IT COMPUTES THE STEP SIZE SUCH THAT J(U(I+1)) IS MINIMIZED
C     X          : STATE VECTOR
C     CTL        : CONTROL
C     SRH        : SEARCH DIRECTION
C     STEP       : STEP SIZE THAT MINIMIZES THE COST FUNCTIONAL
C     COSTOD     : J(U(I))
C     STORU      : TEMPERARAY STORAGE OF CONTROL
C     STORX      : TEMPERARAY STORAGE OF STATES
C     ALPH       : ALPHA (STEP LENGHT)
C     ALP        : STORAGE VECTOR OF ALPHA
C     CF         : STORAGE VECTOR OF COST FUNCTIONAL
C     MAX        : MAX NUM OF DOUBLING THE ALPHA
C     MAXI       : MAX NUM OF REDUCING THE ALPHA
C     MAXX       : MAX NUM OF INCREASING ALPHA BY 100 TIMES
C
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 J1,J2,J3,J4,J5,J6,L1,L2,L3
      DIMENSION X(1002,6),CTL(1002,3),SRH(1002,3),
     &     STORU(1002,3),STORX(1002,6),TT(1002),ALP(4),CF(4)
      COMMON / ALPHA / AA0
      COMMON / WC / J1,J2,J3,J4,J5,J6,L1,L2,L3
      COMMON / FWC / B1,B2,B3,B4,B5,B6
      COMMON / BBPF / BB1,BB2,BB3,BB4,BB5,BB6
      COMMON / XVEC / STORX
C
C     INITIALISE STORAGE VECTORS : ALPHA,COST FUNCTION
C
      AAA0 = AA0
      IJK = 0
      KLM = 0
      MAX = 20
      MAXI = 20
      MAXX = 20
7     DO 10 I = 1,3
      ALP(I) = 0.
10    CF(I) = 0.
```

```
      ALP(4) = 0.
      CF(4) = COSTOD
      II = 0
      ALPH = AAA0
C     WRITE (8,77) AAA0,IJK
C77    FORMAT (T7,'INITIAL ALPHA=',E16.8,5X,'LOOP =',I2)
C*****COMPUTE U(I+1) WITH INITIAL GUESS OF ALPHA,
C*****SOLVE STATE EQN AND COMPUTE J(U(I+1))
      DO 20 J = 1,3
      DO 20 I = 1,KOUNT
      STORU(I,J) = CTL(I,J) + ALPH*SRH(I,J)
20    CONTINUE
      CALL STATE ( STORU,KOUNT, STORX )
      CALL COST ( STORX,STORU,H,KOUNT, DUM,COST1 )
C*****UPDATE ALPHA AND CORRESPONDING COST FUNCTION
      ALP(3) = ALP(4)
      ALP(4) = ALPH
      CF(3) = CF(4)
      CF(4) = COST1
101   STOCF = CF(4)
      II = II + 1
C*****DOUBLE ALPHA
      ALPH = 2.*ALPH
      DO 102 J = 1,3
      DO 102 I =1,KOUNT
102   STORU(I,J) = CTL(I,J) + ALPH*SRH(I,J)
      CALL STATE (  STORU,KOUNT, STORX )
      CALL COST ( STORX,STORU,H,KOUNT, DUM,COST1 )
      DO 30 I = 1,3
      ALP(I) = ALP(I+1)
30    CF(I) = CF(I+1)
      ALP(4) = ALPH
      CF(4) = COST1
      IF ( CF(4).GT.STOCF ) GO TO 150
      IF ( II.GT.MAX ) GO TO 350
      GO TO 101
C*****REDUCE ALPHA SINCE J(U) STARTS TO INCREASE
150   ALPH = ( ALPH + ALPH/2. ) / 2.
      DO 251 J = 1,3
      DO 251 I = 1,KOUNT
251   STORU(I,J) = CTL(I,J) + ALPH*SRH(I,J)
      CALL STATE ( STORU,KOUNT, STORX )
      CALL COST ( STORX,STORU,H,KOUNT, DUM,COST1 )
      DO 33 K = 1,2
      CF(K) = CF(K+1)
      ALP(K) = ALP(K+1)
33    CONTINUE
      CF(3) = COST1
      ALP(3) = ( ALP(3) + ALP(4) ) / 2.
C     WRITE (6,500) (CF(I),ALP(I),I=1,4)
C500   FORMAT (T7,'J(U) = ',E21.14,5X,'ALPHA = ',E16.8)
      IF ( (CF(4).GT.CF(3)).AND.(CF(3).GT.CF(2)).AND.(CF(2).GT.CF(1)) )
     &    GO TO 40
      GO TO 50
```

```
40     CONTINUE
C      WRITE (6,60)
C60     FORMAT (T7,'COST IS INCREASING MONOTONICALLY IN THE BEGINNING,',
C     &   'TRY SMALLER ALPHA')
       AAA0 = 0.05 * AAA0
       IJK = IJK + 1
       IF ( IJK.GT.MAXI ) GO TO 50
       GO TO 7
50     CALL CURFIT ( CF,ALP, STEP )
       WRITE (6,501) STEP
501    FORMAT (T7,'OPTIMAL STEP LENGHT IS',E16.8)
       RETURN
350    WRITE (6,400) AAA0,KLM
400    FORMAT (T7,'ESTI ALPHA TOO SMALL,TRY LARGER ALPHA',E16.8,5X,I2)
       AAA0 = 100*AAA0
       KLM = KLM + 1
       IF ( KLM.GT.MAXX ) GO TO 999
       GO TO 7
999    STOP
       END
       SUBROUTINE CURFIT ( CF,ALP, ALPMIN )
C
C      IT PERFORMS CURVE FITTING THROUGH FOUR POINTS AND LOCATE THE MIN
C      ALPMIN     : OPTIMAL ALPHA
C      P          : CURVE FITTING POLYNOMIAL
C
       REAL *8 CF(4),CF1,CF2,CF3,CF4,LENGHT
       REAL *8 ALP(4),ALP1,ALP2,ALP3,ALP4,A,Y
       REAL *8 L1,L2,L3,L4,P,ALPMIN
C
C      DEFINE STATEMENT FUNCTIONS
C
       L1(A)  =  (  (A-ALP2)/(ALP1-ALP2)  )*(  (A-ALP3)/(ALP1-ALP3)  )*
      &          (  (A-ALP4)/(ALP1-ALP4)  )
       L2(A)  =  (  (A-ALP1)/(ALP2-ALP1)  )*(  (A-ALP3)/(ALP2-ALP3)  )*
      &          (  (A-ALP4)/(ALP2-ALP4)  )
       L3(A)  =  (  (A-ALP1)/(ALP3-ALP1)  )*(  (A-ALP2)/(ALP3-ALP2)  )*
      &          (  (A-ALP4)/(ALP3-ALP4)  )
       L4(A)  =  (  (A-ALP1)/(ALP4-ALP1)  )*(  (A-ALP2)/(ALP4-ALP2)  )*
      &          (  (A-ALP3)/(ALP4-ALP3)  )
C
       P(Y)  = CF1*L1(Y)  + CF2*L2(Y)  + CF3*L3(Y)  + CF4*L4(Y)
C
       CF1 = CF(1)
       CF2 = CF(2)
       CF3 = CF(3)
       CF4 = CF(4)
C
       ALP1 = ALP(1)
       ALP2 = ALP(2)
       ALP3 = ALP(3)
       ALP4 = ALP(4)
C
       N = 100
```

