NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

NUMERICAL ALGORITHM FOR A CLASS OF OPTIMAL CONTROL PROBLEMS WITH APPLICATIONS TO ROBOTICS AND ECOLOGY

by

TIJANI SELLAMI

A thesis
presented to the University of Ottawa
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in
The Department of Electrical Engineering
Faculty of Engineering

ISBN 0-315-46853-X

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

To

my parents Aifa S. and Abdellah O.

ABSTRACT

A methodology has been developed for computing the time optimal controls for a system governed by a set of linear or nonlinear ordinary differential equations. The method has been used to compute the optimal controls for two types of robots : Cylindrical robot and a robot with an articulated arm with two links and an ecological system. The method is based on successive approximation of the time optimal controls by controls which are optimal for a suitable sequence of terminal problems. These later problems are easily solved by conjugate gradient technique. The method avoids solving nonlinear two-point-boundary-value problems thereby gaining some computational simplicity. Further, if a system is not even controllable our method will still provide a control that takes the system to a point nearest to the target. In this case it is not even necessary to assume the existence of time optimal controls.

Techniques of optimal control theory have been utilized in identifying the parameters of a physical system. For illustration these results are then used to determine the unknown of certains robotics and ecological systems.

ACKNOWLEDGEMENTS

The author wishes to express his indetness and deep gratitude to his advisor prof. N.U. Ahmed for his generous encouragement, understanding and guidance throughout this work, without which this thesis would not have been possible.

Special thanks are also due to Dr. T. Dabbous for his helpful suggestions and discussions.

Special thanks to the members of the system and control group, especially to Mr. S.S. Lim and Mr. L. Peng, for their suggestions and discussions.

The financial assistance of the Tunisian Government during the period of this research is gratefully acknowledged.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathcal{U}$=admissible set of controls

H=Hamiltonian

ψ=adjoint vector

$x^u(t)$=response of the state equations to the control u at time t

x_d=desired state

$\mathcal{A}$=attainable set

ρ_H=Hausdorff metric

F(t,x)=contingent function

$\varphi(t)$=minimum distance between the attainable set and the target

$K(R^n)$=class of all compact subsets of R^n.

F=force

T=torque

u=control vector

x=state vector

CHAPTER I

INTRODUCTION

In recent years much attention has been focused upon optimizing the behavior of systems described by a set of differential equations. The search for the control which attains the desired objective while minimizing (or maximizing) a defined system criterion constitutes the fundamental problem of optimization theory. A particular problem of optimization may concern with the minimization of the time required to drive a system from one desired state x_0 to some required target state x_d; this is known as the time optimal control problem. Another class of optimization may deal with the identification of some unknown parameters introduced in the system equations. The main aim of this thesis is to study numerically the last two problems for a class of systems governed by a set of ordinary differential equations.

1.1 TIME OPTIMAL CONTROL

In the time optimal control problem, it is of interest to find the best control in the set of admissible controls that drives the system from one desired state to another in minimum time. It was LaSalle [29] who first gave a complete solution to this problem. In the last two decades the interest in time optimal control has increased significantly (see for example [1-13]). The time optimal control theory has been developed in the literature and has been used in many applications such as industrial robot.

The time optimal control of a system of rigid bodies connected in series by singledegree of freedom joint is studied in [1]. A suboptimal feedback control, which provides a close approximation of the optimal control is developed. The suboptimal control is expressed in terms of switching curves for each of the controls. These curves are obtained from the linearized equations of motion for the system. Approximations are made for the effects of gravity loads and angular velocity terms in the nonlinear equations of motion. However, the possibility of singular time-optimal controls was not investigated.

About ten years later, the interest in time-optimality of robot motions increased significantly [2]-[6]. In [2], a theoretical procedure has been developed for comparing the performance of arbitrary selected admissible controls among themselves and with the optimal solution of a nonlinear optimal control problem. A recursive scheme converging to the time optimal solutions was presented. The approach has been applied to the approximately optimal control of a trainable manipulator with seven degrees of freedom, where the controller is used for motion coordination and optimal execution of objecthandling tasks. A similar, adaptive scheme for a two-link robot was given in [3]. In [4], the optimization problem was presented completely and the available mathematical design tools were discussed. In [5], an approximate time-optimal controller was combined with a linear regulator for the final phase of the trajectory. The dynamic programming method of synthesis of optimal trajectories was treated in [6]. Since the complexity of the necessary calculations for time-optimal controls is considerable, several schemes for simplification were proposed in [6]. The only method found to be suitable for such a complex optimization was based on dynamic programming. An algorithm for determining optimal velocity distribution for a given manipulator tip trajectory was elaborated in detail. The problem of time-optimal motions along arbitrary prespecified partial trajectories was studied in [7] and [8]. In

[7], a solution has been presented to the problem of minimizing the cost of moving a robotic manipulator along a specified geometric path subject to input torque/force constraints, taking the coupled, nonlinear dynamics of the manipulator into account. The proposed method uses dynamic programming to find the positions, velocities, accelerations, and torques that minimize the cost. As a numerical example, the path planning method was simulated for a two-jointed robotic manipulator. In [8], an optimal control policy which results in minimum-time motion for a robotic manipulator along any predetermined path in three dimensional space was presented. The technique permits the manipulator user to specify completely the path of the arm. The special case of rectilinear motion was studied in [9]. In this method there is only one degree of freedom, (the acceleration along the trajectory). The problem of time-optimal control motion of a load by a two-link manipulator is examined. The control functions are moments of forces applied to the axes of the manipulator joints and are limited in absolute values. Optimal and quasi-optimal controls were derived for rectilinear motion of the center of mass of a load from one point to another in such a way that the system becomes quiescent following the completion of the process. It was assumed that the mass of the manipulator is negligibly small in comparison to the mass of the load and that the manipulator links have equal lengths. The method of parameter optimization was used in [10] and [11]. It reduces the optimal control problem to the problem of finding a suitable parameterization for the control variables. In [12], a simple algebraic solution is obtained to an optimal control problem in joint space for a general robotic manipulator. Open-loop and closed-loop suboptimal solution were found by solving algebraic equations.

Recently, Geering et al [13] have presented a complete analysis for time optimal motions of three types of industrial robots: robot with cylindrical coordinates, robot with spherical coordinates and a robot with an articulated arm. Techniques of optimal control theory have been utilized in the time optimal control problem. The maximum principle was used to formulate the problem as two-point boundary-value differential equations. In order to solve this problem the shooting method was used. It has been shown also that in the case of the IBM articulated arm robot, the shooting method alone did not work. Therefore, both the parameter optimization and the shooting techniques were utilized in order to find the switching times for the controls, starting with various control parametrizations, minimizing the problem duration and using the penalty approach for the prescribed final state. Convergence of the shooting algorithm was achieved with the Lagrange multiplier of the penalty term of the previous step as an initial guess for the final costate vector. In the other hand, to solve the optimal control problem, another popular technique so called gradient has been widely considered.

The gradient method consists of updating the controls using the gradient vector, in a way such that successive iterates produce maximum reduction of the cost. This technique has been applied to several optimal control problems by Kelley [15, 16], and Bryson and Denham [17]. Various modifications to include penalty functions and other methods of treating equality and inequality constraints have been presented in [18]-[24]. The gradient method has the ability to generate successively improved trajectories with very poor starting values. However, they tend to converge slowly as convergence is approached. To improve the convergence of the gradient method near the optimal trajectory, the conjugate gradient method can be used to update the controls. The convergence of the gradient and conjugate gradient methods for the singular, finite-dimensional quadratic minimization problems have been developed in [30]. A class of associated nonsingular quadratic problems is defined to show that the gradient method has slower convergence on singular problems than on corresponding nonsingular approximations to the singular problems while the conjugate gradient method has more rapid convergence.

In this thesis, we have proposed a simpler technique for solving time optimal control problems without using the shooting method. Our algorithm is based on the conjugate gradient technique computing the optimal controls for a class of simpler terminal control problems over an arbitrary but variable time interval giving the distance of the attainable set to the desired target as a function of time and then locating the first minimum of this function over the given interval. If the first minimum is zero, then the corresponding time is the optimal time and the associated control is the time optimal control. In case there is no time optimal control our method still provides a control that takes the system to a point in the attainable set closest to the target. Thus our method does not require the assumption of existence of time optimal controls and it is also insensitive to the presence or absence of singular arcs. An added advantage of the algorithm is that it also provides a numerical technique for direct verification of controllability of the system. Our method also avoids the complexity of solving two-point boundary-value problems thereby gaining some computational simplicity.

In order to illustrate the usefulnes of the proposed method, the algorithm has been used to compute the time optimal control of two robots : Cylindrical robot and a robot with an articulated arm with two links and an ecological system.

1.2 PARAMETER IDENTIFICATION

An important and essential aspect of modelling any physical system is the identification of parameters in the model equation.

Techniques of optimal control theory have been utilized in identification of parameters in a system described by a set of differential equations. Note that the identification problem could be considered as a special case of the general control problem [Ahmed 14].

The time optimal control algorithm developed in this thesis can be easily modified to compute the unknown parameters introduced in the differential equations which describe the evolution of a physical system. The modified algorithm can determine the unknown parameters so that the corresponding response of the model equation approximates as closely as possible the actual response of the physical system. For numerical illustration, the modified algorithm has been applied to two robots and an ecological system.

1.3 MAJOR CONTRIBUTION OF THE THESIS

Development of an efficient algorithm that can be used for computing time optimal controls as well as the unknown parameters for a class of systems governed by ordinary differential equations applied to robotics and ecology.

1.4 OUTLINE OF THE THESIS

In chapter II, a simple but efficient algorithmm and its theoretical basis for computing the time optimal control are developed. Also, the necessary conditions of optimality for a general control problem and a brief discussion of the the gradient technique and the shooting methods are provided.

In chapter III, we consider the application of the proposed algorithm to two robots and an ecological system. In chapter IV, the algorithm developed in chapter II is modified to compute the unknown parameters introduced in the differential equations which describes the behavior of the system. In order to illustrate the effectiveness of the (modified) algorithm, we have considered the identification problem for practical systems (robotic and ecology). Finally, in chapter V conclusions of the thesis are presented.

CHAPTER II

TIME-OPTIMAL CONTROL AND ALGORITHM

2.1 INTRODUCTION

The behavior of a physical system over a certain time interval is generally described by a set of equations. These equations represent the mathematical relations between the states and the inputs to the system. Each system has one or more control variables (inputs) which can be altered in such a way as to create a desired response from the system. These controls may not be known a priori and in order to determine their magnitudes it is necessary to solve a control problem.

In many Engineering issues, some of the control problems seek the control variables that can transfer a system from an arbitrary fixed initial state to another desired in minimum time. This is known as the time optimal control problem. The solution of such problem requires the solution of a two-point boundary-value problem which is a difficult task.

In this chapter, we propose a simple but efficient algorithm for solving the time optimal control problem without using the shooting method. The method is based on the conjugate gradient technique. It avoids the solution of a two-point boundary-value problem and it consists of computing the optimal controls for a class of simpler terminal control problems over an arbitrary but variable time interval giving the distance of the attainable set to the desired target as a function of time and then locating the first minimum over the given interval. If the first minimum is zero, then the corresponding time is the optimal time and the associated control is the time optimal control. Further, if a system is not even controllable our method will still provide a control that takes the system to a point in the attainable set nearest to the target.

The algorithm and its theoretical basis are presented in section 5 and 3, respectively. The formulation of a general control problem and the corresponding necessary conditions of optimality are given in section 2. A brief discussion of the shooting method and the gradient technique is provided in section 4.

2.2 GENERAL CONTROL PROBLEM

Consider the system

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)), \quad 0 < t \leq T \\ x(0) &= x_0 \end{aligned} \tag{2.1}$$

with the cost functional

$$J(x,u) = \int_0^T l(t, x(t), u(t))dt + \phi(x(T)) \tag{2.1$'$}$$

Here x(t) $\in R^n$ denotes the state variable; $x_0 \in R^n$ is the initial condition and $u(t) \in R^m$ is the control variable. We assume that the control u takes values from a compact convex set U$\subset R^m$. Let

$$\mathcal{U} \equiv \{u : \ u \text{ is measurable; } u(t) \in U; t \geq 0\}$$

denote the set of admissible controls and

$$S \equiv \{(x,u) : \ x \in \mathrm{AC}([0,T], R^n), \ u \in \mathcal{U}, \ \dot{x} = f(t,x,u) \text{ a.e, } t \in [0,T]\}.$$

denote the set of admissible pairs where AC(I,R^n) denotes the class of all absolutely continuous functions on I=[0,T] with values in R^n. The basic control problem is to find a pair $(x^*, u^*) \in S$ such that

$$J(x^*, u^*) \leq J(x, u) \text{ for all } (x, u) \in S.$$

Apair satisfying this property is called optimal.

The following theorem gives the necessary conditions that an optimal pair must satisfy.

Theorem 2.1 (necessary conditions of optimality)

Consider the control problem (2.1) – (2.1)′ and suppose that f is measurable in t, C^1 in x and continuous in u and l is C^1 in all the variables. Define the Hamiltonian H: $[0,T] \times R^n \times R^n \times R^m \to R$ by

$$H(t, x(t), \psi(t), u(t)) = (f(t, x(t), u(t))\psi(t) + l(t, x(t), u(t)).$$

In order that the pair (x^*, u^*) be optimal, it is necessary that there exists a $\psi^* \in AC([0,T], R^n)$ such that the triple (x^*, u^*, ψ^*) simultaneously satisfies the following equations and inequalities

(i) $\dot{x}^*(t) = f(t, x^*, u^*)$ a.e. on $[0, T]$

$x^*(0) = x_0$

(ii) $\dot{\psi}^*(t) = -l_x(t, x^*(t), u^*(t)) - f_x{}'(t, x^*(t), u^*(t))\psi^*$

$\psi(T) = (\partial\phi/\partial x)(x(T))$

and

(iii) $H(t, x^*(t), \psi^*(t), u^*(t)) \leq H(t, x^*(t), \psi^*(t), u)$

for all $u \in U$

Remark 2.2

The time optimal control problem is a special case of the optimal control problem for $l = 1$ and $\phi = 0$.

2.3 TIME OPTIMAL CONTROL AND THEORETICAL BASIS FOR THE ALGORITHM

In this section we present a general statement of the time optimal control problem and develop the theoretical basis for our algorithm.

Consider the system of nonlinear differential equations (2.1). We assume that for each $u \in U$ and $x_0 \in R^n$, the system (2.1) has a unique solution $x = x(t, x_0, u)$ for all finite $t \geq 0$. Let $x_d \in R^n$ be another given point. Our goal is to find a control $u^* \in \mathcal{U}$ that drives the system (2.1) from the initial state x_0 to the desired target x_d in minimum time.

Let $\mathcal{U}(x_0, x_d)$ denote the set of all controls $\{u\} \in \mathcal{U}$ that transfer the system (2.1) from the state x_0 to the state x_d in finite time. Clearly if $\mathcal{U}(x_0, x_d)$ is empty the question of time optimal control does not arise. However if $\mathcal{U}(x_0, x_d) \neq \emptyset$ then one may ask the question as to whether or not there exists a time optimal control ?

Suppose $\mathcal{U}(x_0, x_d) \neq \emptyset$ and let $u \in \mathcal{U}(x_0, x_d)$ and define

$$\tau(u) \equiv \inf\{t \geq 0 : x(t, x_0, u) \equiv x^u(t) = x_d\}$$

Clearly the question of existence of time optimal control is equivalent to the question of existence of a $u^* \in \mathcal{U}(x_0, x_d) \subset \mathcal{U}$ such that

$$\tau(u^*) \leq \tau(u) \; \forall \; u \in \mathcal{U}(x_0, x_d) \subset \mathcal{U}.$$

In general the answer to the question of existence of time optimal control is negative. However if the contigent function $F(t,x) \equiv f(t,x,U)$ is convex for all $(t,x) \in [0,\infty) \times R^n$ and the system is controllable then one can prove the existence of time optimal controls. Given that a time optimal control exists, the *Pontryagin minimum principle* [24] gives a set of necessary conditions that an optimal control must satisfy.

For convenience of the reader we quote this result below.

Theorem 2.3 (Pontryagin's n.c of optimality)

In order that the pair $\{u^*, x^*\}$ be optimal with τ^* the optimal time it is necessary that there exists a $\psi^* \in AC([0,\tau^*], R^n)$ such that the triple $\{u^*,\ x^*,\ \psi^*\}$ simultanuousely satisfies the following set of differential equations and inequalities

$$\dot{x}^* = f(t,x^*,u^*), \quad t \in [0,\tau^*], x^*(0) = x_0, x^*(\tau^*) = x_d$$

$$\dot{\psi}^* = -f_x(t,x^*,u^*)\psi^*$$

$$H(t,x^*(t),u^*(t),\psi^*(t)) \leq H(t,x^*(t),v,\psi^*(t))$$

for all $v \in U$ and for almost all $t \in\ I_{\tau^*} = (0,\tau^*]$, (2.2)

and

$$H(\tau^*,x^*(\tau^*),u^*(\tau^*),\psi^*(\tau^*)) + 1 = 0, \qquad \text{where } H = (f,\psi). \qquad \blacksquare$$

If a time optimal control exists then the above necessary conditions can be used to identify the controls that contain the optimal ones. In case the problem (2.2) has a unique solution, any control computed by use of the necessary conditions is the optimal control.

If the set $\mathcal{U}(x_0, x_d)$ is empty, or, in particular, if there is no time optimal control, then the necessary conditions have no meaning. In that case one may like to find a control that drives the system from the point x_0 to one that is closest to the target x_d. Indeed for each $t \geq 0$ let

$$\mathcal{A}(t) \equiv \{\xi \in R^n : \xi = x^u(t) \text{ for some } u \in \mathcal{U}\}$$

denote the attainable set. Under fairly general assumptions we can show that the set $\mathcal{A}(t)$ is compact for each finite $t \geq 0$, and that $t \to \mathcal{A}(t)$ is continuous in a suitable metric. Before we can prove this result we need the following definition.

Let $K(R^n)$ denote the class of all compact subsets of R^n. For $A \in K(R^n)$ define

$$\rho(z, A) = \inf\{\|z - a\|, a \in A\}.$$

Clearly for $z \in A$, $\rho(z, A) = 0$. For $A, B \in K(R^n)$ define

$$\rho_H(A, B) = \max\{\sup_{z \in B} \rho(A, z), \sup_{z \in A} \rho(z, B)\}$$

and note that $(K(R^n), \rho_H)$ is a metric space. The metric ρ_H is called the *Housdorff metric.*

Lemma 2.4

Suppose f is measurable in t for $t \geq 0$, and continuous in the rest of the variables and that U is a compact subset of R^r with the contingent function $F(t, x) = f(t, x, U)$

being a closed convex subset of R^n. Suppose there exists an $h \in L_1^{\text{loc}}$, possibly depending on the set U such that

$$\|f(t,x,u)\| \leq |h(t)|[1+\|x\|] \qquad \text{a.e}$$

for all $x \in R^n$ and $u \in U$. Then for each finite $t \geq 0$, the attainable set $\mathcal{A}(t)$ is compact and further the set valued map $t \longrightarrow \mathcal{A}(t)$, $t \geq 0$, is continuous in the Housdorff metric ρ_H.