```
      LENGHT = ( ALP4-ALP1 ) / N
      ALPMIN = ALP1
      DO 40 I = 1,N
      Y = ALP1 + FLOAT(I)*LENGHT
      IF ( P(Y).LT.P(ALPMIN) ) ALPMIN = Y
40    CONTINUE
      RETURN
      END
      SUBROUTINE NEWCTL (KOUNT,SRH,STEP, CTL )
C
C     IT COMPUTES THE NEW CONTROL U(I+1)
C     CTL   : CONTROL
C     STEP  : THE STEP THAT MINIMIZES THE COST
C
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 ID0
      DIMENSION STORX(1002,6),CTL(1002,3),SRH(1002,3),DCT(3)
      COMMON / XVEC / STORX
      COMMON / XSTATE / AD0,ED0,ID0,OD0,WD0,RD0
      DCT(1) = AD0
      DCT(2) = ED0
      DCT(3) = ID0
      DO 10 I = 1,3
      IF ( I.EQ.1 ) K = 2
      IF ( I.EQ.2 ) K = 1
      IF ( I.EQ.3 ) K = 3
      DO 10 J = 1,KOUNT
      CTL(J,I) = CTL(J,I) + STEP*SRH(J,I)
10    CONTINUE
      RETURN
      END
      FUNCTION DAD (AD,ED,RD,FR,FT)
C
C     FR,FT,FN : CONTROL TORQUES
C     A,E,I,O,W,R : STATE VARIABLES
C
C     D.E. FOR THE SEMIMAJOR AXIS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL *8 MU,I0
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      DAD = ( 2.*((A0+AD)**2)/(MU*(A0+AD)*(1.-(E0+ED)**2))**0.5 )
     &    *((1.+(E0+ED)*DCOS(R0+RD))*FT + (E0+ED)*FR*DSIN(R0+RD))
      RETURN
      END
      FUNCTION DED (AD,ED,RD,FR,FT)
C
C     D.E. FOR ECCENTRICITY OF THE LOCUS
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL *8 MU,I0
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      T1 = ((A0+AD)*(1.-(E0+ED)**2)/MU)**0.5
      T2 = 2.*DCOS(R0+RD)+(E0+ED)*(1.+DCOS(R0+RD)*DCOS(R0+RD))
```

```
      T3 =  (FT/(1.+(E0+ED)*DCOS(R0+RD)))
      T4 = FR*DSIN(R0+RD)
      DED = T1 * ( (T2*T3)  + T4 )
      RETURN
      END
      FUNCTION DID (AD,ED,WD,RD,FN)
C
C     D.E. FOR INCLINATION
C
      IMPLICIT REAL*8  (A-H,O-Z)
      REAL *8 MU,I0
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      T1 = ((A0+AD)*(1.-(E0+ED)**2)/MU)**0.5
      T2 =  (DCOS(W0+WD+R0+RD)*FN)/(1.+(E0+ED)*DCOS(R0+RD))
      DID = T1 * T2
      RETURN
      END
      FUNCTION DOD (AD,ED,ID,WD,RD,FN)
C
C     D.E. FOR LONGITUDE OF THE ASCENDING NODE
C
      IMPLICIT REAL*8  (A-H,O-Z)
      REAL *8 MU,I0,ID
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      T1 = ((A0+AD)*(1.-(E0+ED)**2)/MU)**0.5
      T2=(DSIN(W0+WD+R0+RD)*FN)  /  (DSIN(I0+ID)*(1.+(E0+ED)*DCOS(R0+RD)))
      DOD = T1 * T2
      RETURN
      END
      FUNCTION DWD  (AD,ED,ID,WD,RD,FR,FT,FN)
C
C     D.E. FOR ARGUMENT OF PERIAPSIS
C
      IMPLICIT REAL*8  (A-H,O-Z)
      REAL *8 MU,I0,ID
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      T1 =  (  ((A0+AD)*(1.-(E0+ED)**2)/MU)**0.5  )/(E0+ED)
      T2 =  (  (2.+(E0+ED)*DCOS(R0+RD))*FT*DSIN(R0+RD)  )/
     &          (1.+(E0+ED)*DCOS(R0+RD))
      T3 =  ((E0+ED)*FN*DSIN(W0+WD+R0+RD))/
     &    ((1.+(E0+ED)*DCOS(R0+RD))*DTAN(I0+ID))
      DWD = T1*(T2 - FR*DCOS(R0+RD)  - T3)
      RETURN
      END
      FUNCTION DRD  (AD,ED,RD,FR,FT)
C
C     D.E. FOR TRUE ANOMALY EPOCH
C
      IMPLICIT REAL*8  (A-H,O-Z)
      REAL *8 MU,I0
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      T1 =  (  ((A0+AD)*(1.-(E0+ED)**2)/MU)**0.5  )/(E0+ED)
      T2 = - (2.+(E0+ED)*DCOS(R0+RD))*FT*DSIN(R0+RD)/
     &    (1.+(E0+ED)*DCOS(R0+RD))
```

```
      DRD = T1*(T2 + FR*DCOS(R0+RD))
      RETURN
      END
      FUNCTION DS1 (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FR,FT,FN,J1)
C
C     COSTATE EQUATION 1
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL *8 ID,I0,I,J1,MU
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      A = A0 + AD
      E = E0 + ED
      I = I0 + ID
      W = W0 + WD
      R = R0 + RD
         F0 = 2.*J1*AD
      T1 = ((1.+E*DCOS(R))*FT + E*FR*DSIN(R))/(MU*(1.-(E**2))*A)
      T2 = 4.*A*(MU*(1.-(E**2))*A)**0.5
      T3 = (A**2)*MU*(1.-(E**2))*(MU*(1.-(E**2))*A)**(-0.5)
         F1 = T1 * (T2 - T3)
      T5 = ( 0.5*(A*(1.-E**2)/MU)**(-0.5) )* ( (1.-(E**2))/MU )
      T6=( (FT*(2.*DCOS(R)+E*(1.+(DCOS(R)*DCOS(R)))))/(1.+E*DCOS(R)) )
     &          +FR*DSIN(R)
         F2 = T5 * T6
      T7 = 0.5*((A*(1.-(E**2)))/MU)**(-0.5)
      T8 = (1.-(E**2))/MU
      T9 = (FN*DCOS(W+R))/(1.+E*DCOS(R))
         F3 = T7*T8*T9
      T10 = (FN*DSIN(W+R))/(DSIN(I)*(1.+E*DCOS(R)))
         F4 = T7*T8*T10
      T11 = (FT*(2.+E*DCOS(R))*DSIN(R))/(1.+E*DCOS(R))
      T12 = FR*DCOS(R)
      T13 = (E*FN*DSIN(W+R))/((1.+E*DCOS(R))*DTAN(I))
         F5 = (T7/E) * T8 * (T11 - T12 - T13)
         F6 = (T7/E) * T8 * (-T11 + T12)
      DS1 = -( - F0 - F1*S1 - F2*S2 - F3*S3 - F4*S4 - F5*S5 - F6*S6 )
      RETURN
      END
      FUNCTION DS2 (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FR,FT,FN,J2)
C
C     COSTATE EQUATION 2
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL *8 ID,I0,I,J2,MU
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      A = A0 + AD
      E = E0 + ED
      I = I0 + ID
      W = W0 + WD
      R = R0 + RD
         F0 = 2.*J2*ED
      T1 = (1.+E*DCOS(R))*FT + E*FR*DSIN(R)
      T2 =(2.*MU*(A**3)*E*(MU*A*(1.-(E**2)))**(-0.5))/(MU*A*(1.-(E**2)))
      T3 = (2.*(A**2)*(FT*DCOS(R)+FR*DSIN(R)))/(MU*A*(1.-(E**2)))**0.5
```

```
      F1 = (T1 * T2) + T3
     T4 =  - (A*E)/(MU**0.5)
     T5 = ( FT*(2.*DCOS(R) +E*(1.+(DCOS(R)*DCOS(R)))) /
    &      (1.+E*DCOS(R)) )   + FR*DSIN(R)
     T6 = (A*(1.-E**2))**(-0.5)
     T7 = FT *  (A*(1.-(E**2))/MU)**0.5
     T8 = ( (1.+E*DCOS(R))*(1.+(DCOS(R)*DCOS(R))) -
    &     (2.*DCOS(R)+E*(1.+(DCOS(R)*DCOS(R))))*DCOS(R)  )
    &          / (1.+E*DCOS(R))**2
      F2 = (T4 * T5 * T6)  +  (T7 * T8)
     T9 = (DCOS(W+R)) / (1.+E*DCOS(R))
     T10 = (A*(1.-E*E)/MU)**(-0.5)
     T11 = (A*E)/MU
     T12 = ((A*(1.-E*E))/MU)**0.5
     T13 = (DCOS(W+R)*DCOS(R)) / (1.+E*DCOS(R))**2
      F3 = -FN * ( (T9*T10*T11 ) + (T12*T13) )
     T14 = DSIN(W+R)/(DSIN(I)*(1.+E*DCOS(R)))
     T15 = (DSIN(W+R)*DCOS(R)) / (DSIN(I)*(1.+E*DCOS(R))**2)
      F4 = -FN * ( (T14*T10*T11) + (T12*T15) )
     T16 = ( (2.+E*DCOS(R))*FT*DSIN(R)/(1.+E*DCOS(R)) ) - FR*DCOS(R)
     T17 = (E*FN*DSIN(W+R)) /(DTAN(I)*(1.+E*DCOS(R)))
     T18 = (A*(1.-E*E)/(MU*E*E))**(-0.5)
     T19 = ( E*E*E*A*MU+A*MU*E*(1.-E*E)) / (E*E*E*E*MU*MU)
     T20 = FT*DSIN(R)*( (1.+E*DCOS(R))*DCOS(R)-(2.+E*DCOS(R))*DCOS(R) )
    &  / (1.+E*DCOS(R))**2
     T21 = ( FN*DSIN(W+R)) / (DTAN(I)*(1.+E*DCOS(R))**2)
      F5 = ( - (T16-T17)*T18*T19 ) + (T12/E)*(T20-T21)
     T22 = ( (2.+E*DCOS(R))*FT*DSIN(R)/(1.+E*DCOS(R)) ) - FR*DCOS(R)
     T23 = (A*(MU*E*E + MU*(1.-E*E))) / (E*E*E*MU*MU)
      F6 = ( T22 * T18 * T23 )  -   ( (T12/E) * T20 )
     DS2 = -( - F0 - F1*S1 - F2*S2 - F3*S3 - F4*S4 - F5*S5 - F6*S6 )
     RETURN
     END
     FUNCTION DS3 (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FN,J3)
C
C     COSTATE EQUATION 3
C
     IMPLICIT REAL*8 (A-H,O-Z)
     REAL *8 ID,IO,I,J3,MU
     COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
     A = A0 + AD
     E = E0 + ED
     I = I0 + ID
     W = W0 + WD
     R = R0 + RD
      F0 = 2.*J3*ID
      F1 = 0.
      F2 = 0.
      F3 = 0.
     T1 = (A*(1.-E*E)/MU)**0.5
     T2 = DSIN(W+R)/(1.+E*DCOS(R))
     T3 = DCOS(I)/(DSIN(I)*DSIN(I))
      F4 = -FN * T1 * T2 *T3
     T4 = 1./(DSIN(I)*DSIN(I))
```