Proof

Let x^u denote any solution of (2.1) corresponding to $u \in \mathcal{U}$. Then under the given assumptions it follows from Gronwall's lemma [14] that

$$(1+\|x^u(t)\|) \leq (1+\|x_0\|)\exp\left(\int_0^t |h(\theta)|\, d\theta\right) \text{ for } u \in \mathcal{U} \text{ and } 0 \leq t < \infty.$$

Hence, for $t < \infty$, the attainable set $\mathcal{A}(t)$ is a bounded subset fo R^n. Further, one can show that for any compact interval $I = [0,T]$, $T < \infty$, the set of trajectories $\mathcal{X} \equiv \{\{x^u\} \in C(I,R^n)$ satisfying (2.1) for some $u \in \mathcal{U}\}$ is a bounded subset of $C(I,R^n)$ and it is also equicontinuous. Hence by *Ascoli-Arzela* theorem [31] the set $\mathcal{X}$ is a relatively compact subset of $C(I,R^n)$. That the set $\mathcal{X}$ is closed follows from the fact that the contingent function $F(t,x) \equiv f(t,x,U)$ is closed and convex for $(t,x) \in I \times R^n$. This, combined with relative compactness, implies that the set $\mathcal{X}$ is a compact subset of $C(I,R^n)$. Hence for each $t \in I$, the set $\mathcal{A}(t) \equiv \{\xi \in R^n : \xi = x(t) \text{ for } x \in \mathcal{X}\}$ is closed. Thus $\mathcal{A}(t)$ is both closed and bounded and consequently compact. The continuity of the set valued map $t \longrightarrow \mathcal{A}(t)$ follows from the inequality

$$\rho_H(\mathcal{A}(t_1),\mathcal{A}(t_2)) \leq \beta \int_{t_1}^{t_2} |h(\theta)|d\theta, 0 \leq t_1 \leq t_2 \leq T, \tag{2.3}$$

where β is a finite positive number depending only on the bound of the set $\mathcal{X}$ restricted to the interval $[0,t_2]$. Since $0 \leq T < \infty$ but otherwise arbitrary, the lemma is proved. ∎

The following result is an immediate consequence of the above lemma.

Theorem 2.5

Consider the system (2.1) with x^u denoting its solution corresponding to $u \in \mathcal{U}$. Let g be any real valued continuous function on R^n and suppose the hypotheses of lemma (2.4) hold. Then for any finite $\tau > 0$, there exists an $u^* \in \mathcal{U}$ such that $g(x^{u^*}(\tau)) \le g(x^u(\tau))$ for all $u \in \mathcal{U}$. That is

$$g(x^{u^*}(\tau)) = \inf\{g(\xi), \xi \in \mathcal{A}(\tau)\}.$$

Proof

Since g is continuous and $\mathcal{A}(\tau)$ is compact for $\tau < \infty$, g restricted to $\mathcal{A}(\tau)$, attains both its minimum and maximum on $\mathcal{A}(\tau)$. Hence there exists an $u^* \in \mathcal{U}$ satisfying the assertion of the theorem.■

For each finite $\tau \ge 0$, define

$$\varphi(\tau) = \rho(x_d, \mathcal{A}(\tau)) = \inf\{\rho(x_d, \xi), \xi \in \mathcal{A}(\tau)\}. \tag{2.4}$$

Since for each $0 \le t < \infty, \mathcal{A}(t)$ is compact by lemma (2.4), and the function $\xi \to \rho(x_d, \xi)$ is continuous on R^n, it follows from theorem (2.5), that $\rho(x_d, \xi), \xi \in \mathcal{A}(t)$, attains both its maximum and minimum on $\mathcal{A}(t)$. Hence $\varphi(\tau)$, $0 \le \tau < \infty$, is well defined and it gives the distance of the points in $\mathcal{A}(\tau)$ nearest to the target x_d. Since $\mathcal{A}(\tau)$ is compact, the set

$$\mathcal{N}_0(\tau) \equiv \{\xi \in \mathcal{A}(\tau) : \|\xi - x_d\| = \varphi(\tau)\} \tag{2.5}$$

is nonempty and there is at least one control $u^* \in \mathcal{U}$ such that $x^{u^*}(\tau) \in \mathcal{N}_0(\tau)$. Hence the control u^* must satisfy the necessary conditions of optimality for the following terminal control problem:

$$\dot{x}(t) = f(t,x,u), 0 \leq t \leq \tau < \infty, \; x(0) = x_0, \tag{2.6}$$

$$J(\tau, u) = (1/2)\|x^u(\tau) - x_d\|^2 \equiv \min. \tag{2.7}$$

The corresponding necessary conditions of optimality is given by the Pontryagin's minimum principle which is quoted below for convenience of reference.

Theorem 2.6

In order that $u^* \in \mathcal{U}$ or equivalently the pair $\{u^*, x^*\}$ be optimal for the terminal control problem (2.6), (2.7) it is necessary that there exists $\psi \in AC(I_\tau, R^n), I_\tau = [0, \tau]$, (absolutely continuous functions from I_τ to R^n) such that the triple $\{u^*, x^*, \psi^*\}$ satisfies the following differential equations and inequalities:

$$\text{(i) } \dot{x}^* = f(t, x^*, u^*) \text{ for a.e } t \in (0, \tau], \; x^*(0) = x_0, \tag{2.8i}$$

$$\text{(ii) } \dot{\psi}^* = -f_x{}'(t, x^*, u^*)\psi^* \text{ for a.e } t \in [0, \tau), \psi^*(\tau) = x^*(\tau) - x_d, \tag{2.8ii}$$

and

$$\text{(iii) } H(t, x^*(t), u^*(t), \psi^*(t)) \leq H(t, x^*(t), v, \psi^*(t)) \text{ a.e in } [0, \tau], \tag{2.8iii}$$

and for all $v \in U$. ■

In general the system (2.1) may not be controllable from x_0 *to* x_d, or the time optimal control may not have a solution but, by virtue of theorem (2.5) and the subsequent discussion, the problem (2.6) -(2.7) has always a solution. Therefore the necessary conditions of optimality (2.8i)-(2.8iii) are nonvacuous and can be used to determine $\varphi(\tau)$ (see 2.4) or equivalently the controls satisfying (2.7). This is particularly true whenever the necessary conditions (2.8) are also sufficient or they have a unique solution. In case the Hamiltonian is convex in the control variable, the above necessary conditions are also sufficient. For this the reader is refered to [Clarke, Theorem 5.4.2, pp.219-220]. If the original time optimal control problem has a solution the set

$$I_0 \equiv \{t \geq 0 : \varphi(t) = 0\} \tag{2.9}$$

is nonempty and hence $\tau^* \equiv \inf(I_0)$ exists and there is a control $u^* \in \mathcal{U}$ such that $x^{u^*}(\tau^*) = x_d$. If on the otherhand $\varphi(t) > 0$ for all $t \geq 0$ then the system is not controllable with respect to the pair $\{x_0, x_d\}$. In this situation one may be content with a control that steers the system from x_0 to an attainable point, that is a point $\xi^* \in \bigcup \mathcal{A}(\tau)$, $\tau \geq 0$ closest to the target x_d. Indeed since $t \rightarrow \mathcal{A}(t)$ is continuous from $[0, \infty)$ to $K(R^n)$ in the Housdorff metric ρ_H, and

$$|\varphi(t) - \varphi(s)| \leq \rho_H(\mathcal{A}(t), \mathcal{A}(s)) \tag{2.10}$$

for all $t, s \in [0, \infty)$, the function φ is continuous and non-negative and therefore has a minimum over any given compact interval $I_T = [0, T], T < \infty$.

Letting $m^0(T)$ denote the minimum of φ over the interval $I_T \equiv [0, T]$, and $\tau^0(T)$ the first time in I_T at which φ equals $m^0(T)$ it is clear that $T \rightarrow m^0(T)$ is a monotone nonincreasing function of T and that $\tau^0(T) \leq T$. The algorithm developed in the paper and presented in the following section is able to compute $m^0(T)$ and $\tau^0(T)$ for each finite T.

Note that if T^* is the first time at which $m^0(T^*) = 0$ (in the event time optimal control exists) then $\varphi(T^*) = 0$, and $m^0(T) = 0$ for all $T \geq T^*$, though $\varphi(T), T > T^*$, need not be zero unless $\mathcal{A}(T^*) \subset \mathcal{A}(T)$ for all $T \geq T^*$.

As indicated earlier, the time optimal control problem has been (numerically) solved in the literature by the use of the shooting method which involves the solution of two-point boundary value problem. It should be noted, however, that the application of gradient technique to the (numerical) solution of time optimal control has not been considered yet in the literature. In the next section we present, for convenience, a brief description for both shooting and gradient techniques.

2.4 SHOOTING METHOD AND GRADIENT TECHNIQUE

In this section, we present a brief discussion of the shooting method and the gradient technique.

2.4.1 Shooting Method

Two-point boundary-value problems associated with systems of linear and nonlinear ordinary differential equations occur in many branches of mathematics , engineering, and the various sciences. In these problems, conditions are specified at the endpoints of an interval, and a solution of the differential equations over the interval is sought which satisfies the given endpoint conditions.

Consider the control problem

$$\text{minimize}\{J(u) = \int_{t_1}^{t_2} l(t, x(t), u(t))dt\} \tag{2.11}$$

subject to the dynamic constraint

$$\dot{x}(t) = f(t, x(t), u(t)), \qquad u \in \mathcal{U} \text{ and } t_1 < t_2$$
$$x(t_1) = x_0 \tag{2.12}$$
$$x(t_2) = x_1$$

where $x(t) \in R^n$ and $u(t) \in R^m$ denote the state and control variables, respectively. The initial and terminal state vectors are given by $x_0 \in R^n$ and $x_1 \in R^n$, respectively.
Define the Hamiltonian $H(t, x, \psi, u)$ by

$$H(t, x, \psi, u) = l(t, x(t), u(t)) + (f(t, x(t), u(t)), \psi(t)) \tag{2.13}$$

where the costate vector $\psi(t) \in R^n$ is the solution of the adjoint system

$$\dot{\psi}(t) = -f_x{}'(t, x(t), u(t))\psi(t) - l_x(t, x(t), u(t)), \qquad t_1 \le t < t_2$$
$$\psi(t_2) = 0 \tag{2.14}$$

Suppose there exists a function $\eta : R^n \times R^n \to U$ such that

$$M(x, \psi) \equiv \max_{u \in U} H(t, x, \psi, u) = H(t, x, \psi, \eta(x, \psi))$$

for all $(x, \psi) \in R^n \times R^n$.
Using the function η and the canonical equations $\dot{x} = H_\psi$, $\dot{\psi} = -H_x$, we can transform the original control problem (2.11)-(2.12) to obtain

$$\dot{x} = f_1(x, \psi), \quad x(t_1) = x_0, \; x(t_2) = x_1$$
$$\dot{\psi} = f_2(x, \psi) \tag{2.15}$$

where

$$f_1(x,\psi) = f(x,\eta(x,\psi))$$

and

$$f_2(x,\psi) = -f_x{}'(x,\eta(x,\psi))\psi - l_x(x,\eta(x,\psi))$$

The system(2.15) consits of 2n nonlinear sets of differential equations with n initial and n terminal conditions. These problems are known as two-point boundary-value problems. Generally, these problems cannot be solved analytically and a numerical approach must be provided.

In the shooting method, a set of values of the unspecified conditions is assumed ($\psi(t_1) = \xi$), and the differential equations (2.15) are numerically solved to obtain $x = x(t, x_0, \xi)$ and $\psi = \psi(t, x_0, \xi)$. If the computed terminal values satisfy the specified terminal conditions, the problem is completely solved. If they do not, one has to introduce an error function

$$E(t_2,\xi) = \|x(t,x_0,\xi) - x_1\|^2 \tag{2.16}$$

to adjust the missing initial conditions.

The original control problem is then reduced to that of minimizing the error function. In order to minimize the error function one uses an iterative process based on the gradient technique. The computation of the gradient to update the missing initial conditions is extremely laborious and time consuming. In this case, the gradient vector $g_\xi \in R^n$ is given by

$$g_\xi = \lim_{\xi\to 0}(E(t_2,\xi+\epsilon) - E(t_2,\xi))/\epsilon$$

The next iterative set of the missing conditions is taken as

$$\psi(t_2) = \xi + \alpha g_\xi$$

where $\alpha \in R$ is sufficiently small so that

$$E(t_2, \xi + \alpha g_\xi) \leq E(t_2, \xi)$$

The procedure of updating the missing initial conditions will be repeated until the error function cannot decrease anymore.

2.4.2 Gradient Method

Consider the control problem (2.11)-(2.12).

The gradient method consists of updating the control using the gradient vector, in a way successive iterates produce maximum reduction of the cost. The iterative scheme attempts to generate a sequence of controls u_i which eventually converge to a minimizing control u^*. Further, it involves the iterative rule

$$u_{i+1} = u_i + \alpha_i s_i \tag{2.17}$$

An initial control $u_0 \in R^m$ is chosen arbitrary. At each iteration a direction $s_i \in R^m$ is determined and a step size $\alpha_i \in R$ is chosen sufficiently small such that

$$J(u_i + \alpha_i s_i) \leq J(u_i + \lambda s_i) \text{ for all } \lambda$$

In the gradient method $s_i = -g_i$, where $g_i \in R^m$ is given by

$$g_i(t) = (\partial H/\partial u)(t, x_i(t), \psi_i(t); u_i(t)) \tag{2.18}$$

with H is the Hamiltonian given by (2.13).

The gradient method has the property that the computed solution tends to converge to a local minimum. Thus once a minimum is attained, it is necessary to change the initial guess and search for other minima in the control space.

The control sequence generated by the gradient method does converge to a local minimum even if the initial guess is poor. However, convergence becomes slower as the minima is approached. In computer application, convergence of the gradient method can be substantially improved by using conjugate gradient technique for updating the controls. In the conjugate gradient technique, s_i is taken as

$$s_i = -g_i + (< g_i, g_i > / < g_{i-1}, g_{i-1} >)s_{i-1} \qquad \text{with } \; s_0 = -g_0$$

where $< a, b >= a'b$ denotes the inner product in R^m, for $a,\ b \in R^m$.

2.5. ALGORITHM

Using the necessary conditions of optimality presented in section 3, we develop an iterative algorithm to compute the time optimal control. The algorithm is based on the conjugate gradient technique and it is presented as follows

Step 0 : Let τ be an arbitrary positive number (equations (2.6)-(2.7))

Step 1 : Guess $u_1 \in \mathcal{U}$ and set n=1.

Step 2 : Solve the initial value problem (2.6) with $u(t) = u_n(t)$ giving $x_n(t) = x(t, x_0, u_n)$.

Step 3 : Using the data x_n and u_n, solve the adjoint system

$$\dot{\psi}(t) = -f_x{}'(x_n(t), u_n(t))\psi(t)$$

$$\psi(\tau) = x_n(\tau) - x_d$$

backward in time to determine ψ_n.

Step 4 : Compute the gradient vector (equation 2.18)

Step 5 :

(i) If $n = 1$ set $s_n(t) = -g_n(t)$

(ii) If $n \neq 1$ set $s_n(t) = -g_n(t) + (\|g_n(t)\|^2 / \|g_{n-1}(t)\|^2)\ s_{n-1}(t)$

Step 6 :

Step 6.1: If $g_n(t) \neq 0$, then modify $u_n(t)$ to $u_{n+1}(t) = u_n(t) + \epsilon s_n(t)$ by choosing $\epsilon > 0$ sufficiently small so that for all t, $u_{n+1}(t) \in U$ and $J(\tau, u_{n+1}) \leq J(\tau, u_n)$.

(i) If $u_{n+1}(t) \geq u_{max}$, for some t$\in [0,\tau]$, then $u_{n+1} = u_{max}$.

(ii) If $u_{n+1} \leq u_{min}$, for some t$\in [0,\tau]$, then $u_{n+1} = u_{min}$.

(Where the control constraints are $u_{min} \leq u(t) \leq u_{max}$, for all t.)

(iii) If $|J(\tau, u_{n+1}) - J(\tau, u_n)| \leq \delta$, for small $\delta > 0$, then set $\varphi(\tau) = J(\tau, u_n)$ and go to step7, otherwise set $u_n = u_{n+1}, n = n + 1$, and go to step2 .

Step 6.2: If at the nth stage $g_n(t) = 0$ for $t \in [0, \tau]$, then u_n is a local minimizing element of $J(\tau, u)$ and set $\varphi(\tau) = J(\tau, u_n)$.

Step 7 : Take $\Delta\tau > 0$, go to step1 and compute $\varphi(\tau + \Delta\tau)$ and $\varphi(\tau - \Delta\tau)$ by solving (2.6) and (2.7) replacing τ by $\tau \pm \Delta\tau$.

(i) If $\varphi(\tau - \Delta\tau) < \varphi(\tau)$, set $\tau = \tau - 2\Delta\tau$ and go to step1.

(ii) If $\varphi(\tau + \Delta\tau) < \varphi(\tau)$, set $\tau = \tau + 2\Delta\tau$ and go to step1.

(iii) Stopping criterion : If $\varphi(\tau+\Delta\tau) > \varphi(\tau)$ and $\varphi(\tau-\Delta\tau) > \varphi(\tau)$ for $\Delta\tau$ sufficiently small, then $\varphi(\tau)$ is a local minimum and stop.

Remark 2.7

In step 6, the step size ϵ to update the control is determined using the Lagrange Extrapolation. For more details, the reader is referred to appendix A.

2.6 SUMMARY

In this chapter, we have proposed a simple algorithm and its mathematical justification for the computation of the time optimal control. The method is based on approximation of time optimal controls by a sequence of controls which are optimal for a suitable class of terminal control problems. These later problems are easily solved by conjugate gradient technique. The algorithm avoids the solution of a two-point boundary-value problem which is a difficult task. Our method is also able to determine if a system is controllable or not. Further, Pontryagin's minimum principle of optimality for a general control problem and a brief discussion of the gradient technique and shooting method have been provided. Three numerical examples illustrating the efficiency of the algorithm will be presented in the next chapter.