```
      F5 = FN * T1 *T2 *T4
      F6 = 0.
   DS3 = -( - F0 - F1*S1 - F2*S2 -F3*S3 - F4*S4 - F5*S5 -F6*S6 )
   RETURN
   END
   FUNCTION DS4 (S1,S2,S3,S4,S5,S6,OD,J4)
C
C  COSTATE EQUATION 4
C
   IMPLICIT REAL*8 (A-H,O-Z)
   REAL *8 ID,IO,J4,MU
   COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      F0 = 2.*J4*OD
      F1 = 0.
      F2 = 0.
      F3 = 0.
      F4 = 0.
      F5 = 0.
      F6 = 0.
   DS4 = -( - F0 - F1*S1 - F2*S2 - F3*S3 - F4*S4 - F5*S5 - F6*S6 )
   RETURN
   END
   FUNCTION DS5 (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FN,J5)
C
C  COSTATE EQUATION 5
C
   IMPLICIT REAL*8 (A-H,O-Z)
   REAL *8 ID,I0,I,J5,MU
   COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
   A = A0 + AD
   E = E0 + ED
   I = I0 + ID
   W = W0 + WD
   R = R0 + RD
      F0 = 2.*J5*WD
      F1 = 0.
      F2 = 0.
   T1 = ((A*(1.-E*E))/MU) **0.5
   T2 = (DSIN(W+R))/(1.+E*DCOS(R))
      F3 = - FN * T1 * T2
   T3 = DCOS(W+R) / ( DSIN(I)*(1.+E*DCOS(R)) )
      F4 = FN * T1 * T3
   T4 = ( DCOS(W+R)*DCOTAN(I) ) / (1.+E*DCOS(R))
      F5 = - FN * T1 * T4
      F6 = 0.
   DS5 = -( - F0 - F1*S1 - F2*S2 - F3*S3 - F4*S4 - F5*S5 - F6*S6 )
   RETURN
   END
   FUNCTION DS6 (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FR,FT,FN,J6)
C
C  COSTATE EQUATION 6
C
   IMPLICIT REAL*8 (A-H,O-Z)
   REAL *8 ID,I0,I,J6,MU
```

```
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      A = A0 + AD
      E = E0 + ED
      I = I0 + ID
      W = W0 + WD
      R = R0 + RD
         F0 = 2.*J6*RD
      T1 = (2.*A*A*E)/(MU*A*(1.-E*E))**0.5
      T2 = FR*DCOS(R) - FT*DSIN(R)
         F1 = T1 * T2
      T3 = ((A*(1.-E*E))/MU)**0.5
      T4 = (1.+E*DCOS(R)) * (-2.*DSIN(R)-E*DSIN(2*R))
      T5 = E*DSIN(2*R) + E*E*DSIN(R)*(1.+DCOS(R)*DCOS(R))
      T6 = (1.+E*DCOS(R))**2
         F2 = T3 * ( (( FT*(T4+T5) ) / T6 ) + FR*DCOS(R) )
      T7 = E*DSIN(R)*DCOS(W+R) - (1.+E*DCOS(R))*(DSIN(W+R))
      T8 = (1.+E*DCOS(R))**2
         F3 = FN * T3 * T7 / T8
      T9 = (1.+E*DCOS(R)) * DCOS(W+R)
      T10 = E * DSIN(R) * DSIN(W+R)
      T11 = DSIN(I) * (1.+E*DCOS(R))**2
         F4 = FN * T3 * (T9+T10) / T11
      T12 = (1.+E*DCOS(R)) * (2.*DCOS(R)+E*DCOS(2*R))
      T13 = E * (2.+E*DCOS(R)) * DSIN(R) * DSIN(R)
      T14 = (1.+E*DCOS(R))**2
      T15 = FR*DSIN(R)
      T16 = E*FN/DTAN(I)
      T17 = (1.+E*DCOS(R)) * ( DCOS(W+R) )
      T18 = E*DSIN(W+R)*DSIN(R)
         F5 = (T3/E) * ( FT*(T12+T13)/T14 + T15 - (T16*(T17+T18)/T14) )
         F6 = (T3/E) * ( -FT*(T12+T13)/T14 -T15 )
      DS6 = -( - F0 - F1*S1 - F2*S2 - F3*S3 - F4*S4 - F5*S5 - F6*S6 )
      RETURN
      END
      SUBROUTINE RKGT (AD,ED,ID,OD,WD,RD,FR,FT,FN,H)
C
C     THIS SUBROTINE COMPUTES THE D.E. BY FOURTH ORDER R-K METHOD
C
      IMPLICIT REAL*8 (A-H,I,K,O-Z)
C
C     COMPUTE K1'S FOR DAD,....,DRD
C
      KA1 = H*DAD ( AD , ED , RD , FR , FT )
      KE1 = H*DED ( AD , ED , RD , FR , FT )
      KI1 = H*DID ( AD , ED , WD , RD , FN )
      KO1 = H*DOD ( AD , ED , ID , WD , RD , FN )
      KW1 = H*DWD ( AD , ED , ID , WD , RD , FR , FT , FN )
      KR1 = H*DRD ( AD , ED , RD , FR , FT )
C
C     COMPUTE K2'S FOR DAD,....,DRD
C
      KA2 = H*DAD ( AD+KA1/2. , ED+KE1/2. , RD+KR1/2. , FR , FT )
      KE2 = H*DED ( AD+KA1/2. , ED+KE1/2. , RD+KR1/2. , FR , FT)
      KI2 = H*DID ( AD+KA1/2. , ED+KE1/2. , WD+KW1/2. , RD+KR1/2. , FN )
```

```
      KO2 = H*DOD ( AD+KA1/2. , ED+KE1/2. , ID+KI1/2. ,
     &          WD+KW1/2. , RD+KR1/2. , FN )
      KW2 = H*DWD ( AD+KA1/2. , ED+KE1/2. , ID+KI1/2. ,
     &          WD+KW1/2. , RD+KR1/2. , FR , FT , FN )
      KR2 = H*DRD ( AD+KA1/2. , ED+KE1/2. , RD+KR1/2. , FR , FT )
C
C     COMPUTE K3'S FOR DAD,...,DRD
C
      KA3 = H*DAD ( AD+KA2/2. , ED+KE2/2. , RD+KR2/2. , FR , FT )
      KE3 = H*DED ( AD+KA2/2. , ED+KE2/2. , RD+KR2/2. , FR , FT)
      KI3 = H*DID ( AD+KA2/2. , ED+KE2/2. , WD+KW2/2. , RD+KR2/2. , FN )
      KO3 = H*DOD ( AD+KA2/2. , ED+KE2/2. , ID+KI2/2. ,
     &          WD+KW2/2. , RD+KR2/2. , FN )
      KW3 = H*DWD ( AD+KA2/2. , ED+KE2/2. , ID+KI2/2. ,
     &          WD+KW2/2. , RD+KR2/2. , FR , FT , FN )
      KR3 = H*DRD ( AD+KA2/2. , ED+KE2/2. , RD+KR2/2. , FR , FT )
C
C     COMPUTE K4'S FOR DAD,...,DRD
C
      KA4 = H*DAD ( AD+KA3    , ED+KE3    , RD+KR3    , FR , FT )
      KE4 = H*DED ( AD+KA3    , ED+KE3    , RD+KR3    , FR , FT )
      KI4 = H*DID ( AD+KA3    , ED+KE3    , WD+KW3    , RD+KR3    , FN )
      KO4 = H*DOD ( AD+KA3   , ED+KE3   , ID+KI3 ,WD+KW3 , RD+KR3   , FN )
      KW4 = H*DWD (AD+KA3 , ED+KE3 , ID+KI3 ,WD+KW3 , RD+KR3 ,FR,FT,FN)
      KR4 = H*DRD ( AD+KA3    , ED+KE3    , RD+KR3    , FR , FT )
C
C     COMPUTE NEW VALUE FOR AD
C
      AD = AD + ( KA1 + 2.*KA2 + 2.*KA3 + KA4 ) /6.
      ED = ED + ( KE1 + 2.*KE2 +2.*KE3 + KE4 ) / 6.
      ID = ID + ( KI1 + 2.*KI2 + 2.*KI3 + KI4 ) /6.
      OD = OD + ( KO1 + 2.*KO2 + 2.*KO3 + KO4 ) /6.
      WD = WD + ( KW1 + 2.*KW2 + 2.*KW3 + KW4 ) /6.
      RD = RD + ( KR1 + 2.*KR2 + 2.*KR3 + KR4 ) /6.
      RETURN
      END
      SUBROUTINE COSTAT (AD,ED,ID,OD,WD,RD,FR,FT,FN)
C
C     THIS SUBROUTINE COMPUTE THE COSTATE EQUATIONS OF THE ORBITAL
C     SATELLITE USING FOURTH ORDER RUNGE KUTTA METHOD
C
C     J1,...,J6 ARE WEGHTING FACTORS OF THE STATES
C     H : STEP SIZE OF INTEGRATION
C
      IMPLICIT REAL *8 (A-H,I,J,K,L,M,O-Z)
      COMMON / SS / S1,S2,S3,S4,S5,S6
      COMMON / WC / J1,J2,J3,J4,J5,J6,L1,L2,L3
      COMMON / SIZE / H
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
C
C     COMPUTE K1'S FOR DS1,...,DS6
C
      KS11 = H*DS1 (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FR,FT,FN,J1)
      KS21 = H*DS2 (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FR,FT,FN,J2)
```

```
      KS31 = H*DS3  (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FN,J3)
      KS41 = H*DS4  (S1,S2,S3,S4,S5,S6,OD,J4)
      KS51 = H*DS5  (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FN,J5)
      KS61 = H*DS6  (S1,S2,S3,S4,S5,S6,AD,ED,ID,WD,RD,FR,FT,FN,J6)
C
C     COMPUTE K2'S FOR DS1,...,DS6
C
      KS12 = H*DS1  (S1+KS11/2.,S2+KS21/2.,S3+KS31/2.,S4+KS41/2.,
     &               S5+KS51/2.,S6+KS61/2.,AD,ED,ID,WD,RD,FR,FT,FN,J1)
      KS22 = H*DS2  (S1+KS11/2.,S2+KS21/2.,S3+KS31/2.,S4+KS41/2.,
     &               S5+KS51/2.,S6+KS61/2.,AD,ED,ID,WD,RD,FR,FT,FN,J2)
      KS32 = H*DS3  (S1+KS11/2.,S2+KS21/2.,S3+KS31/2.,S4+KS41/2.,
     &               S5+KS51/2.,S6+KS61/2.,AD,ED,ID,WD,RD,FN,J3)
      KS42 = H*DS4  (S1+KS11/2.,S2+KS21/2.,S3+KS31/2.,S4+KS41/2.,
     &               S5+KS51/2.,S6+KS61/2.,OD,J4)
      KS52 = H*DS5  (S1+KS11/2.,S2+KS21/2.,S3+KS31/2.,S4+KS41/2.,
     &               S5+KS51/2.,S6+KS61/2.,AD,ED,ID,WD,RD,FN,J5)
      KS62 = H*DS6  (S1+KS11/2.,S2+KS21/2.,S3+KS31/2.,S4+KS41/2.,
     &               S5+KS51/2.,S6+KS61/2.,AD,FD,ID,WD,RD,FR,FT,FN,J6)
C
C     COMPUTE K3'S FOR DS1,...,DS6
C
      KS13 = H*DS1  (S1+KS12/2.,S2+KS22/2.,S3+KS32/2.,S4+KS42/2.,
     &               S5+KS52/2.,S6+KS62/2.,AD,ED,ID,WD,RD,FR,FT,FN,J1)
      KS23 = H*DS2  (S1+KS12/2.,S2+KS22/2.,S3+KS32/2.,S4+KS42/2.,
     &               S5+KS52/2.,S6+KS62/2.,AD,ED,ID,WD,RD,FR,FT,FN,J2)
      KS33 = H*DS3  (S1+KS12/2.,S2+KS22/2.,S3+KS32/2.,S4+KS42/2.,
     &               S5+KS52/2.,S6+KS62/2.,AD,ED,ID,WD,RD,FN,J3)
      KS43 = H*DS4  (S1+KS12/2.,S2+KS22/2.,S3+KS32/2.,S4+KS42/2.,
     &               S5+KS52/2.,S6+KS62/2.,OD,J4)
      KS53 = H*DS5  (S1+KS12/2.,S2+KS22/2.,S3+KS32/2.,S4+KS42/2.,
     &               S5+KS52/2.,S6+KS62/2.,AD,ED,ID,WD,RD,FN,J5)
      KS63 = H*DS6  (S1+KS12/2.,S2+KS22/2.,S3+KS32/2.,S4+KS42/2.,
     &               S5+KS52/2.,S6+KS62/2.,AD,ED,ID,WD,RD,FR,FT,FN,J6)
C
C     COMPUTE K4'S FOR DS1,...,DS6
C
      KS14 = H*DS1  (S1+KS13,S2+KS23,S3+KS33,S4+KS43,S5+KS53,
     &               S6+KS63,AD,ED,ID,WD,RD,FR,FT,FN,J1)
      KS24 = H*DS2  (S1+KS13,S2+KS23,S3+KS33,S4+KS43,S5+KS53,
     &               S6+KS63,AD,ED,ID,WD,RD,FR,FT,FN,J2)
      KS34 = H*DS3  (S1+KS13,S2+KS23,S3+KS33,S4+KS43,S5+KS53,
     &               S6+KS63,AD,ED,ID,WD,RD,FN,J3)
      KS44 = H*DS4  (S1+KS13,S2+KS23,S3+KS33,S4+KS43,S5+KS53,
     &               S6+KS63,OD,J4)
      KS54 = H*DS5  (S1+KS13,S2+KS23,S3+KS33,S4+KS43,S5+KS53,
     &               S6+KS63,AD,ED,ID,WD,RD,FN,J5)
      KS64 = H*DS6  (S1+KS13,S2+KS23,S3+KS33,S4+KS43,S5+KS53,
     &               S6+KS63,AD,ED,ID,WD,RD,FR,FT,FN,J6)
C
C     COMPUTE NEW VALUES FOR S1,...,S6
C
      S1 = S1 + ( KS11+2.*KS12+2.*KS13+KS14 ) / 6.
      S2 = S2 + ( KS21+2.*KS22+2.*KS23+KS24 ) / 6.
```