CHAPTER III

NUMERICAL SIMULATION OF TIME OPTIMAL CONTROL

3.1 INTRODUCTION

In chapter II, the time optimal control problem (in a general set up) has been considered and the corresponding necessary conditions of optimality has been presented. Using the conjugate gradient technique, we have developed an iterative scheme with the help of which time optimal controls can be computed. In this chapter, we consider the application of this algorithm to two kinds of robots and an ecological system. This chapter is organized as follows:

The applications of the algorithm to the time optimal control problem of Automelec ACR robot with cylindrical coordinates and IBM articulated arm robot with two links are given in section 2 and 3, respectively. Dynamic models of the two robots were taken from [13]. The complete solution of the time optimal control of an ecological system is given in section 4. The ecological model has been developed in [25].

3.2 ROBOT WITH CYLINDRICAL COORDINATES

In order to maximize the productivity of the robot, the total assembly time must be minimized. In other words, both the layout of the component's stacks around the workpiece must be optimized and each of the steps of assembly must be executed in minimum time. The problem is equivalent to find a control in the set of admissible controls that minimizes both the motion of the robot to the component location and the transportation of that component to the location of assembly. However, the problem of time optimal motions of articulated robotic manipulators is computationally cumbersome, mainly because of their highly coupled and nonlinear dynamics.

The cylindrical robot has three degrees of freedom and its geometry is shown in Fig.3.1.

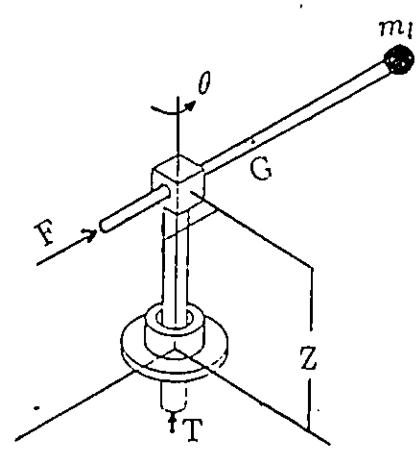


Fig.3.1. Sketch of the Cylindrical Robot.

The first degree of freedom is the azimuth rotation θ which is driven by a limited torque T. The second degree of freedom is the radial translation d which is driven by a limited force F. The third degree of freedom is the vertical translation Z. The remaining degrees of freedom are in the robot's hand. Their effects are neglected and the hand and the load are lumped into a point mass m_l.

Neglecting the friction, the cylindrical robot is governed by the following set of differential equations [13],

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= \{u_1(t) + m_k x_1(t) x_4^2(t) + m_l[d_0 + x_1(t)] x_4^2(t)\}/(m_k + m_l) \\
\dot{x}_3(t) &= x_4(t) \\
\dot{x}_4(t) &= \{u_2(t) - 2m_k x_1(t) x_2(t) x_4(t) - 2m_l[d_0 + x_1(t)] x_2(t) x_4(t)\}/M[x_1(t)]
\end{aligned} \tag{3.1}$$

with $x(t) \equiv [x_1(t), x_2(t), x_3(t), x_4(t)]' \in R^4$ and the control vector $u(t) \equiv [u_1(t), u_2(t)]' \in R^2$.

Where

x_1 =translation measured between the azimuth axis Z and the center of gravity G of the naked arm =d

x_2 =rate of the radial translation.

x_3 =azimuth rotation $=\theta$

x_4 =angular velocity of the azimuth rotation.

u_1 =F

u_2 =T

m_k =mass of the naked arm.

m_l =mass of the hand and the load.

M_z =mass moment of inertia of the robot without the arm, hand and load, measured with respect to the azimuth axis Z.

M_c =mass moment of inertia of the naked arm without the load, measured with respect to its center of gravity G.

d_0 =distance between the mass m_l and G.

$M[x_1]$ $=M_z + M_c + m_k x_1^2 + m_l[d_0 + x_1]^2$.

The control constraints are

$$|u_1(t)| \leq F_{\max}$$
$$|u_2(t)| \leq T_{\max}$$

Let us find admissible controls u_1^* and u_2^* satisfying the control constraints which transfer the robot from the fixed initial state [.15, 0, 0, 0] to the fixed final state [.15, 0, $\pi/2$, 0] in minimum time.

For numerical simulation we use the following parameters as given in the above reference,

$m_k = 3.7Kg$, $m_l = 4.6Kg$, $d_0 = .37\text{m}$, $M_z = .28\text{m}^2Kg$,

$M_c = .09\text{m}^2Kg$, $F_{\max} = 15N$, $T_{\max} = 5N\text{m}$.

Using the algorithm in chapter II, the optimal cost $\varphi(\tau)$, the optimal trajectories and controls $u_1^*(t)$ and $u_2^*(t)$ are computed as shown in Fig.3.2-3.4. The optimal time is τ^*=1.16 (sec) which coincides with the result of Geering et al [13]. In Fig.3.6, controls $u_1^*(t)$ and $u_2^*(t)$ are shown for different final times. The required time to compute the optimal time was about 44 minutes (in CPU time in AMBDAHL 5860).

The function $\varphi(\tau)$ is monotone decreasing and $\varphi(t)$=0 for all $t \geq \tau^*$. In this situation the attainable set $\mathcal{A}(t)$ is monotone noncontractive, that is,

$$\mathcal{A}(t_1) \subset \mathcal{A}(t_2) \text{ for } t_1 \leq t_2$$

The time optimal control is purely bang-bang for both of the control variables. Further, from Fig.3.2 and 3.5, it is clear that

$$\begin{aligned} u_1^*(t) &= F_{\max} \text{ for } \psi_2(t) < 0 \\ u_1^*(t) &= -F_{\max} \text{ for } \psi_2(t) > 0 \end{aligned} \tag{3.2}$$

and

$$\begin{aligned} u_2^*(t) &= T_{\max} \text{ for } \psi_4(t) < 0 \\ u_2^*(t) &= -T_{\max} \text{ for } \psi_4(t) > 0 \end{aligned} \tag{3.3}$$

The above results coincide with the theoretical ones, this is shown below.

The Hamiltonian function is given by

$$H = 1 + (f, \psi) = 1 + \psi_1(t)\dot{x}_1(t) + \psi_2(t)\dot{x}_2(t) + \psi_3(t)\dot{x}_3(t) + \psi_4(t)\dot{x}_4(t)$$

After $\dot{x}_i(t)$, i=1,...,4 take the values given in the system equation (3.1), it follows that the Hamiltonian is affine in the controls and it has the form

$$H = g(x, \psi) + g_1(x, \psi)u_1 + g_2(x, \psi)u_2$$

where

$$\begin{aligned} g_1(x, \psi) &= \psi_2(t)/(m_k + m_l) \\ g_2(x, \psi) &= \psi_4(t)/M[x_1] \end{aligned} \tag{3.4}$$

and $g(x, \psi)$ collects the remaining terms.

The optimal controls are obtained by globally minimizing the Hamiltonian with respect to the controls u_1 and u_2. Hence, the optimal controls are essentially determined by the sign of the switching functions g_1 and g_2. Since both of the denominators in (3.4) are positive, the sign of $\psi_2(t)$ and $\psi_4(t)$ are relevant, respectively.

Remark 3.1

For a given set of masses m_l, one has to compute the control policies for each mass separetely and store them. Each time the robot load changes one has to recall the correponding policy.

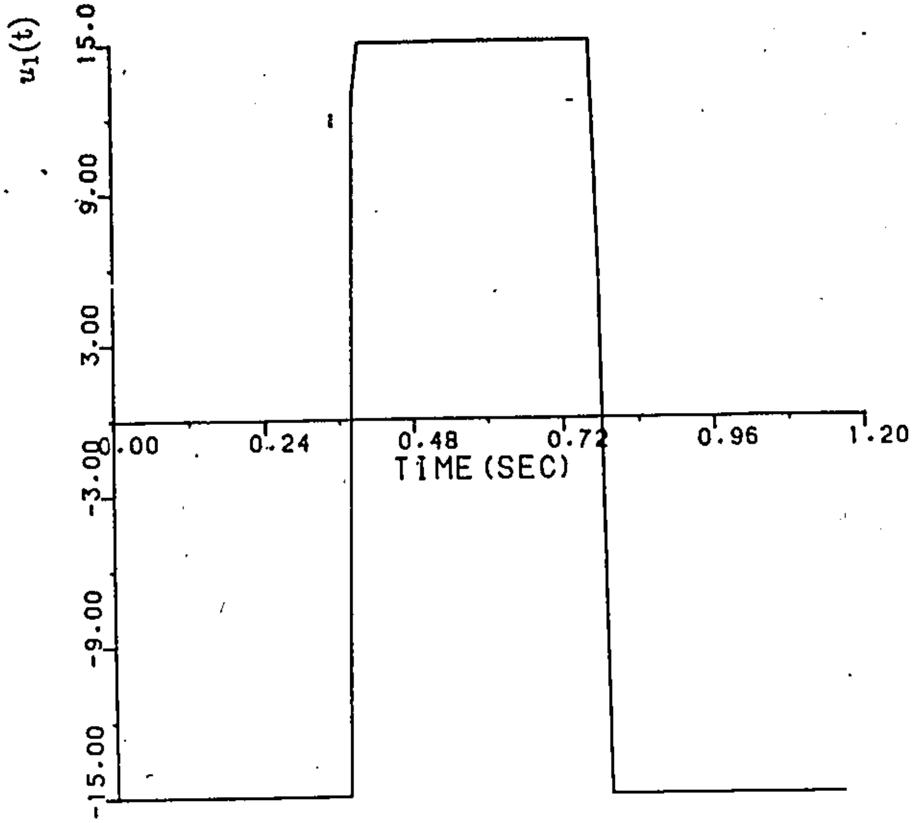


Fig.3.2.a. Optimal Control u_1^*.

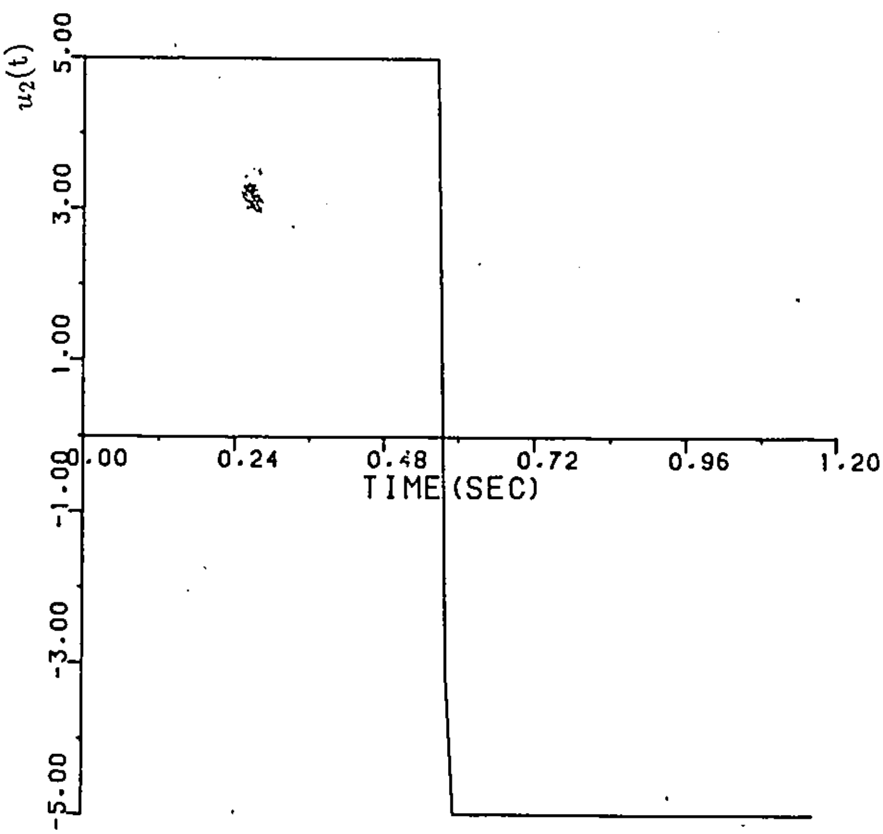


Fig.3.2.b. Optimal Control u_2^*.

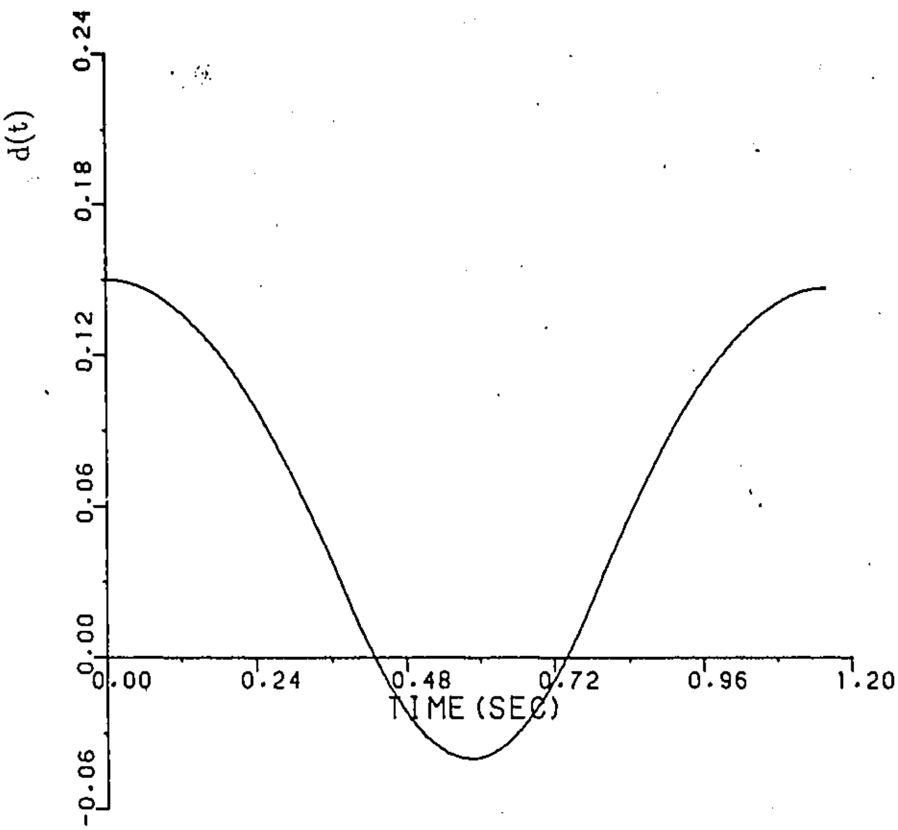


Fig.3.3.a. Optimal State d.

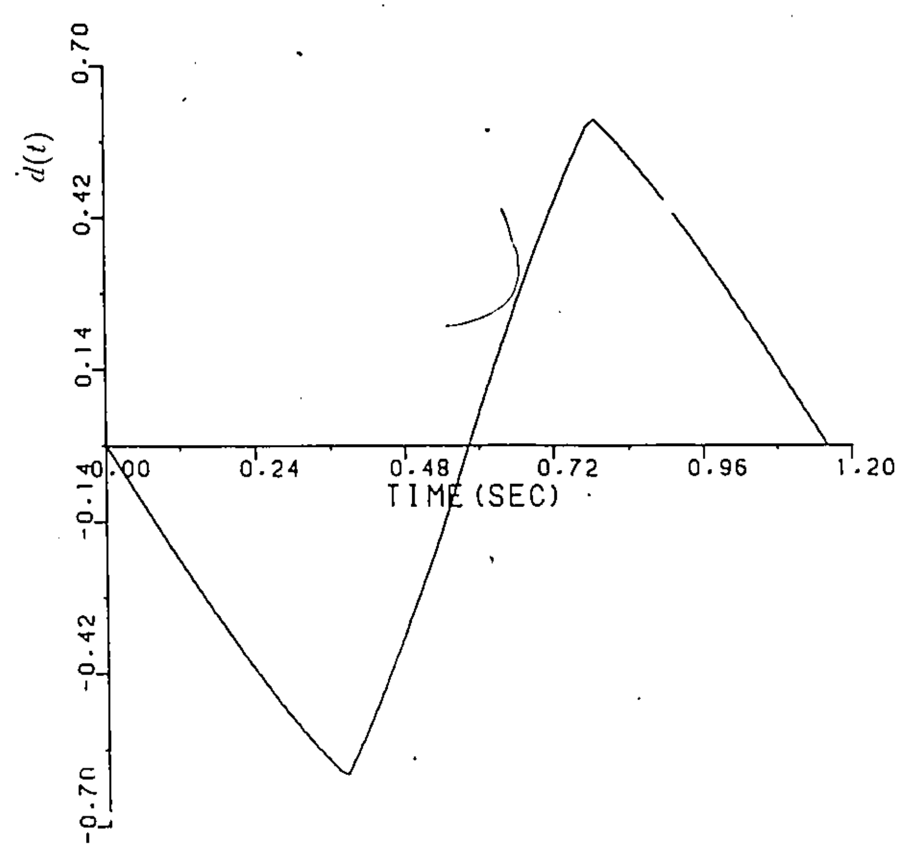


Fig.3.3.b. Optimal State ḋ.

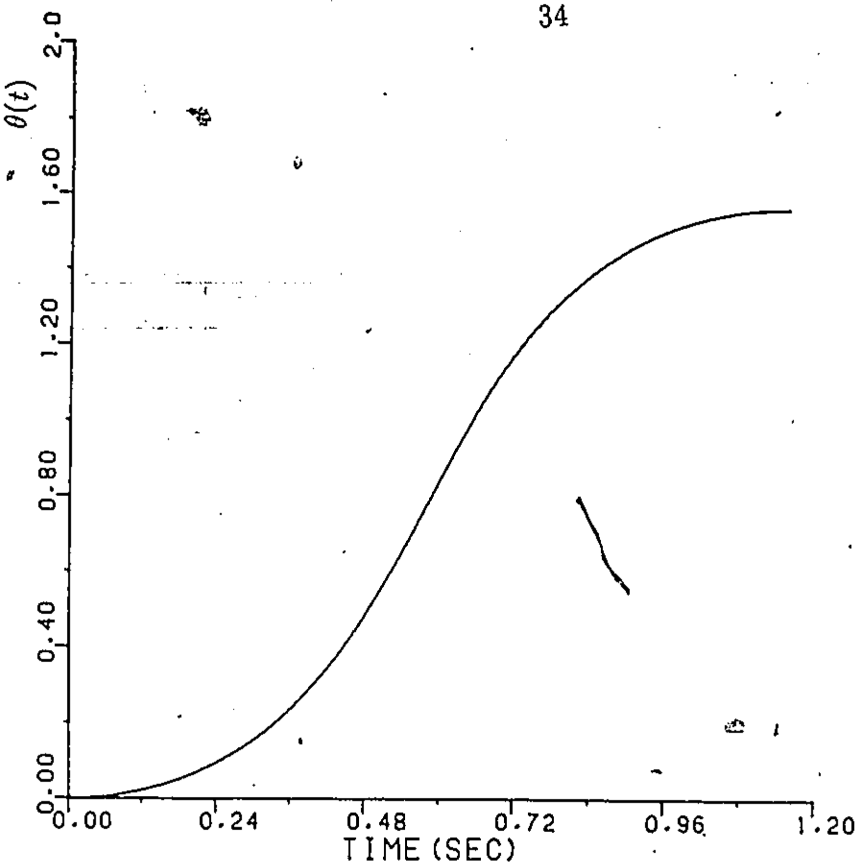


Fig.3.3.c. Optimal State θ .

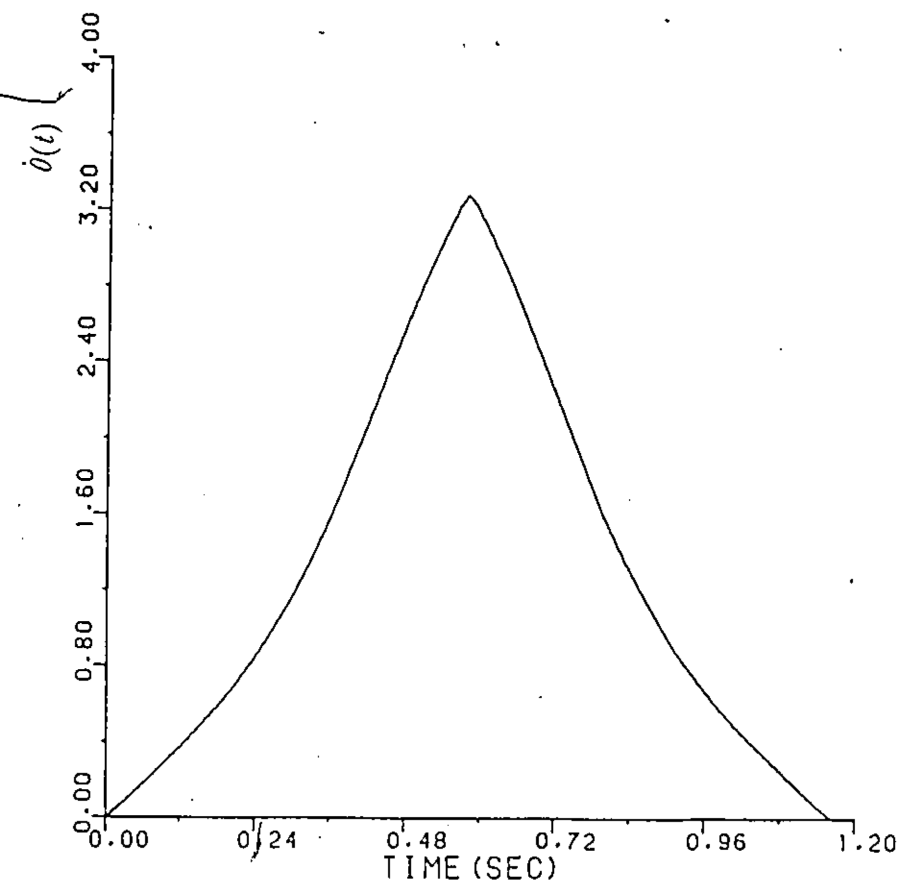


Fig.3.3.d. Optimal State $\dot{\theta}$

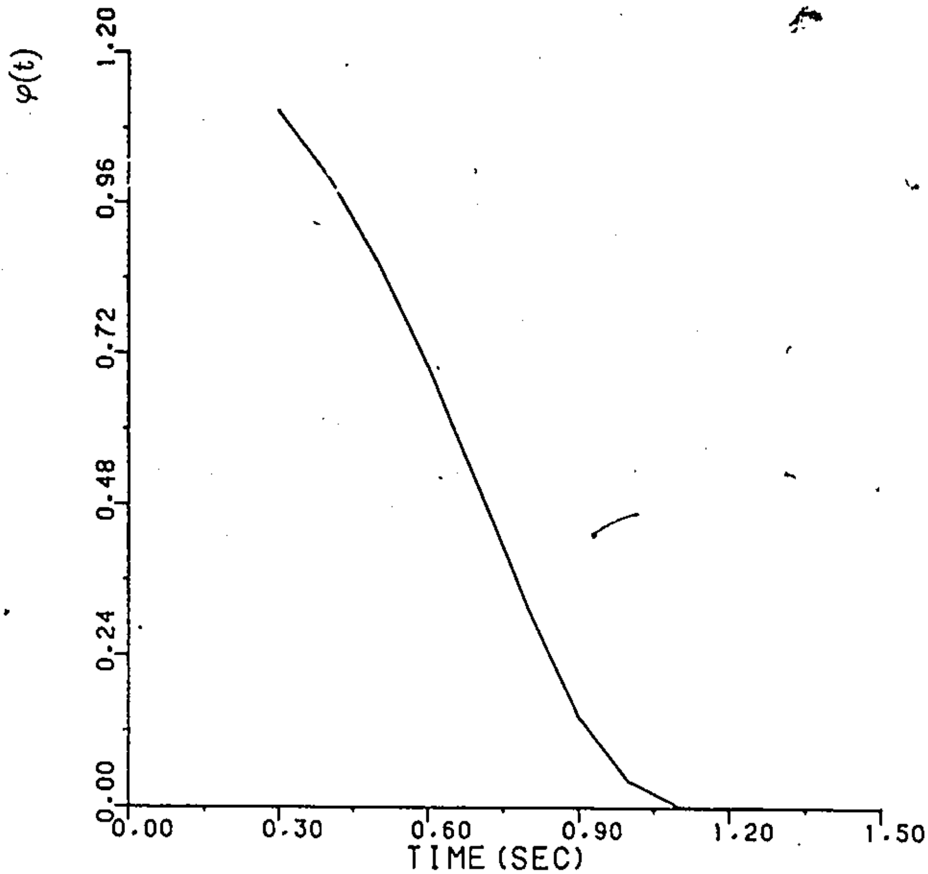


Fig.3.4. The cost Function $\varphi(t)$.

Iter.	cost
0	$.903\times10^{1}$
10	$.113\times10^{-1}$
20	$.901\times10^{-2}$
30	$.873\times10^{-2}$
60	$.511\times10^{-3}$
100	$.299\times10^{-5}$
150	$.372\times10^{-8}$

Table-3.1. Cost as a function of Number of Iterations.

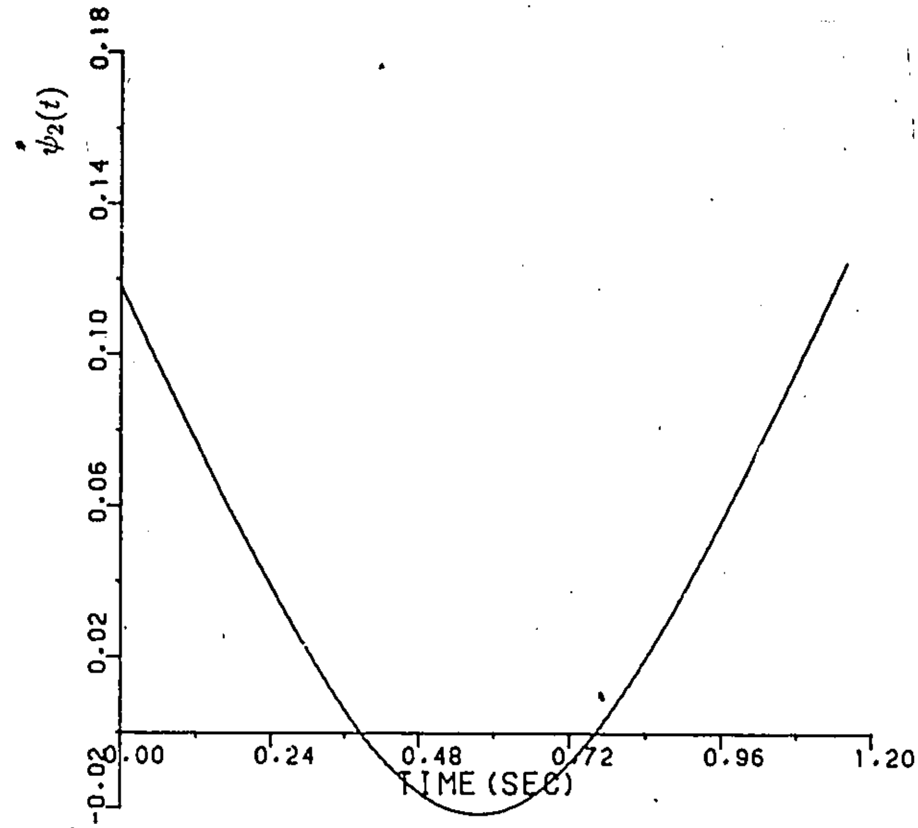


Fig.3.5.a. Optimal Costate ψ_2^*.

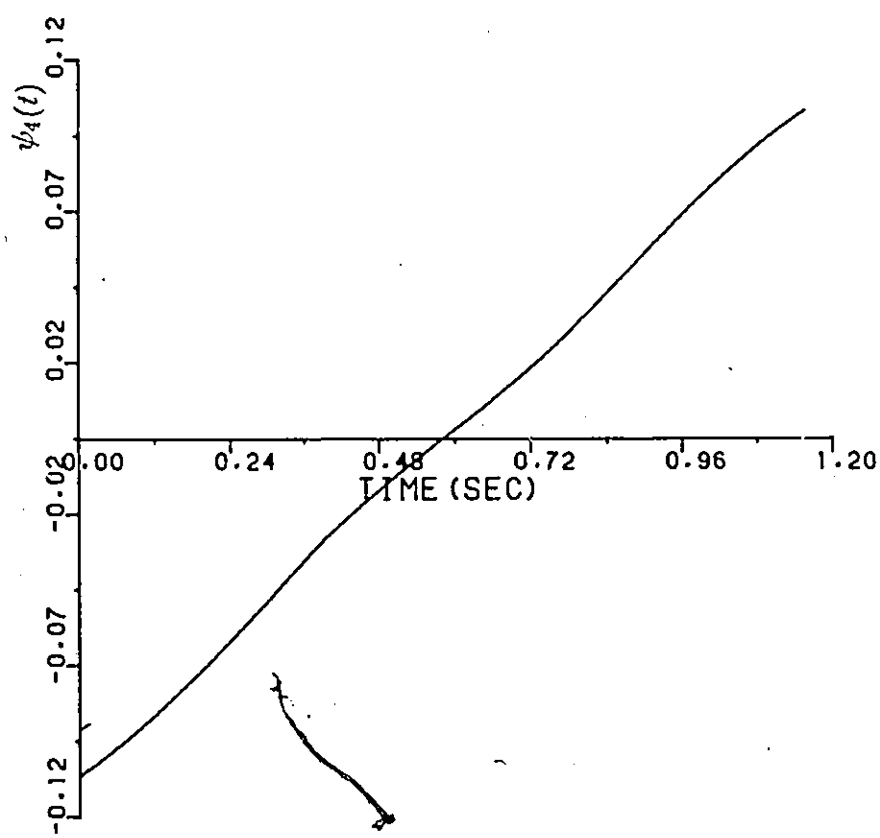


Fig.3.5.b. Optimal Costate ψ_4^*.

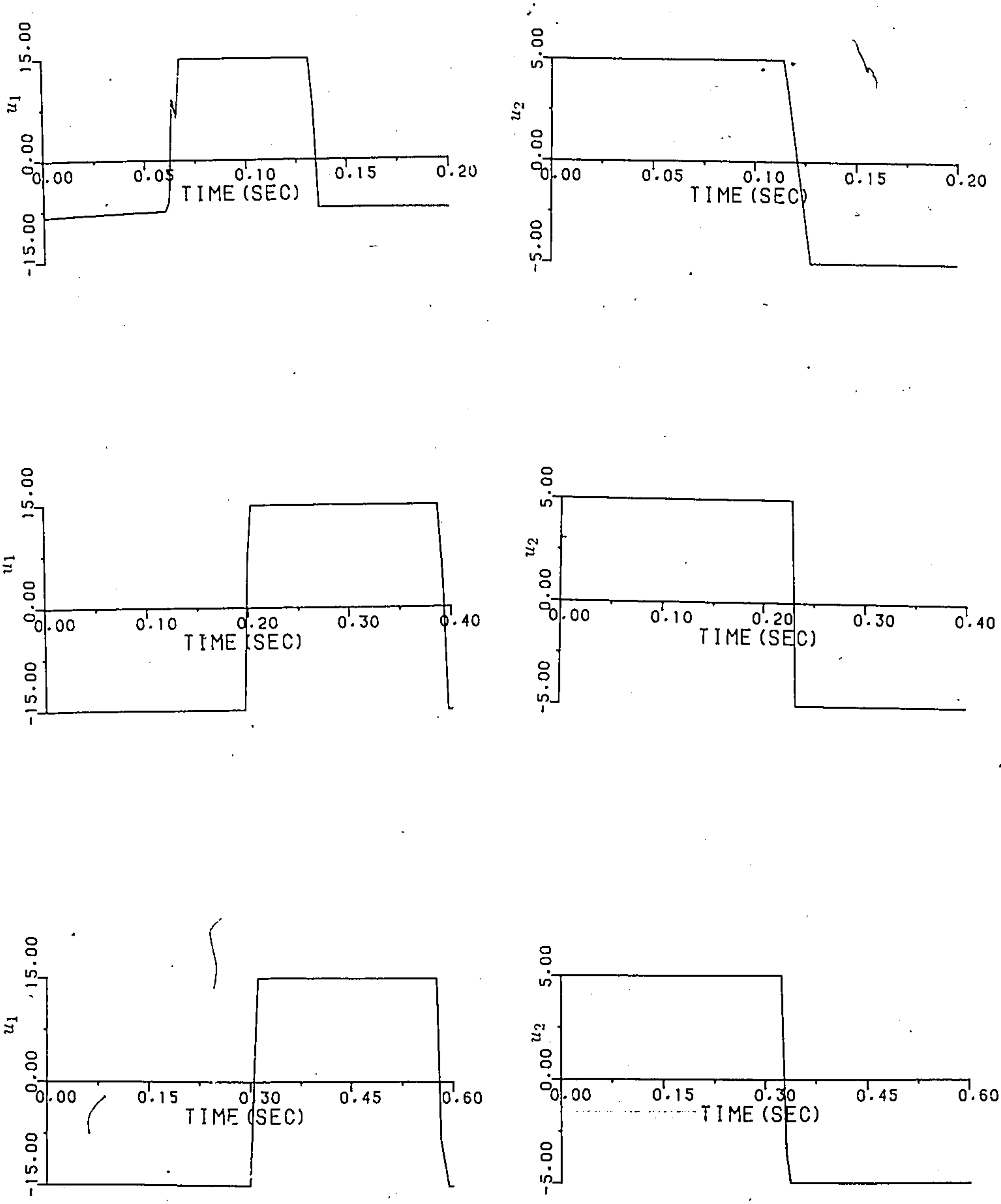
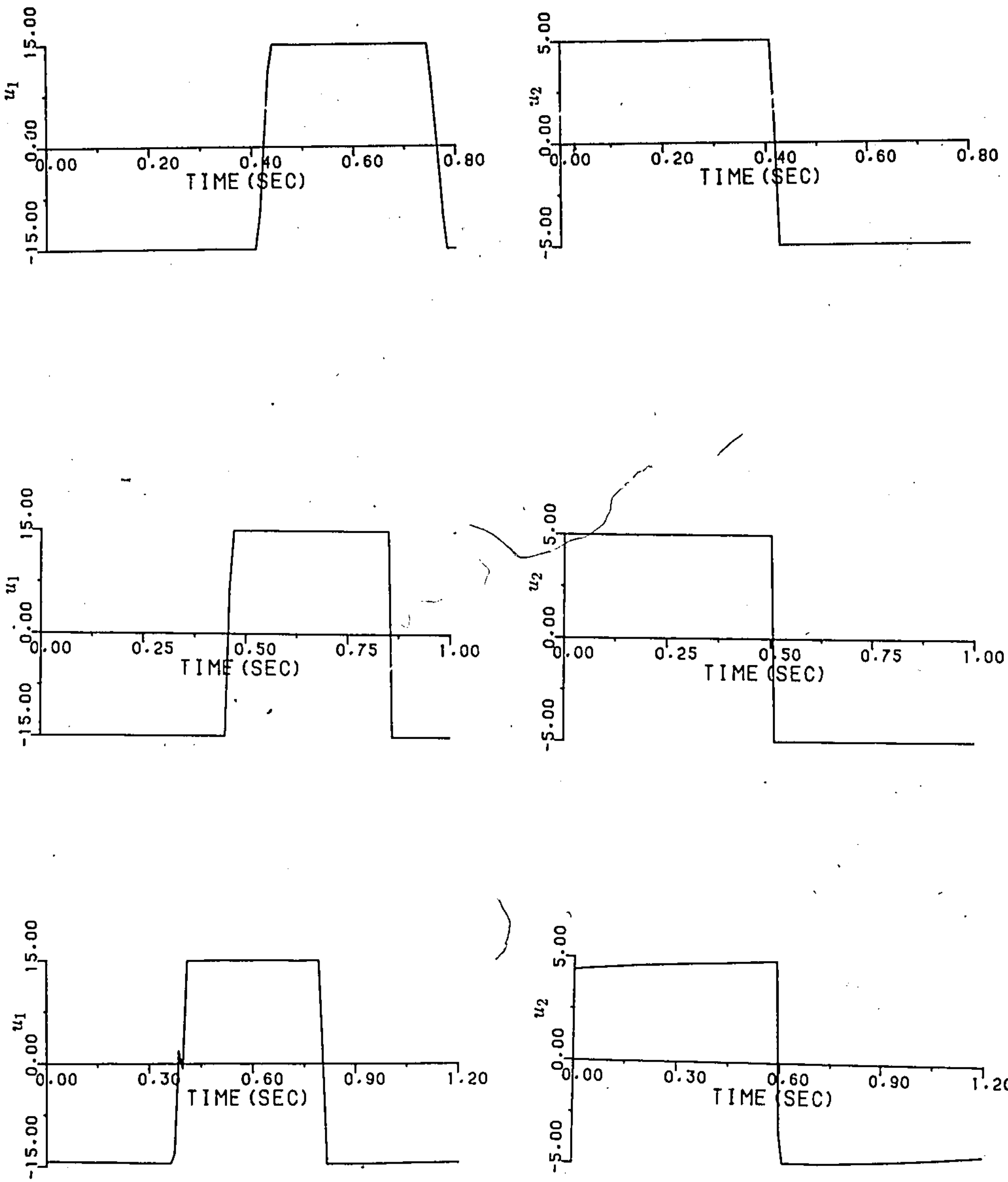


Fig.3.6. Optimal Controls u_1^* and u_2^* at Different Final Times.not optimal time


15.00
0.00
-15.00
u_1
0.00 0.20 0.40 0.60 0.80
TIME (SEC)
5.00
0.00
-5.00
u_2
0.00 0.20 0.40 0.60 0.80
TIME (SEC)
15.00
0.00
-15.00
u_1
0.00 0.25 0.50 0.75 1.00
TIME (SEC)
5.00
0.00
-5.00
u_2
0.00 0.25 0.50 0.75 1.00
TIME (SEC)
15.00
0.00
-15.00
u_1
0.00 0.30 0.60 0.90 1.20
TIME (SEC)
5.00
0.00
-5.00
u_2
0.00 0.30 0.60 0.90 1.20
TIME (SEC)

3.3 HORIZONTAL ARTICULATED ARM ROBOT WITH TWO LINKS

This robot can be regarded as the simplest nontrivial manipulator. It has two degree of freedom and its geometry is shown in Fig.3.7. The first degree of freedom is the azimuth rotation θ_1 of the inner link which is driven by a limited torque T_1. The second degree of freedom is the angular rotation θ_2 of the outer link which is driven by a limited torque T_2. The remaining degrees of freedom used for positioning the endeffector are neglected.