```
      S3 = S3 + ( KS31+2.*KS32+2.*KS33+KS34 ) / 6.
      S4 = S4 + ( KS41+2.*KS42+2.*KS43+KS44 ) / 6.
      S5 = S5 + ( KS51+2.*KS52+2.*KS53+KS54 ) / 6.
      S6 = S6 + ( KS61+2.*KS62+2.*KS63+KS64 ) / 6.
      RETURN
      END
      SUBROUTINE GRAD (AD,ED,ID,OD,WD,RD,FR,FT,FN,J, GRD)
C
C     GRADIENT FUNCTION
C
C     J1,...,J6 ARE THE WEGHTING FACTORS OF THE STATES
C     L1,L2,L3 ARE THE WEGHTING FACTORS OF FUEL (ENERGY)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL *8 MU,I,IO,ID,J1,J2,J3,J4,J5,J6,L1,L2,L3
      DIMENSION GRD(1002,3)
      COMMON / SS / S1,S2,S3,S4,S5,S6
      COMMON / WC / J1,J2,J3,J4,J5,J6,L1,L2,L3
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      COMMON / DCTL / FR0,FT0,FN0
      A = A0 + AD
      E = E0 + ED
      I = I0 + ID
      O = O0 + OD
      W = W0 + WD
      R = R0 + RD
C*****FOR CONSTRAINTED MIN ENERGY CASE
      FRL1 = + 2.*L1*FR
      FTL2 = + 2.*L2*FT
      FNL3 = + 2.*L3*FN
      T1 = (2.*A*A) / (MU*A*(1.-E*E))**0.5
      T2 = ( (A*(1.-E*E))/MU ) **0.5
      T3 = 1. + E*DCOS(R)
      T4 = 2.*DCOS(R) + E*(1.+DCOS(R)*DCOS(R))
      T5 = 2. + E*DCOS(R)
      GRD(J,1) = FRL1 + T1*E*DSIN(R)*S1 + T2*DSIN(R)*S2
     &         - T2*DCOS(R)*S5/E + T2*DCOS(R)*S6/E
      GRD(J,2) = FTL2 + T1*T3*S1 + T2*T4*S2/T3
     &         + T2*T5*DSIN(R)*S5/(E*T3) - T2*T5*DSIN(R)*S6/(E*T3)
      GRD(J,3) = FNL3 + T2*DCOS(W+R)*S3/T3
     &    + T2*DSIN(W+R)*S4/(T3*DSIN(I)) - T2*DCOTAN(I)*DSIN(W+R)*S5/T3
      RETURN
      END
      SUBROUTINE COST ( X,CTL,H,KOUNT, SUM,COSTS )
C
C     THIS SUBROUTINE COMPUTE THE COST FUNCTION OF THE SYSTEM
C     BY TRAPIZOID (EULER METHOD) RULE
C
      IMPLICIT REAL*8 (A-H,J,L,O-Z)
      DIMENSION X(1002,6),CTL(1002,3)
      COMMON / WC / J1,J2,J3,J4,J5,J6,L1,L2,L3
      COMMON / FWC / B1,B2,B3,B4,B5,B6
      COMMON / BBPF / BB1,BB2,BB3,BB4,BB5,BB6
      COMMON / DCTL / FR0,FT0,FN0
```

```
      COMMON / FUCOST / FUEL
      KOUNT1 = KOUNT - 1
      SUM = 0.
      FUEL = 0.
      DO 20 K = 2,KOUNT1
C*****FOR CONSTRAINTED MIN ENERGY CASE
      CL1 = L1*CTL(K,1)**2
      CL2 = L2*CTL(K,2)**2
      CL3 = L3*CTL(K,3)**2
      SUM = SUM + CL1 + CL2 + CL3
     &        J1*X(K,1)**2 + J2*X(K,2)**2 + J3*X(K,3)**2 + J4*X(K,4)**2
     &        J5*X(K,5)**2 + J6*X(K,6)**2
      FUEL=FUEL+L1*CTL(K,1)**2 + L2*CTL(K,2)**2 + L3*CTL(K,3)**2
20    CONTINUE
C*****FOR CONSTRAINTED MIN ENERGY CASE
      CL1 = L1*CTL(KOUNT,1)**2
      CL2 = L2*CTL(KOUNT,2)**2
      CL3 = L3*CTL(KOUNT,3)**2
      SUM = SUM + 0.5*( CL1 + CL2 + CL3
     &       J1*X(KOUNT,1)**2 + J2*X(KOUNT,2)**2
     &      J3*X(KOUNT,3)**2 + J4*X(KOUNT,4)**2 + J5*X(KOUNT,5)**2
     &      J6*X(KOUNT,6)**2 )
      FUEL=FUEL+0.5*(L1*CTL(KOUNT,1)**2+L2*CTL(KOUNT,2)**2
     &    +L3*CTL(KOUNT,3)**2)
C*****FOR CONSTRAINTED MIN ENERGY CASE
      CL1 = L1*CTL(1,1)**2
      CL2 = L2*CTL(1,2)**2
      CL3 = L3*CTL(1,3)**2
      SUM =SUM + 0.5*( CL1 + CL2 + CL3
     &      J1*X(1,1)**2 + J2*X(1,2)**2 + J3*X(1,3)**2 + J4*X(1,4)**2
     &      J5*X(1,5)**2 + J6*X(1,6)**2 )
      FUEL = FUEL + 0.5*( L1*CTL(1,1)**2 + L2*CTL(1,2)**2
     &       L3*CTL(1,3)**2 )
      SUM = SUM * H
      FUEL = FUEL * H
      FICOST = B1*(X(KOUNT,1)+BB1)**2 + B2*(X(KOUNT,2)+BB2)**2
     &          B3*(X(KOUNT,3)+BB3)**2 + B4*(X(KOUNT,4)+BB4)**2
     &          B5*(X(KOUNT,5)+BB5)**2 + B6*(X(KOUNT,6)+BB6)**2
      COSTS = SUM + FICOST
      RETURN
      END
      SUBROUTINE INPUT ( TT,CTL,KOUNT )
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL *8 MU,IO,ID,IDO,J1,J2,J3,J4,J5,J6,L1,L2,L3
      DIMENSION CTL(1002,3),TT(1002),CHAR(20)
      COMMON / XSTATE / ADO,EDO,IDO,ODO,WDO,RDO
      COMMON / SCBLK / MU,AO,EO,IO,OO,WO,RO
      COMMON / WC / J1,J2,J3,J4,J5,J6,L1,L2,L3
      COMMON / FWC / B1,B2,B3,B4,B5,B6
      COMMON / SIZE / H
      COMMON / ALPHA / AAO
      COMMON / DCTL / FRO,FTO,FNO
      READ (5,1302) CHAR
      WRITE (6,1302) CHAR
```