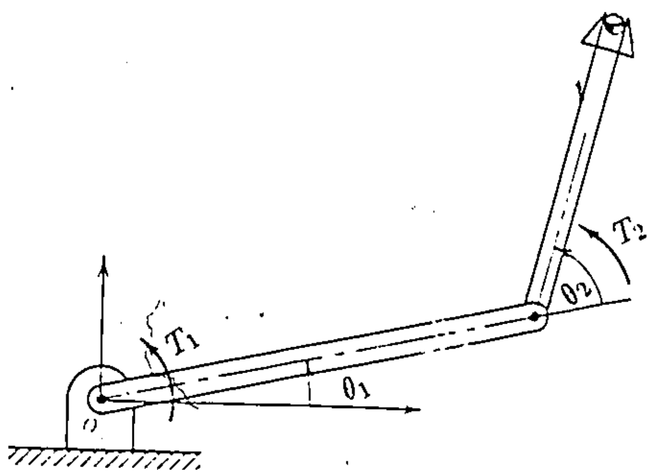


Fig.3.7. Sketch of the Articulated Arm Robot with Two Links.

Neglecting the friction, the articulated arm robot is governed the following set of equations [13],

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= [M_4\{u_1 - u_2 + M_3(x_2 + x_4)^2 \sin(x_3)\} - M_3 \\
&\quad .\{u_2 - M_3 x_2^2 \sin(x_3)\}\cos(x_3)]/[M_4 M_2 - M_3^2 \cos^2(x_3)] \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= [(M_2 + M_3\cos(x_3))\{u_2 - M_3 x_2^2 \sin(x_3)\} - (M_4 + M_3\cos(x_3)) \\
&\quad .\{u_1 - u_2 + M_3(x_2 + x_4)^2 \sin(x_3)\}]/[M_4 M_2 - M_3^2 \cos^2(x_3)]
\end{aligned}
\tag{3.5}
$$

with the state vector $x(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]' \in R^4$ and the control vector $u(t) = [u_1(t), u_2(t)]' \in R^2$.

where

x_1 =angular rotation of the inner link.

x_2 =rate of change of x_1.

x_3 =angular rotation of the outer link.

x_4 =rate of change of x_3.

u_1 $=T_1$

u_2 $=T_2$

M_f =mass moment of inertia of the first link w.r.t. the first axis.

M_s =mass moment of inertia of the second link w.r.t. the second axis.

M_h =mass moment of inertia of the hand and load w.r.t. the position of the hand.

M_1 $= M_s + L_2^2 m_h$

M_2 $= M_f + L_1^2(m_s + m_h)$

M_3 $= L_1(d_s m_s + L_2 m_h)$

M_4 $= M_1 + M_h$

L_1 =length of the first link.

L_2 =length of the second link.

m_s = mass of the second link.

m_h =mass of the hand and the load.

d_s =distance between the center of gravity of the second link and its driving axis.

The control constraints are

$$|u_1(t)| \le T_{1,\max}$$
$$|u_2(t)| \le T_{2,\max}$$

The structure of the time-optimal solution will be discussed for the special case where the robot's arm is stretched both in the initial and in the final position, i.e., for the boundary conditions

$$x_1(0) = x_2(0) = x_3(0) = x_4(0) = 0$$

$$x_1(\tau) = \theta_1 > 0$$

$$x_2(\tau) = 0$$

$$x_3(\tau) = \theta_2 = 0, \text{ or } \pm 2\pi, \ldots$$

and

$$x_4(\tau) = 0$$

Obviously, for θ_1¡ 0 the solution is symmetric.

For numerical simulation we use the following parameters as given in the above reference,

$L_1 = .4\text{m}$, $L_2 = .25\text{m}$, $d_s = .125\text{m}$, $m_s = 15Kg$, $m_h = 6Kg$, $M_f = 1.6\text{m}^2Kg$, $M_s = .43\text{m}^2Kg$, $M_h = .01\text{m}^2Kg$, $T_{1,\max} = 25N\text{m}$, $T_{2,\max} = 9N\text{m}$.

a. $\theta_1 = 2.0$ and $\theta_2 = -2\pi$

The function $\varphi(\tau)$, the optimal trajectories and controls $u_1^*(t)$ and $u_2^*(t)$ are computed using the algorithm in chapter II as shown in Fig.3.8-3.10. The time optimal control is bang-bang for both of the control variables. Mathematically, the analysis proceeds along the lines of example 1. The Hamiltonian is again affine in the controls u_1 and u_2 are nonzero over some interval. The optimal time is τ^*=.964 (sec) which is 25 % better than that computed in [13]. In Fig.3.11, controls u_1^*(t) and u_2^*(t) are shown for different final times. The function $\varphi(\tau)$ is monotone decreasing and φ(t)=0 for all $t \geq \tau^*$.

b. $\theta_1 = .4$ and $\theta_2 = 0$

The function $\varphi(\tau)$ is monotone decreasing as shown in Fig.3.14. The optimal controls and trajectories are computed as shown in Fig.3.12-3.13. The time optimal control is bang-bang for both of the control variables. The optimal time is $\tau^* = .759$ (sec) which is 30% better than that computed in [13].In table (3.2), a summary of the cost at the optimal time is given as a function of the number of iterations.

c. θ_1=1.2 and $\theta_2 = -2\pi$

Using the algorithm in chapter II, the optimal controls and trajectories are computed as shown in Fig.3.15-3.16. The time optimal control is purely bang-bang for both of the control variables. The optimal time is τ^*=.925(sec) which is better than that computed in [13]. The function $\varphi(\tau)$ is monotone decreasing and $\varphi(t)$=0 for all $t \geq \tau^*$.

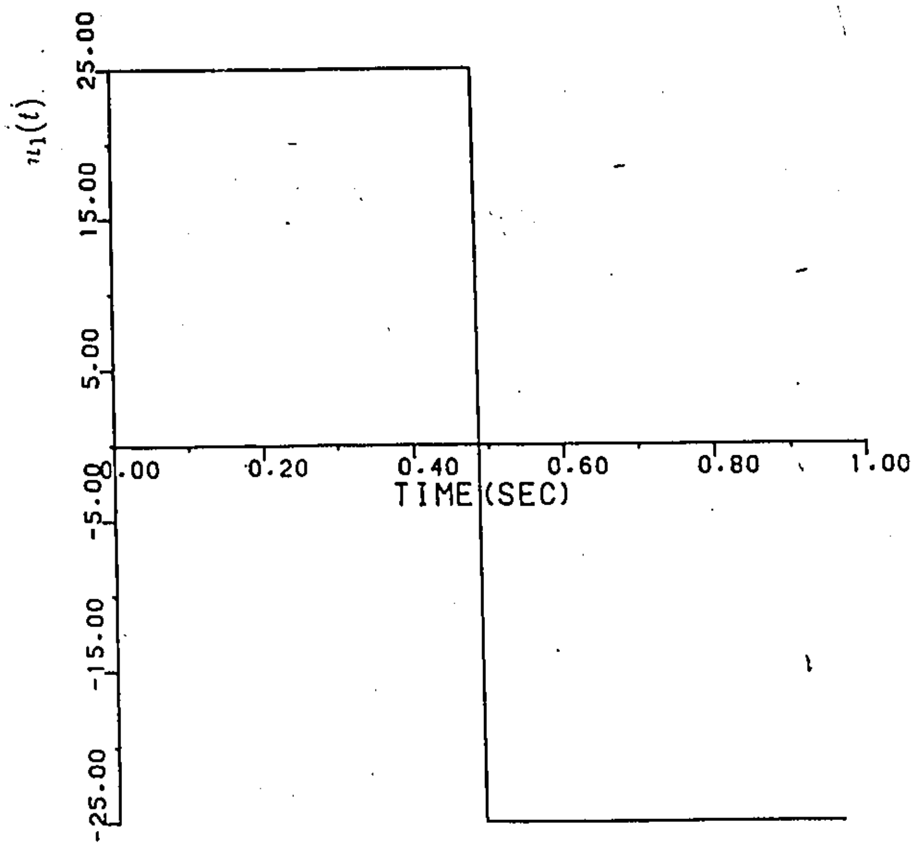


Fig.3.8.a. Optimal Control u_1^*.

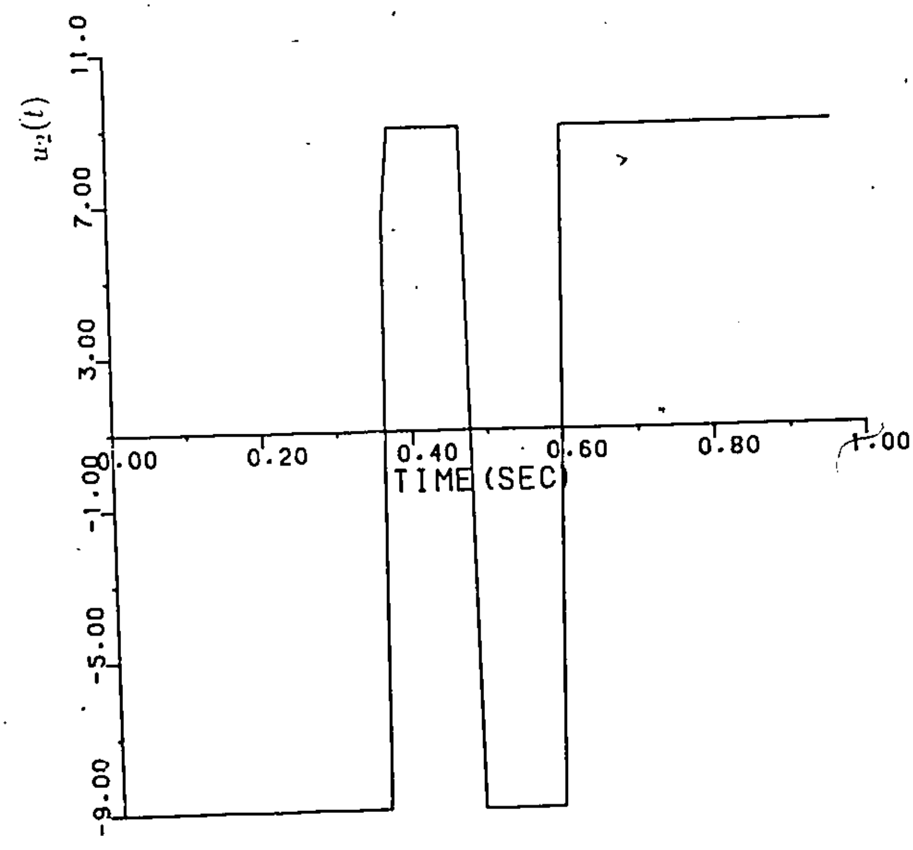


Fig.3.8.b. Optimal Control u_2^*.

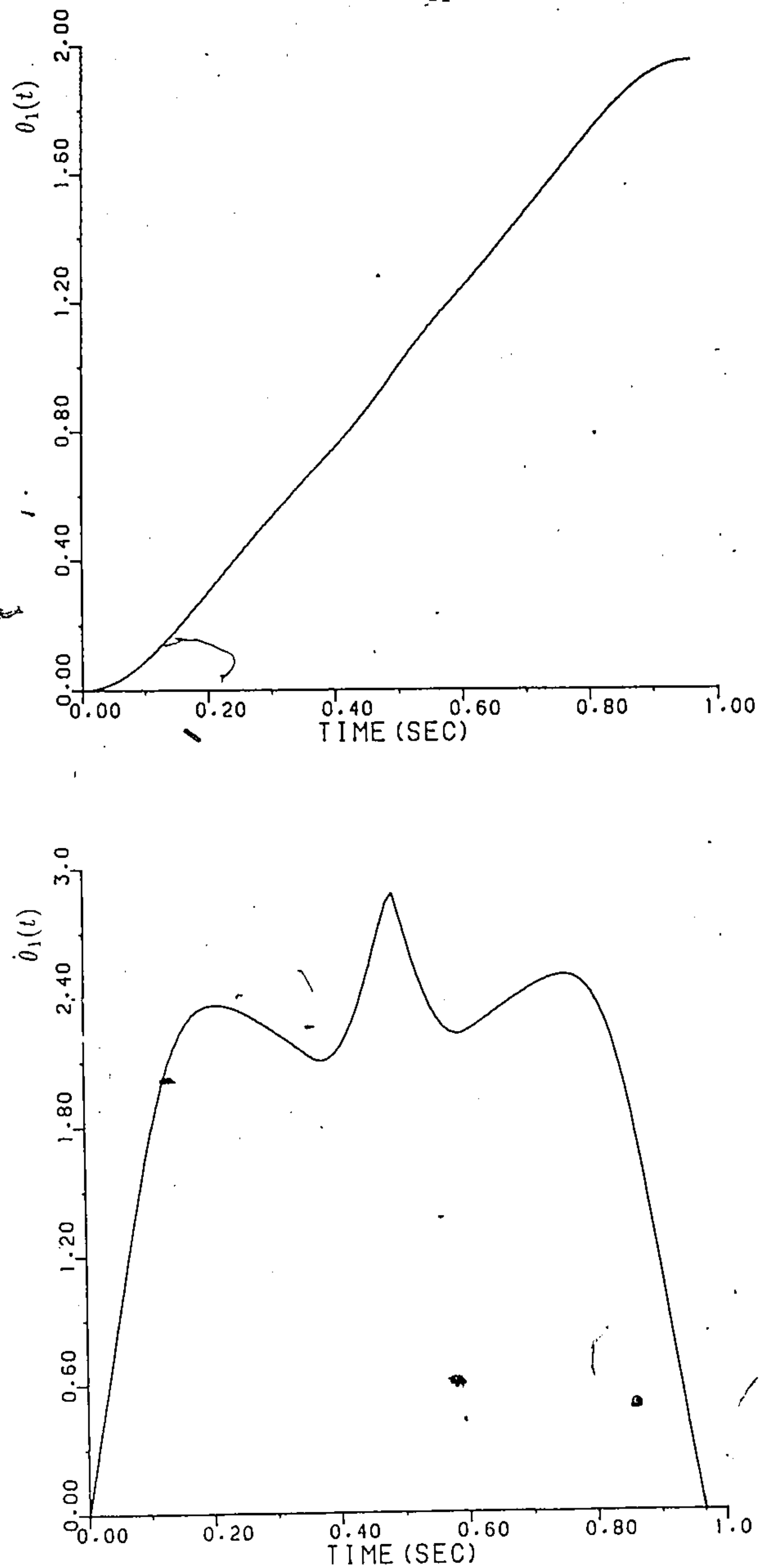
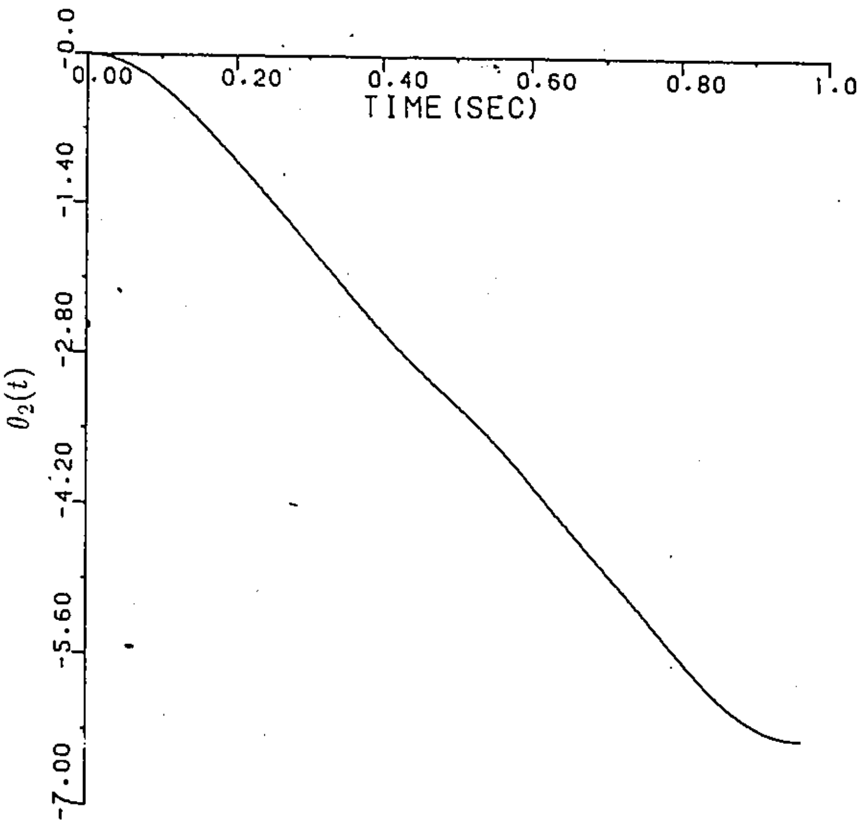