```
1302  FORMAT (20A4)
C*****INPUT INITIAL CONDITIONS
      READ (5,2) A0,E0,I0,O0,W0,R0
2     FORMAT (6(E9.2,1X))
C*****INPUT INITIAL ERROR IN THE STATES
      READ (5,5) AD0,ED0,ID0,OD0,WD0,RD0
5     FORMAT (6(E9.2,1X))
C*****INPUT WEIGHTING FACTORS FOR THE STATES
C*****INPUT WEIGHTING FACTORS FOR STATES AT FINAL TIME
C*****INPUT WEIGHTING FACTORS FOR CONTROLS
      READ (5,3) J1,J2,J3,J4,J5,J6
      READ (5,3) B1,B2,B3,B4,B5,B6
      READ (5,3) L1,L2,L3
3     FORMAT (6(E10.4,1X))
C*****INPUT GRAVATATIONAL CONSTANT(GM),FINAL TIME,STEPSIZE,ALPHA
      READ (5,4) MU,TF,H,AA0
4     FORMAT (E9.2,1X,F6.2,1X,F10.7,1X,E9.2)
C*****INPUT DESIRED CONTROL MAGNITUDE
      READ (5,30) FR0,FT0,FN0
30    FORMAT (3(E10.4,1X))
C*****OUTPUT ALL READ-IN DATA
      WRITE (6,22) A0,E0,I0,O0,W0,R0
22    FORMAT (T7,'INITIAL CONDITIONS FOR THE STATES ARE : ',//,
     &        (6(T8,E16.8,/)),//)
      WRITE (6,26) AD0,ED0,ID0,OD0,WD0,RD0
26    FORMAT (T7,'STATE ERRORS ARE : ',//,6(T8,E16.8,/),//)
      WRITE (6,23) J1,J2,J3,J4,J5,J6
23    FORMAT (T7,'WEIGHTING FACTORS FOR THE STATES ARE : ',//,
     &        (6(T8,E10.4,/)),//)
      WRITE (6,29) B1,B2,B3,B4,B5,B6
29    FORMAT (T7,'WEIGHTING FACTORS FOR FINAL STATES ARE : ',//,
     &        (6(T8,E10.4,/)),//)
      WRITE (6,24) L1,L2,L3
24    FORMAT (T7,'WEIGHTING FACTORS FOR CONTROLS ARE : ',//,
     &        (3(T8,E10.4,/)),//)
      WRITE (6,25) MU,TF,H,AA0
25    FORMAT ( T7,'MU = ',E9.2,/,T7,'FINAL TIME = ',F6.2,/,T7,
     &        'STEP SIZE = ',F10.7,/,T7,'ALPHA = ',E9.2)
      WRITE (6,31) FR0,FT0,FN0
31    FORMAT (T7,'CONSTRAINTED CONTROLS:',/,3(E10.4,/))
      KOUNT = (TF/H) + 1.5
      J2 = DABS( J1*AD0/ED0 )
      J3 = DABS( J1*AD0/ID0 )
C     J1 = J1*1.0E3
      WRITE (6,23) J1,J2,J3,J4,J5,J6
      B2 = DABS( B1*AD0/ED0 )
      B3 = DABS( B1*AD0/ID0 )
C     B1 = B1 * 1.0E5
      WRITE (6,29) B1,B2,B3,B4,B5,B6
      L1 = DABS( J1*AD0/FR0 )
      L2 = L2*DABS( J1*AD0/FT0 )
      L3 = DABS( J1*AD0/FN0 )
      WRITE (6,24) L1,L2,L3
C*****SET UP INITIAL CONTROL PATTERN
```

```
      TT(1) = 0.
      FR00 = DSIGN(FR0,ED0) * (-1)
      FT00 = DSIGN(FT0,AD0) * (-1)
      FN00 = DSIGN(FN0,ID0) * (-1)
      CTL(1,1) = FR00
      CTL(1,2) = FT00
      CTL(1,3) = FN00
      DO 1011 I = 2,KOUNT
      II = I - 1
      TT(I) = TT(II) + H
      CTL(I,1) = FR00
      CTL(I,2) = FT00
      CTL(I,3) = FN00
 1011 CONTINUE
      RETURN
      END
      SUBROUTINE OUTPUT ( TT,X,CTL,COSTOD,EXIT,G,KOUNT )
C
C     THIS SUBROUTINE OUTPUTS RESULTS
C
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 MU,I0,I0X3
      INTEGER EXIT
      DIMENSION X(1002,6),CTL(1002,3),G(1002,3),TT(1002)
      COMMON / SCBLK / MU,A0,E0,I0,O0,W0,R0
      COMMON / FUCOST / FUEL
C*****COMPUTE MEAN ANOMALY
      ANOM1 = ( (1.-E0**2)**0.5 ) * DSIN(R0)
      ANOM2 = E0 + DCOS(R0)
      ANOM3 = ANOM1 * E0
      ANOM4 = 1. + E0*DCOS(R0)
      ANOM = ANOM1/ANOM2
      RM = DATAN(ANOM) - ( ANOM3/ANOM4 )
C*****COMPUTE DESIRED LATITUDE AND LONGITUDE
      UB = DCOS(O0)*DCOS(W0+RM) - DSIN(O0)*DSIN(W0+RM)*DCOS(I0)
      VB = DSIN(O0)*DCOS(W0+RM) + DCOS(O0)*DSIN(W0+RM)*DCOS(I0)
      WB = DSIN(W0+RM)*DSIN(I0)
      DLA = VB/UB
      DLO =  WB/(1.-(WB*WB))**0.5
      DLATI = DATAN( DLA )
      DLONG = DATAN( DLO )
      WRITE (6,7701) DLATI,DLONG
 7701 FORMAT (T7,'LATITUDE = ',E16.8,/,T7,'LONGTITUDE = ',E16.8)
      WRITE (6,9996)
 9996 FORMAT ('1','THE SUB-OPTIMAL CONTROLS AND STATE ERRORS',
     &        ' ARE AS FOLLOWS',//,
     &   T5,'T',8X,'FR',11X,'FT',11X,'FN',12X,'AD',11X,
     &   'ED',11X,'ID',11X,'OD',11X,'WD',11X,'RD')
      DO 8050 I=1,KOUNT,1
      U1=CTL(I,1)
      U2=CTL(I,2)
      U3=CTL(I,3)
      X1 = X(I,1)
      X2 = X(I,2)
```