Fig.3.9.a. Optimal States θ_1 and $\dot{\theta}_1$



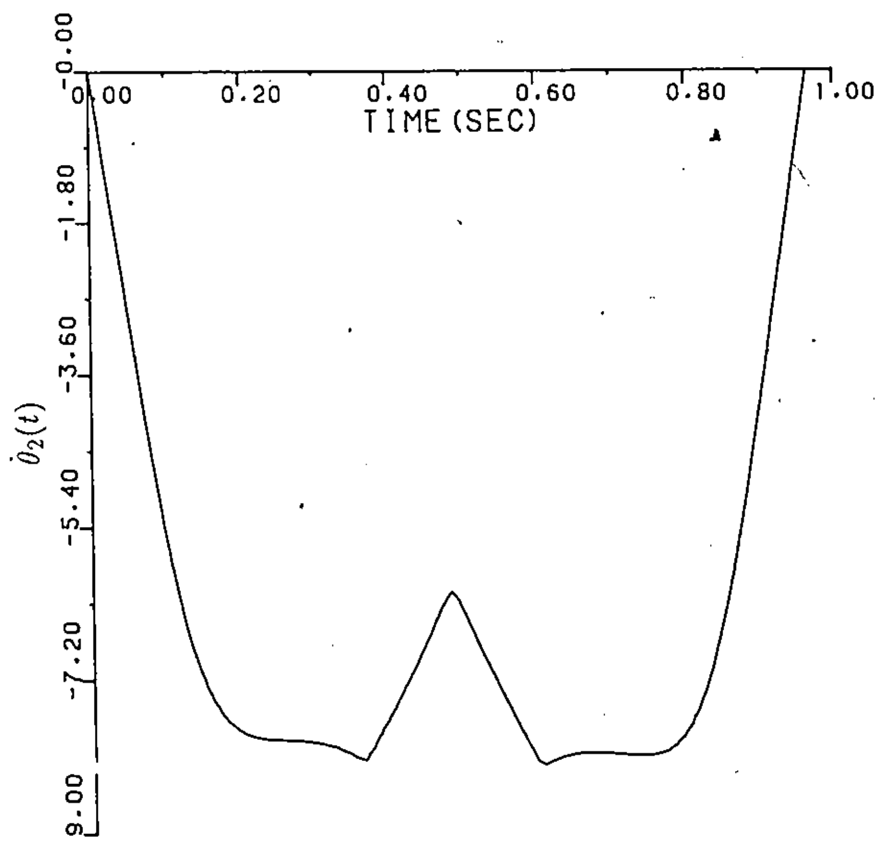


Fig.3.9.b. Optimal States θ_2 and $\dot{\theta}_2$

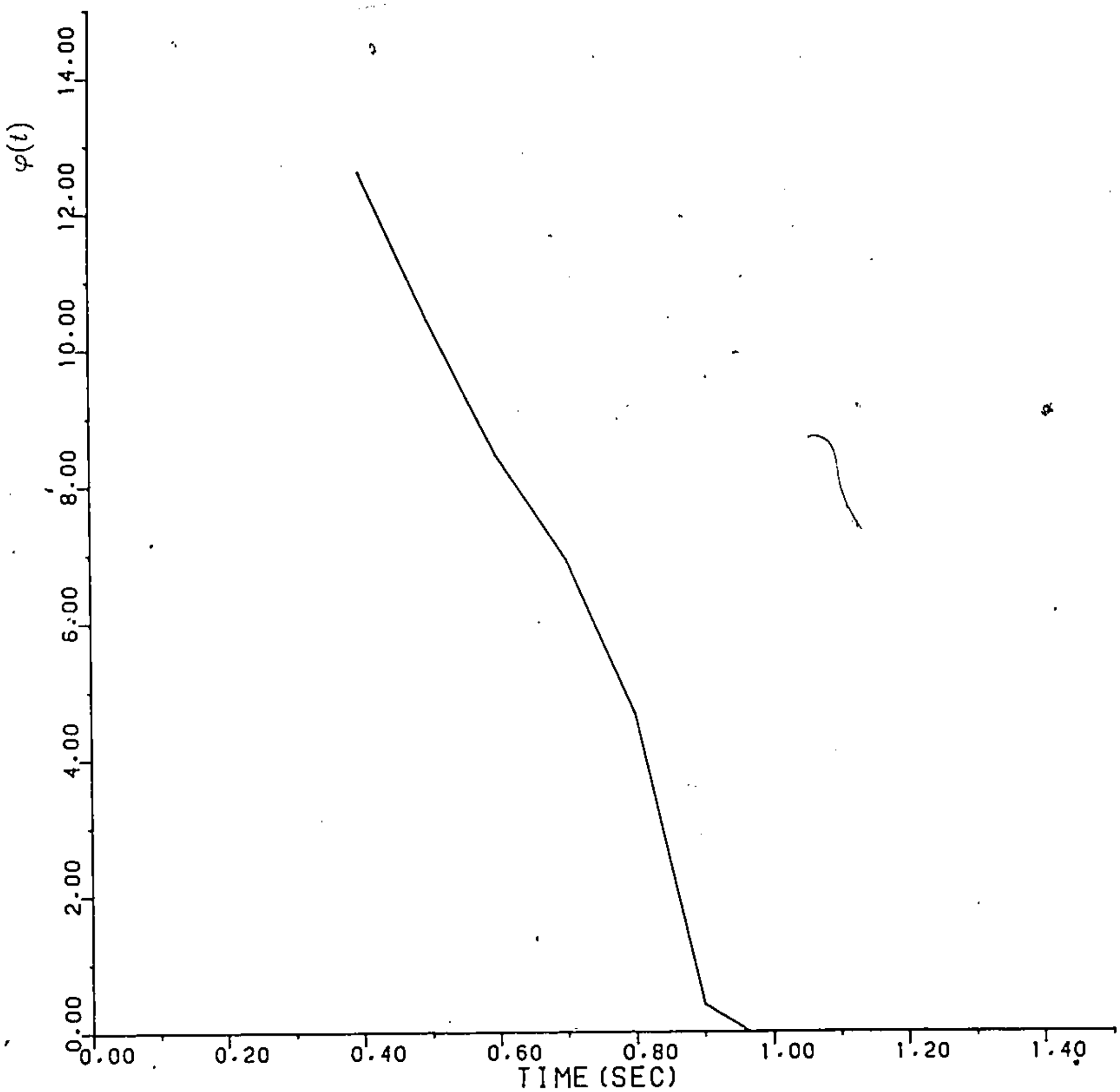


Fig.3.10. The Cost Function φ

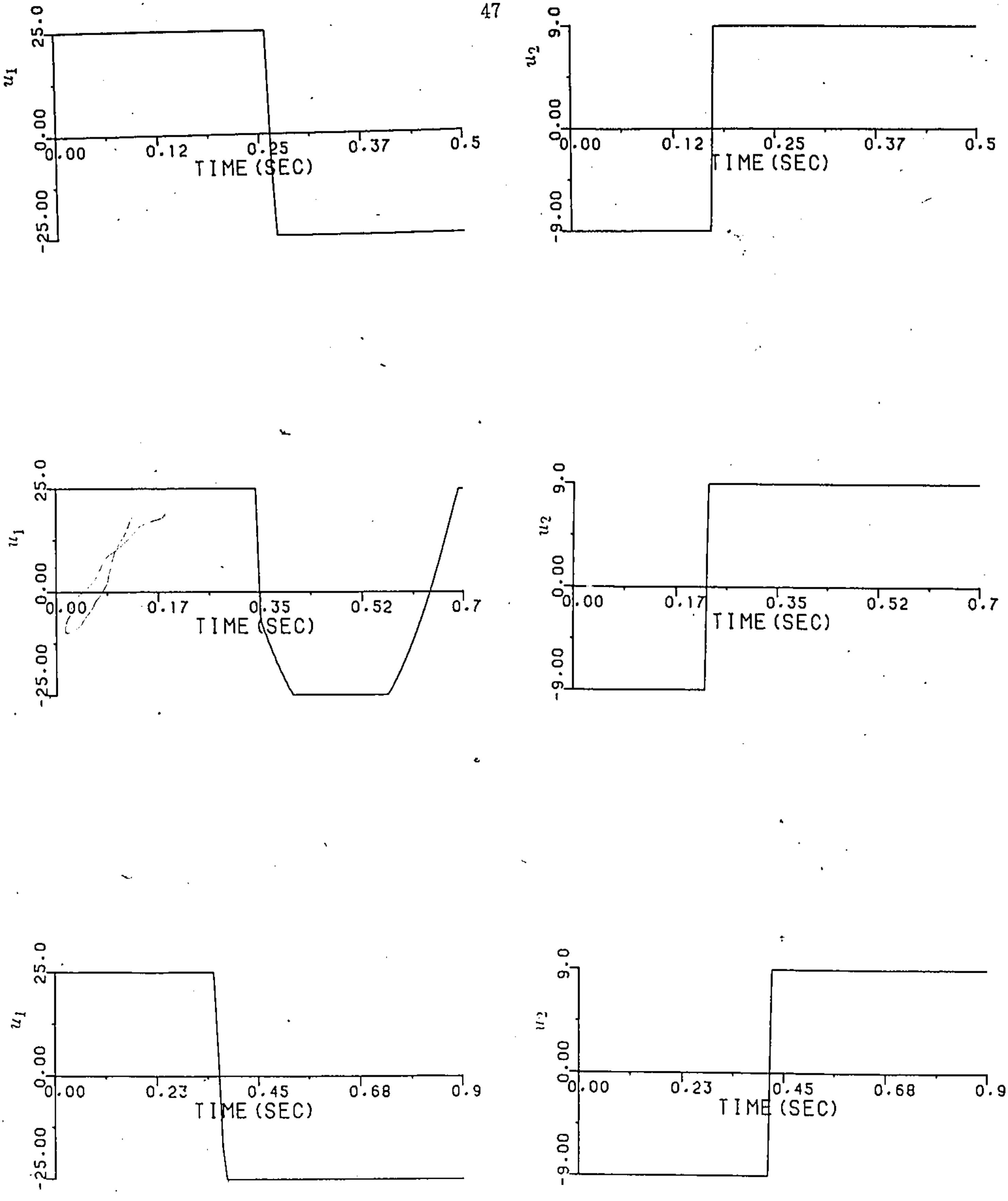
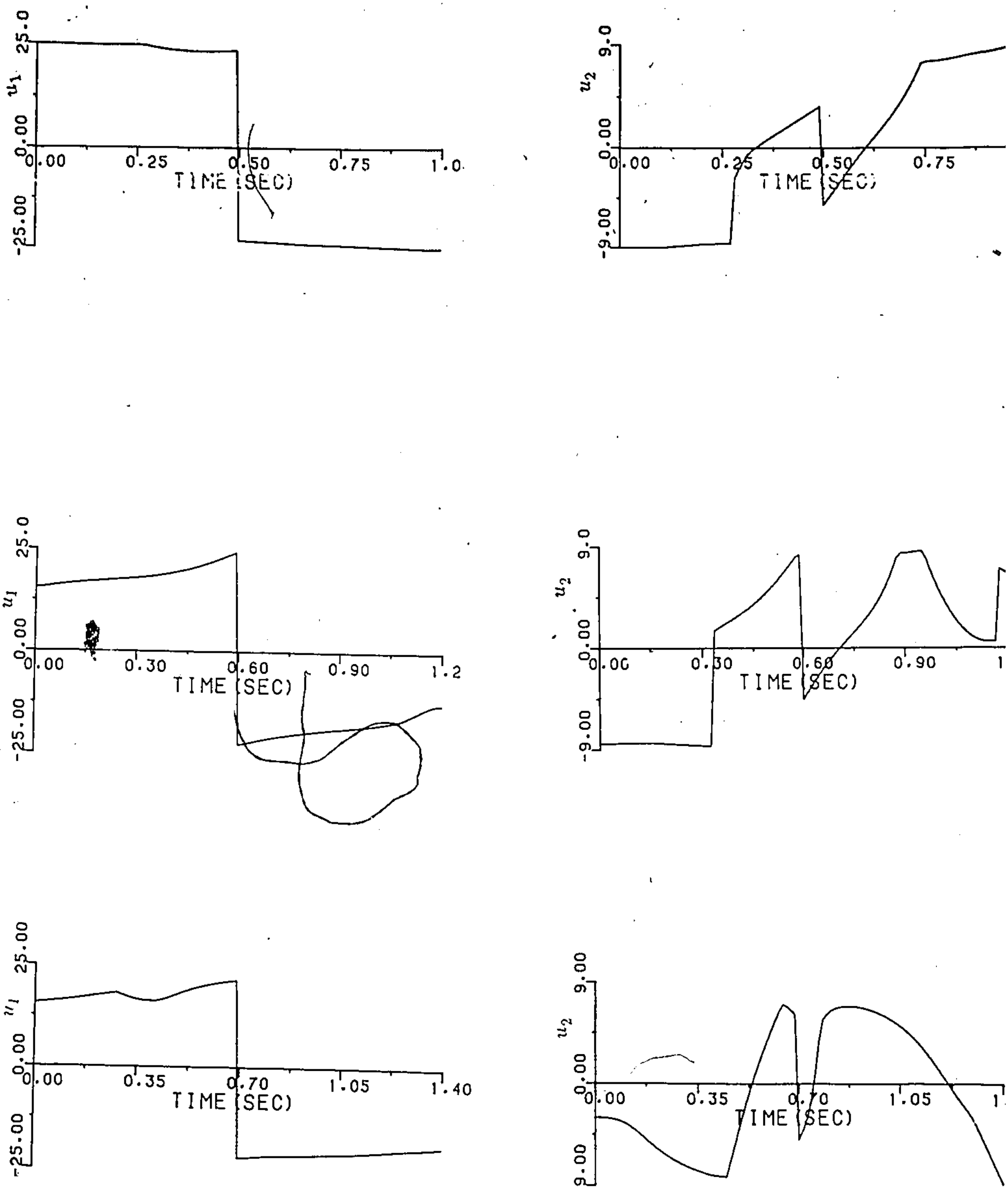


Fig.3.11. Optimal Control u_1^* and u_2^* at Different Final Times not optimal time



u_1
25.0
0.00
-25.00
0.00
0.25
0.50
0.75
1.0
TIME (SEC)
u_2
9.0
0.00
-9.00
0.00
0.25
0.50
0.75
TIME (SEC)
u_1
25.0
0.00
-25.00
0.00
0.30
0.60
0.90
1.2
TIME (SEC)
u_2
9.0
0.00
-9.00
0.00
0.30
0.60
0.90
1
TIME (SEC)
u_1
25.00
0.00
-25.00
0.00
0.35
0.70
1.05
1.40
TIME (SEC)
u_2
9.00
0.00
-9.00
0.00
0.35
0.70
1.05
TIME (SEC)

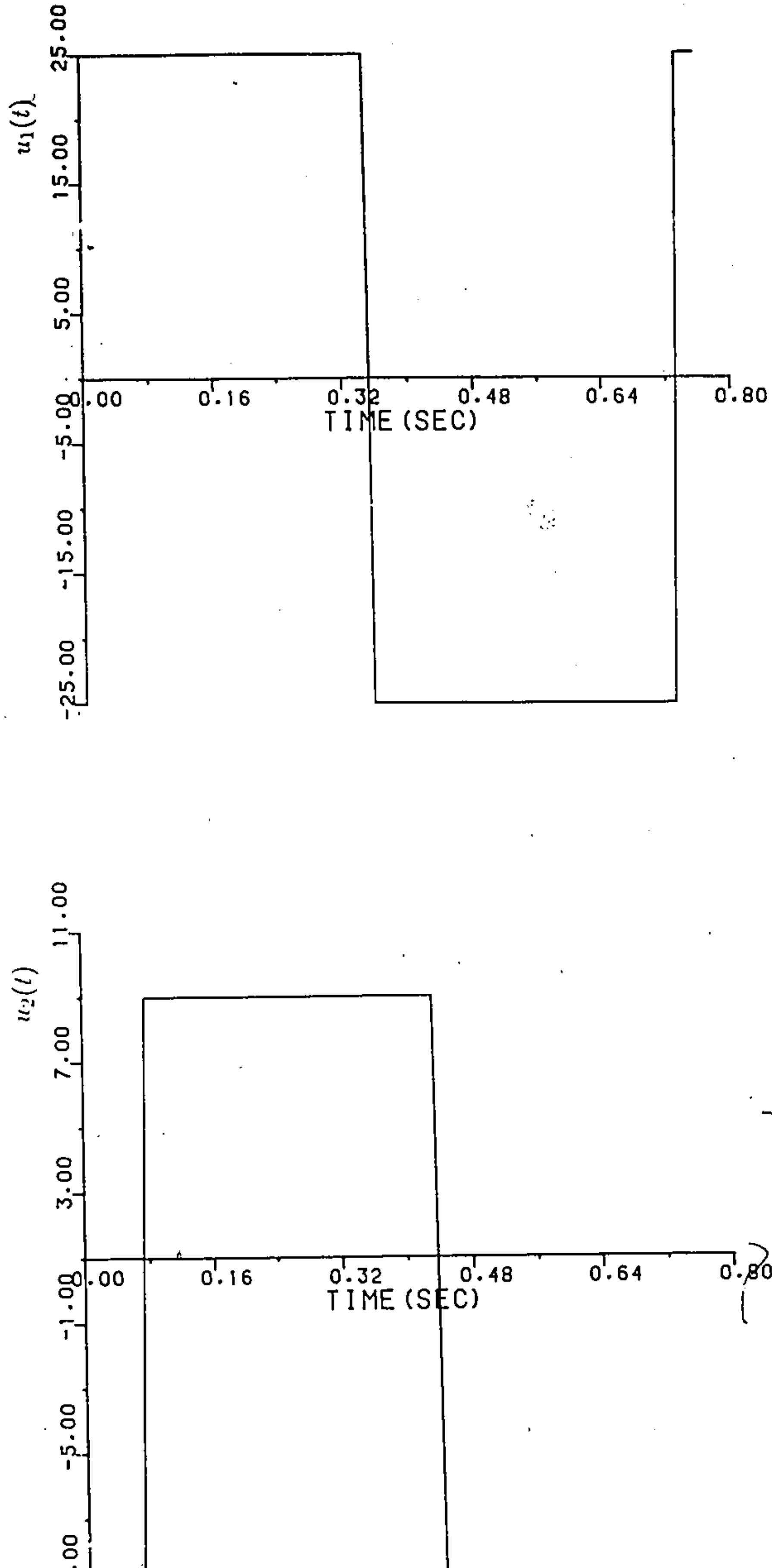


Fig.3.12. Optimal Controls u_1^* and u_2^*.

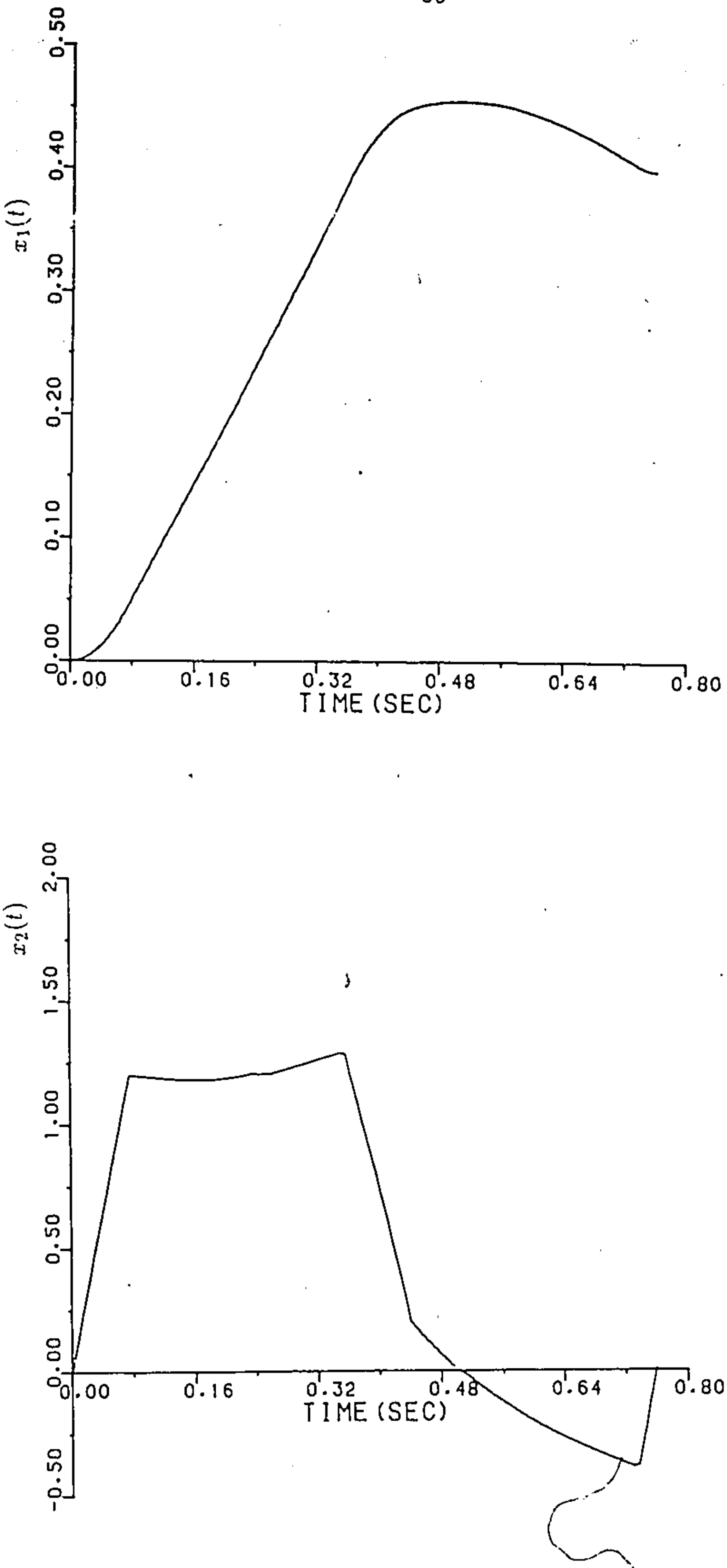
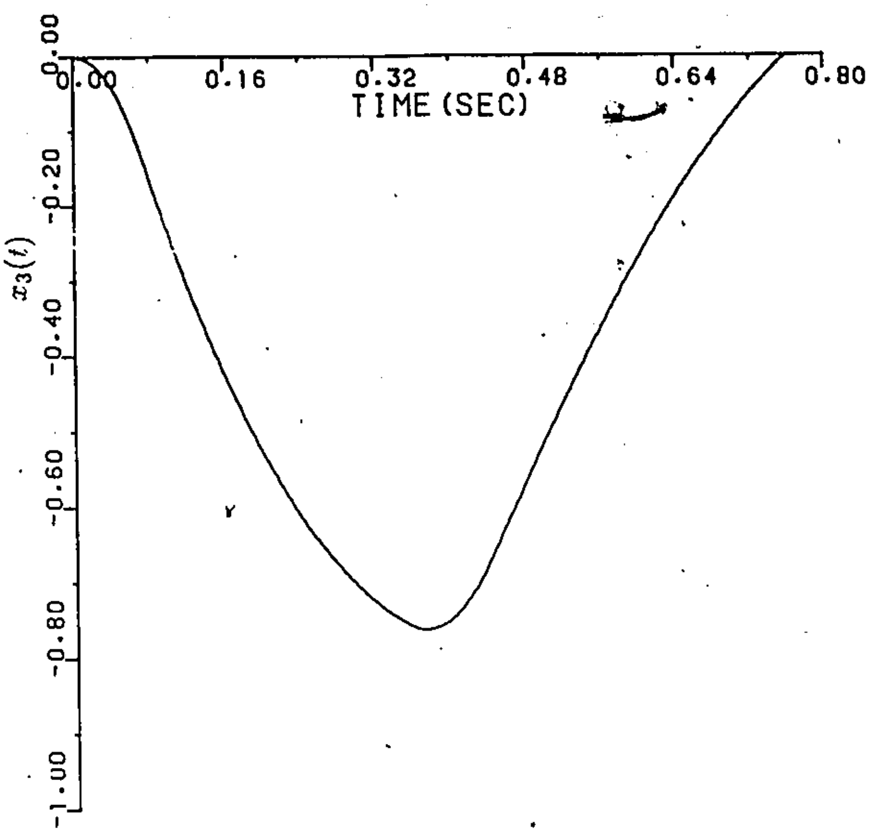


Fig.3.13.a. Optimal States x_1^* and x_2^*.



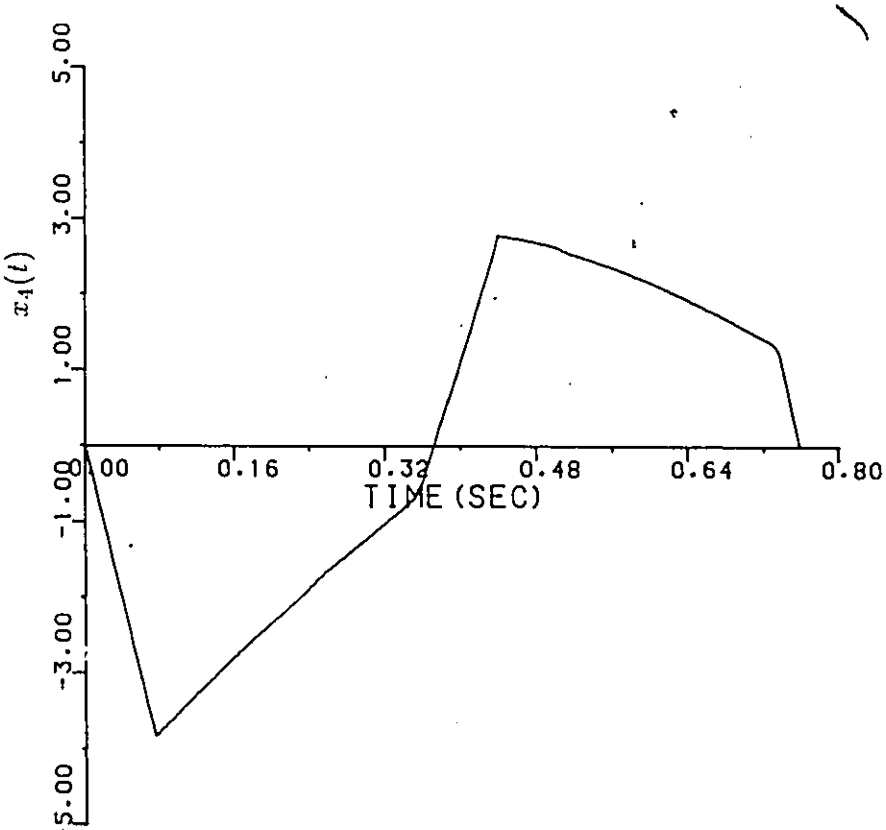


Fig.3.13.b. Optimal States x_3^* and x_4^*.

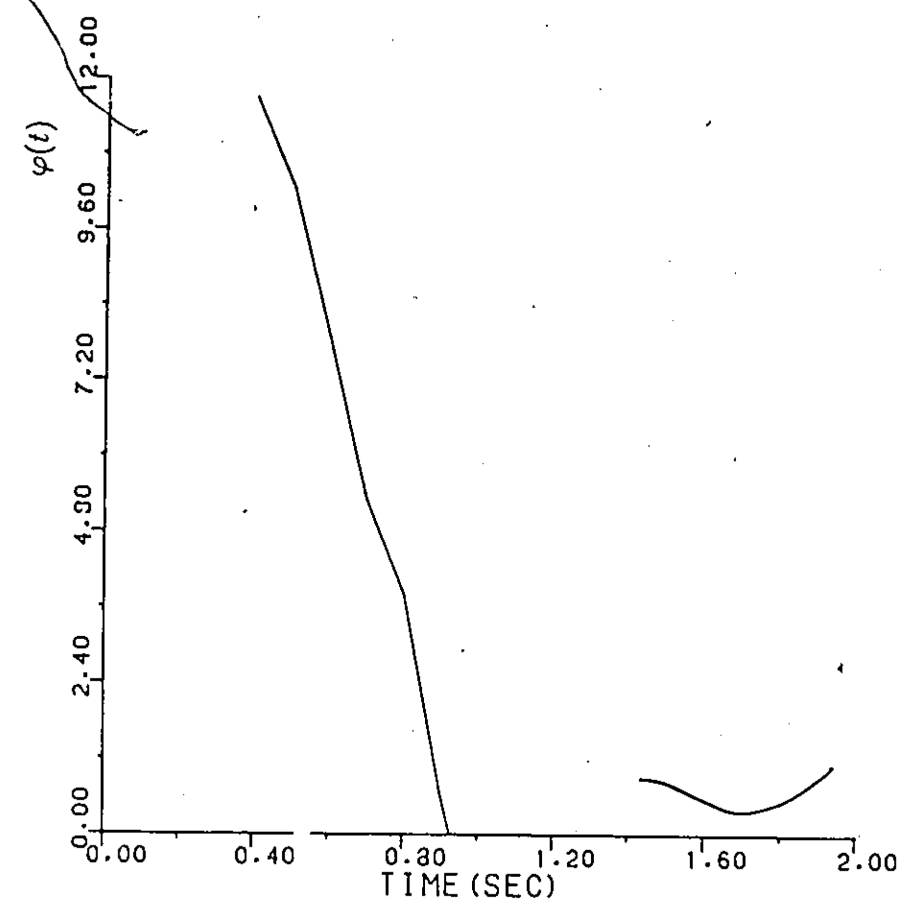


Fig.3.14. The Cost Function φ

Iter.	cost
0	$.228\times10^{1}$
10	$.701\times10^{-2}$
20	$.522\times10^{-2}$
30	$.133\times10^{-2}$
60	$.123\times10^{-3}$
80	$.268\times10^{-4}$
100	$.517\times10^{-5}$
150	$.273\times10^{-7}$
190	$.654\times10^{-10}$

Table 3.2.

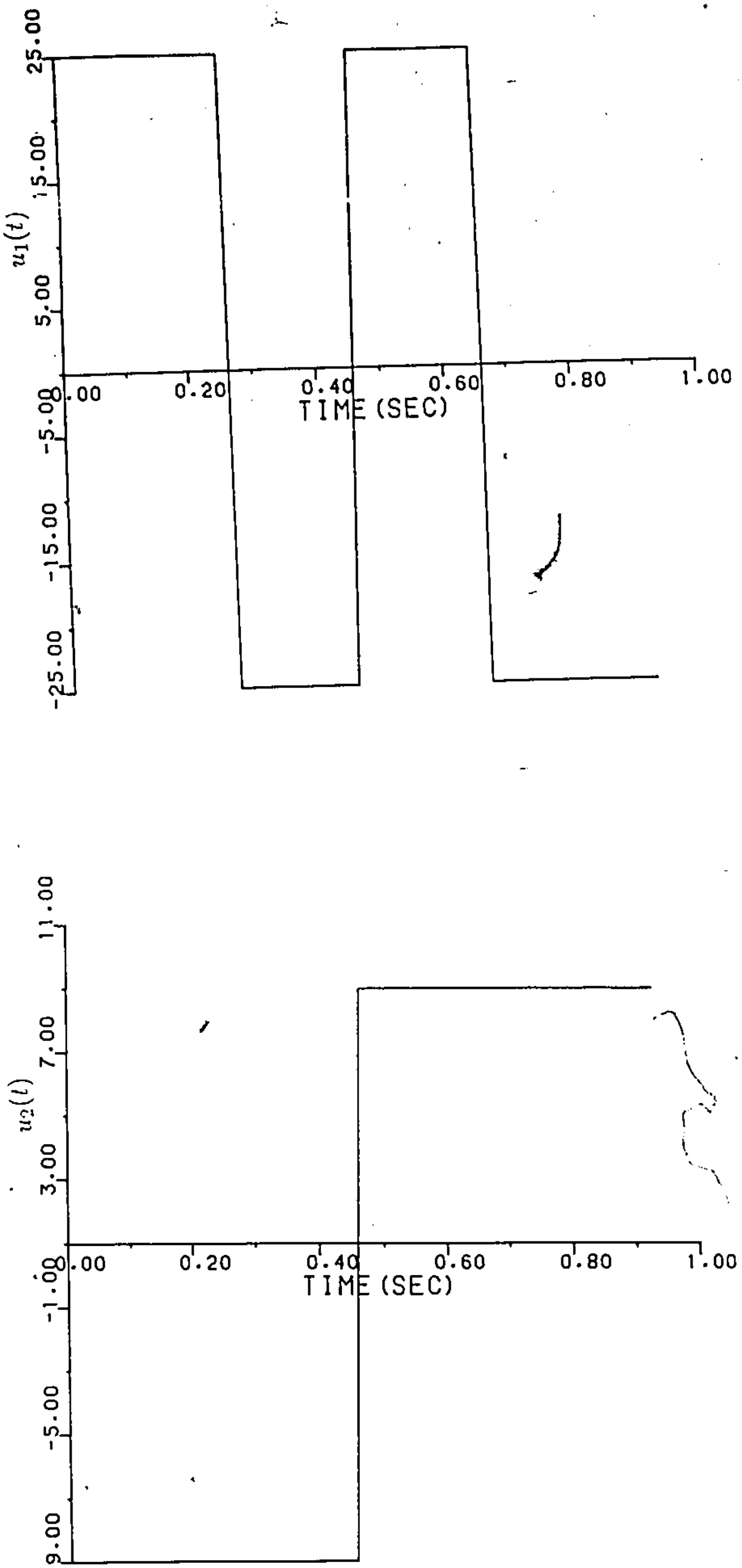
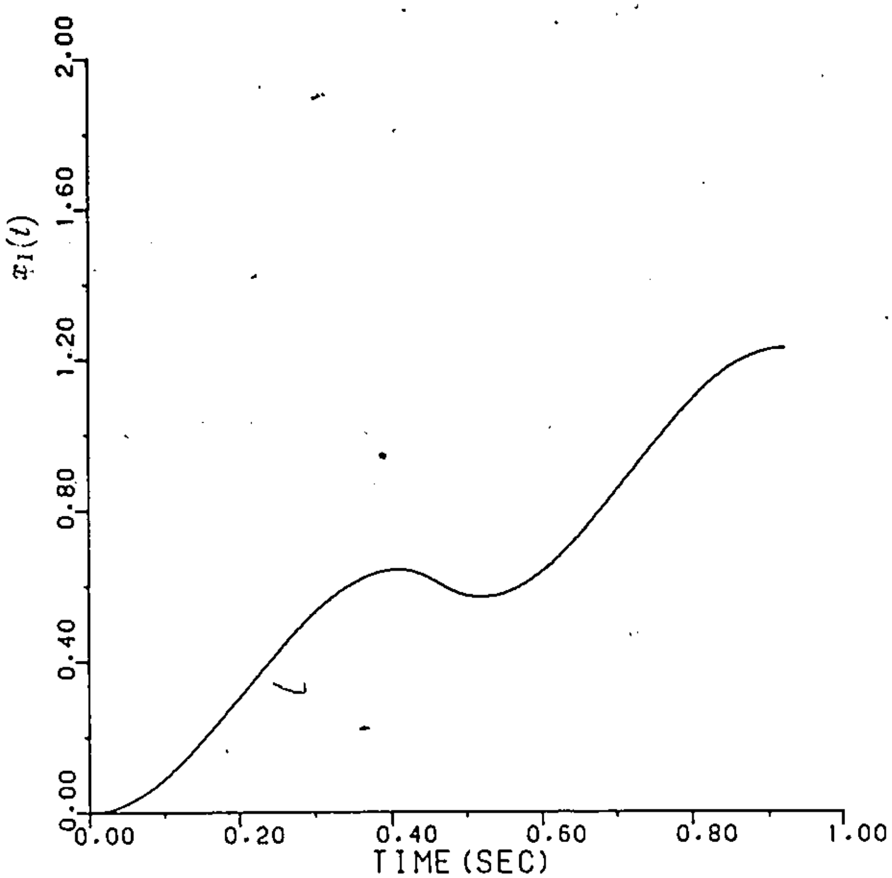


Fig.3.15. Optimal controls u_1^* and u_2^*.



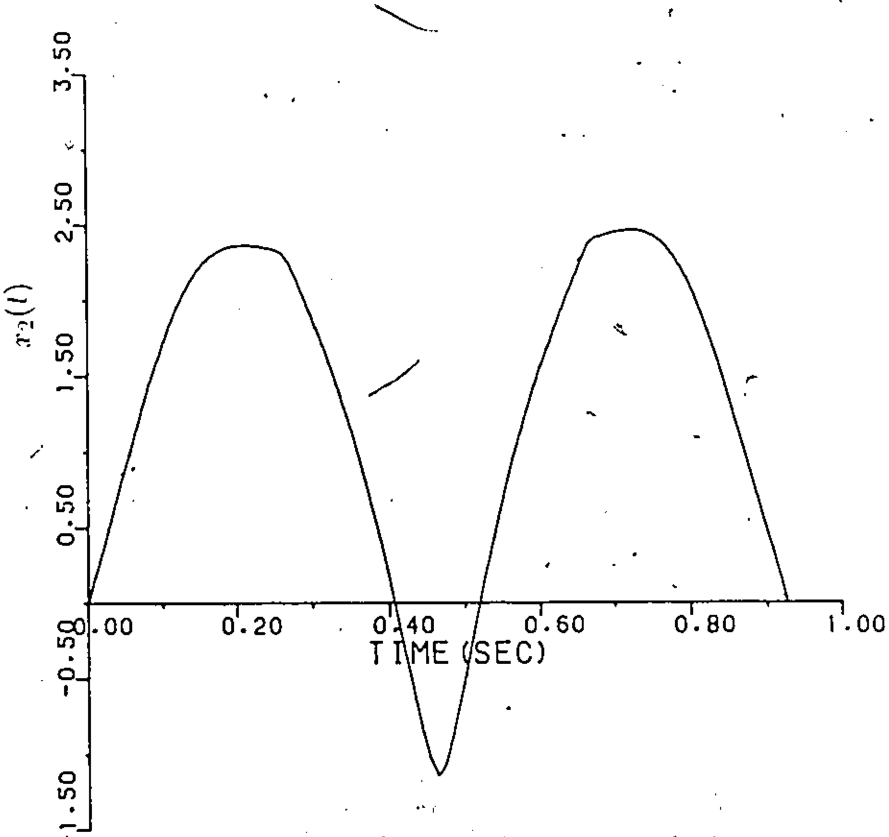


Fig.3.16.a. Optimal States x_1^* and x_2^*.