```
      X3 = X(I,3)
      X4 = X(I,4)
      X5 = X(I,5)
      X6 = X(I,6)
C*****COMPUTE MEAN ANOMALY
      ANOM1 = ( (1.-(E0+X2)**2)**0.5 ) * DSIN(R0+X6)
      ANOM2 = (E0+X2) + DCOS(R0+X6)
      ANOM3 = ANOM1 * (E0+X2)
      ANOM4 = 1. + (E0+X2)*DCOS(R0+X6)
      ANOM = ANOM1/ANOM2
      X6M = DATAN(ANOM) - ( ANOM3/ANOM4 )
C*****COMPUTE CURRENT VALUE OF LONGITUDE AND LATITUDE
      OOX4 = OO + X4
      WXRX = WO + X5 + RO + X6M
      IOX3 = IO + X3
      UC = DCOS(OOX4)*DCOS(WXRX) -
     1     DSIN(OOX4)*DSIN(WXRX)*DCOS(IOX3)
      VC = DSIN(OOX4)*DCOS(WXRX)
     1     DCOS(OOX4)*DSIN(WXRX)*DCOS(IOX3)
      WC = DSIN(WXRX)*DSIN(IOX3)
      CLA =  VC/UC
      CLO =  WC/(1.-(WC*WC))**0.5
      CLATI = DATAN( CLA )
      CLONG = DATAN( CLO )
C*****COMPUTE LONGITUDE AND LATITUDE ERROR
      ELATI = CLATI - DLATI
      ELONG = CLONG - DLONG
      WRITE (6,9998) TT(I),U1,U2,U3,X1,X2,X3,X4,X5,X6
9998  FORMAT (F7.4,1X,3(E12.5,1X),1X,6(E12.5,1X))
      WRITE (9,9800) TT(I),X1,X2,X3,X4,X5,X6
      WRITE (10,9800) TT(I),U1,U2,U3,ELATI,ELONG
9800  FORMAT (F7.4,1X,6(E19.12,1X))
8050  CONTINUE
      WRITE (6,8070) COSTOD,FUEL
8070  FORMAT (/,T5,'TOTAL COST=',E19.12,/,T5,'FUEL COST=',E19.12)
      WRITE (6,10) EXIT
10    FORMAT (T7,'EXIT = ',I6)
      DO 30 I = 1,KOUNT,10
      WRITE (6,20) TT(I),G(I,1),G(I,2),G(I,3)
20    FORMAT (T7,F10.5,1X,(3(E19.12,1X)))
30    CONTINUE
      RETURN
      END
```

REFERENCE

1. L.S.Pontryagin, V.G.Boltyanski, R.V.Gamkrelidge, and E.G.Mishshenko, The Mathematical Theory of Optimal Processes, Interscience Publishers, 1962.

2. W.R.Shelton, Man's Conquer of Space, National Geographic Society, Washington, D.C., 1968.

3. M. Eleccion, Associate Editor, 'Military and Aerospace,' IEEE Spectrum, pp. 80-83, Jan. 1976.

4. G.W.L.McAdam, 'Optimization of Radar Scanning,' CRC and Defense Internal Report, File CRC 9574(rct)P, 1980-1981.

5. J.M.Murray and K.L.Teo, 'On a Computational Algorithm for a Class of Optimal Control Problems Involving Discrete Time Delayed Arguments,' J. Austral. Math. Soc. B20, pp. 315-343, 1978.

6. R.R.Bate, D.D.Mueller, and J.E.White, Fundumentals of Astrodynamics. New York : Dover, pp 397-407, 1971.

7. P.E.Elyasberg, Introduction to the Theory of Flight of Artificial Earth Satellites (translated from Russian), Israel Program of Scientific Translations, Jerusalem, 1967.

8. L.S.Lasdon, S.K.Mitter, and A.D.Waren, 'The Conjugate Gradient Method for Optimal Control Problems,' IEEE Trans. on Automat. Contr., vol. AC-12, No.2, pp. 132-138, April 1967.

9. M.R.Hestenes, Conjugate Direction Methods in Optimization, New York : Springer-Verlag, Chap.IV, 1980.

10. Dimitri P. Bertsekas, 'Partial Conjugate Gradient Methods for a Class of Optimal Control Problems,' IEEE Trans. on Automat. Contr., vol. AC-19, No.3, pp. 209-217, July 1974.

11. Y.C. Chen, G.W.L. McAdam and N.U. Ahmed, 'Optimal Scanning Policy for Phased Array Radar,' (to appear) IEEE Trans. Aerosp. Electron. Syst.

12. M.I.Skolnik, Introduction to Radar Systems, McGraw Hill Book Co., 2 nd edition, Chap.8, 1980.

13. D.P. Meyer and H.A. Mayer, Radar Target Detection, Academic Press, New York and London, pp.318, 1973.

14. N.U.Ahmed and Y.C.Chen, 'First Report on Research and Design of Optimal Scanning Procedure for Phased Array Radar,' Communication Research Center and Department of Defence, DSS-OSV80-00026, Jan. 1981.

15. N.U.Ahmed and Y.C.Chen, 'Final Report on Research and Design of Optimal Scanning Procedure for Phased Array Radar,' Communication Research Center and Department of Defence, DSS-OSV80-00026, March 1981.

16. N.U.Ahmed and A.K.Sen,'Control of Synchronous Orbit Using Lyapunov Stability Theory,' IEEE Trans. Aerosp. Electron. Syst., vol. AES-17, No.1, pp.48-53, Jan. 1981.

17. Y.C. Chen and N.U. Ahmed, 'Optimal Orbital Control Scheme for Synchronous Satellite,' (to appear) IEEE Trans. Aerosp. Electron. Syst.

18. R.Holdaway, 'Satellite Orbit Control Incorporating Attitude Control Techniques,' VII IFAC Symposium on Automatic Control in Space, Vol.2: Edited by E.Gottun, May 17-21, 1976.

19. P.H.W.Cheng, N.D.Geoganas, and J.A.D.Holbrook, 'Kalman Estimation and Control of Dual-Spin Satellites,' IEEE Trans. Aerosp. Electron. Syst., vol. AES-13, No.3, pp.236-245, May 1977.

20. N.U.Ahmed and A.K.Sen,'Stability and Control of a Trap Mode in Syvchronous Orbit,' IEEE Trans. Aerosp. Electron. Syst., vol. AES-16, No.4, pp.474-480, July 1980.

VITA

Name : Yao - Chi Chen

Born : Jan. 30, 1956, Calcutta, India

Education :

High School : Yuet Wah College, Macau

University : University of Ottawa, Ottawa, Ont.
B.A.Sc. in Electrical Engineering, 1980.