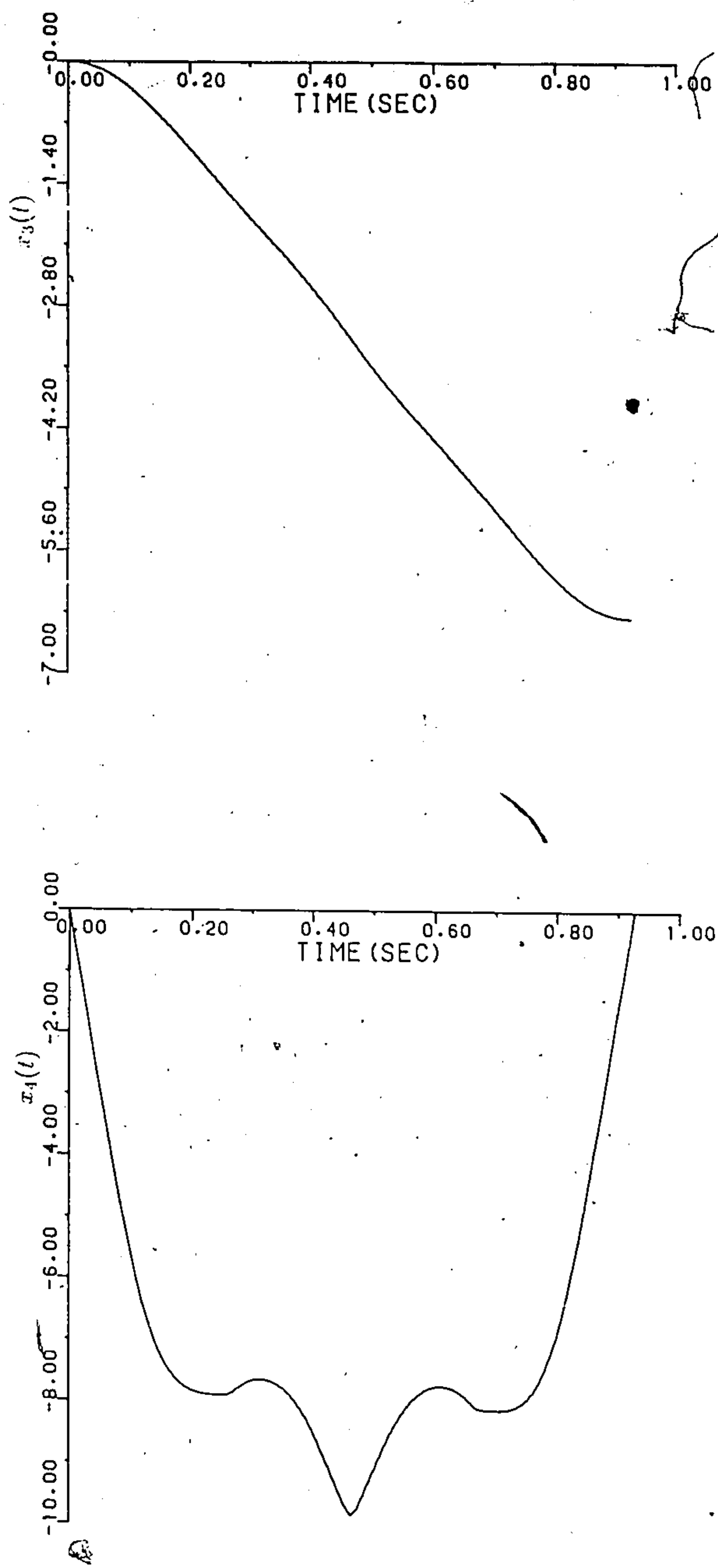


Fig. 3.16.1 Optimal States x_3^* and x_4^*

3.3 ECOLOGICAL SYSTEM

In the eastern Canada and the northeastern United States, the spruce and fir forests have been periodically subject to ravages by a caterpillar called the spruce budworm. For a number of years, a given path of forest is seen to grow with hardly any budworm in evidence. When the trees have reached a certain level of maturity there is, however, an explosive increase in the number of these insects and they begin to defoliate the trees. When a stand of mature trees have been sufficiently denuded over several consecutive years, they wither and die.

Budworm growth depends on the amount of the habitat available, which is measured in average branch surface area per acre of land and on the amount of spruce and firneedles on these branches. Average budworm density is assumed to have logistic growth in which the carrying capacity is proportional to the habitat size.

The ecological system is governed by the following set of equations [25].

$$\begin{aligned} \dot{x}_1 &= ax_1(1 - x_1/bx_3) - [cx_1^2/(dx_3^2 + x_1^2)] - u_1x_1 \\ \dot{x}_2 &= \alpha x_2(1 - x_2) - \beta x_1 x_2^2/x_3 \\ \dot{x}_3 &= \gamma x_3(1 - x_3/\delta x_2) \end{aligned} \tag{3.6}$$

with the state vector $x(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]' \in R^4$ and the control $u(t) = u_1(t) \in R$.

Where

x_1 =average budworm density.

x_2 =percentage of foliage on the trees with x_2 close to unity indicating a healthy forest.

x_3 =average branch surface area per acre of land.

u_1 =the control giving fractional removal rate.

For numerical simulation we choose the following parameters:

$a = 0.5$, $b = 300$, $c = 10.5$, $d = 100$,

$\alpha = 1.0$, $\beta = 0.001$, $\gamma = 1.0$, $\delta = 3 \times 10^4$.

For inititial state [10000, .20, 3000] and for terminal state [1360, .65, 9087] optimal trajetories, control, and $\varphi(\tau)$ are shown in Fig.3.17-3.19. In this case, the optimal cost $\varphi(\tau)$ increases after the optimal time.

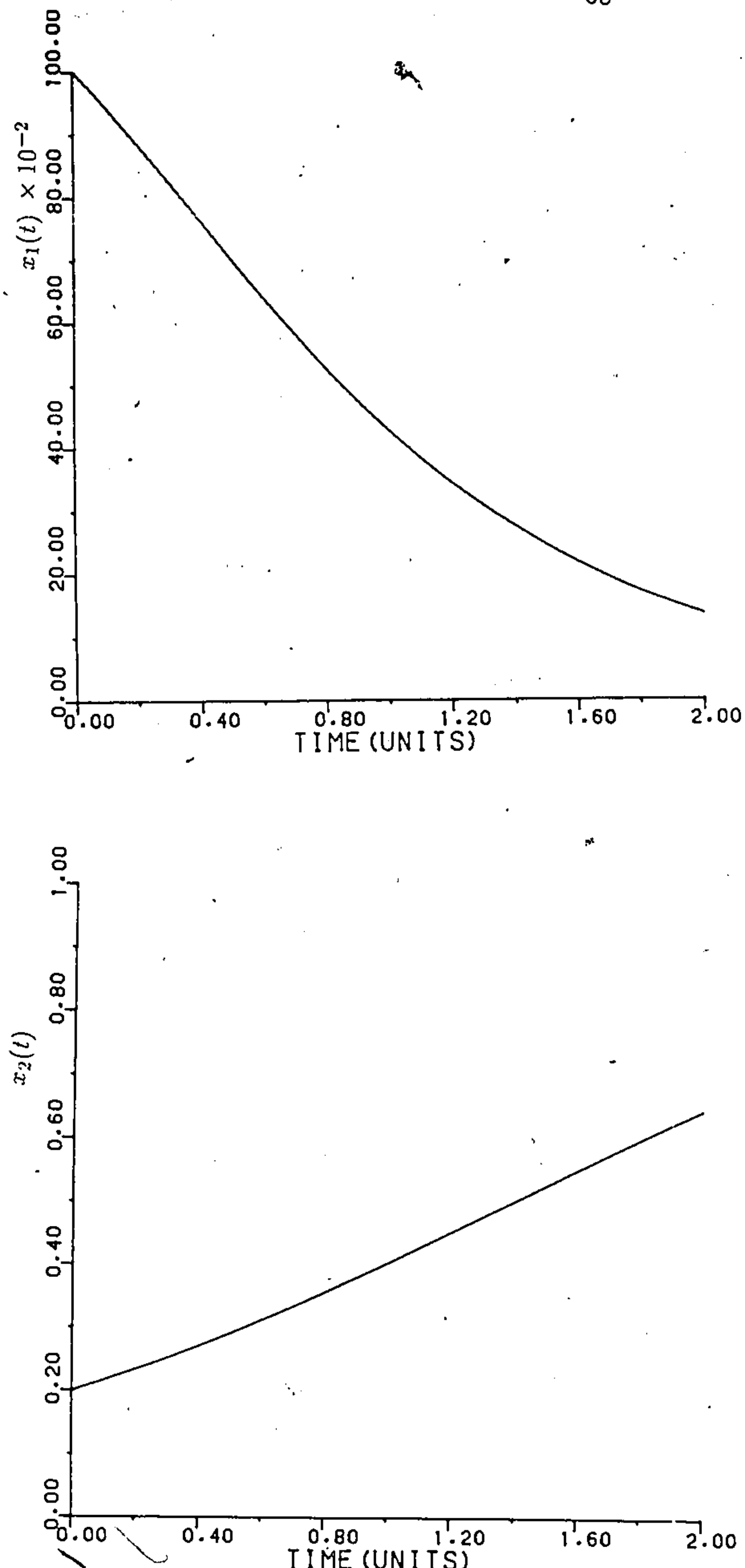


Fig.3.17.a. Optimal States x_1^* and x_2^*.

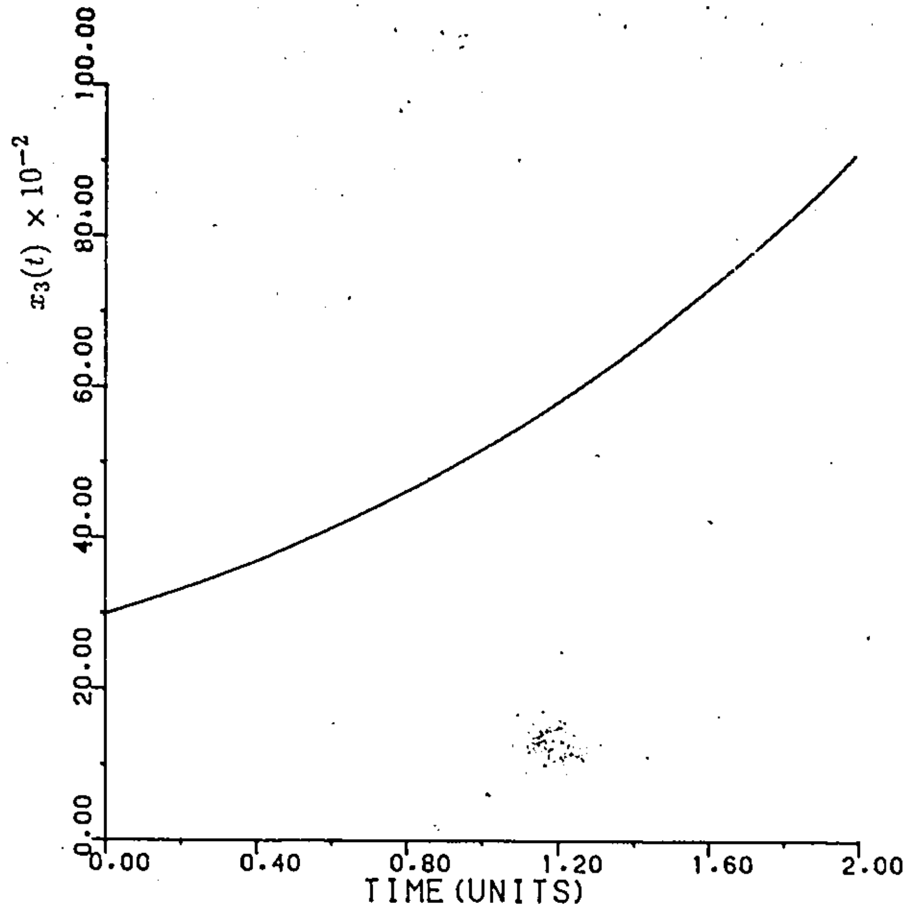


Fig.3.17.b. Optimal State x_3^*.

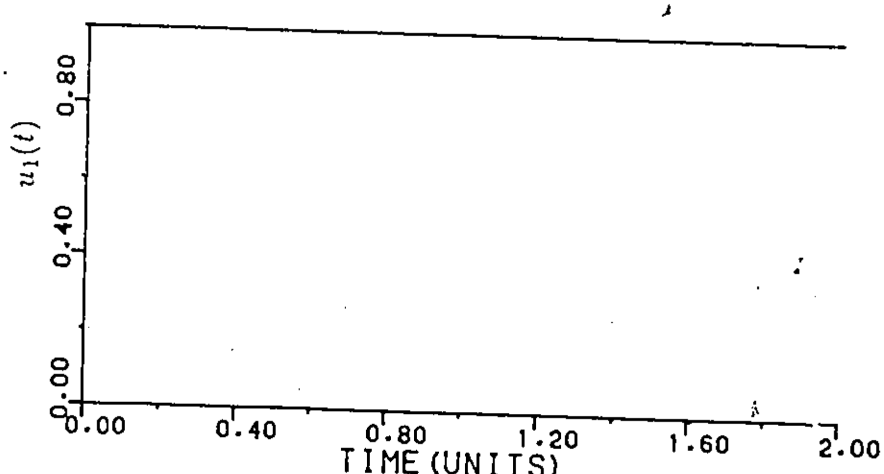


Fig.3.18. Optimal Control u_1^*.

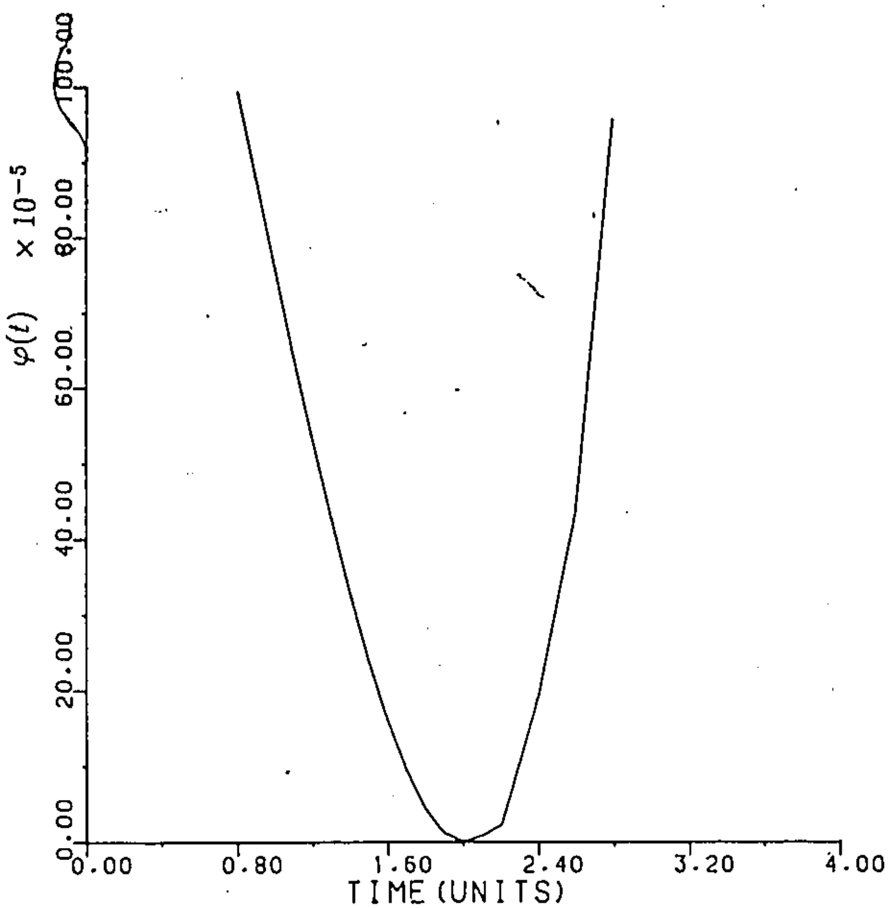


Fig.3.19. The Cost Function φ

3.4 SUMMARY

In this chapter, three numerical examples were considered in order to illustrate the effectiveness of the proposed algorithm for time optimal control problem (see chapter II). In the case of the IBM articulated arm robot, an improvement of the optimal time has been achieved. It is clear, from the three examples, that the proposed method provides the optimal control policy for any system, while using poor initial guesses.

In the next chapter we show how the time optimal control algorithm can be easily modified to be used for the identification problem of systems governed by ordinary differential equations.

CHAPTER IV

NUMERICAL SIMULATION FOR SYSTEM IDENTIFICATION WITH APPLICATIONS TO ROBOTICS AND ECOLOGY

4.1 INTRODUCTION

An important and essential aspect of modelling any physical system is the identification parameters in the model equation. Usually a physical system is modelled on the basis of intuition, some idealizing assumptions and knowledge of functional relationship between the state variables and their time rate of change. The model equations contains certain parameters which are completely unknown because of lack of precise understanding of the system, or only partly known because of poor measurement data. A fundamental problem in system modelling is, then, determination of these unknown parameters so that the corresponding response of the model equation approximates as closely as possible the actual response of the physical system. A very natural approach for solving this problem is to consider it as a control problem.

For numerical computation of the optimal parameters, we use the gradient method as discussed in chapter II. In fact the algorithm given in chapter II can be easily modified to be used for the identification problem. The algorithm is presented in section 3 and its theoretical basis is given in section 2. Three numerical examples illustrating the usefulness of the algorithm are provided in section 4.

4.2 PROBLEM FORMULATION

In this section, we present a general statement of the parameter identification problem and develop the necessary conditions of optimality.

Consider the system

$$\dot{x}(t) = f(t, x, \alpha), \quad t \in [0, T]$$
$$x(0) = x_0 \tag{4.1}$$

where $x(t) \in R^n$ is the state vector and α, taking values in a closed bounded subset P of R^m, is the unknown parameter. The function $f:\ I \times R^n \times R^m \rightarrow R^n$ is known, except for the vector α. It is assumed that the response of the physical system, which is modelled as (4.1), is given in the form of data $y(t)$, $t \in$ I.

We define an identification error as the mean square difference between the system output and the observed data

$$J(\alpha) = (1/2) \int_0^T \|x(t, \alpha) - y(t)\|^2 dt \tag{4.2}$$

where $x(., \alpha)$ is the response of system equation (4.1) corresponding to the parameter α and the initial condition x_0.

The identification problem may be stated as follows:

Find a parameter $\alpha^* \in$ P such that $J(\alpha^*) \leq J(\alpha)$ for all $\alpha \in$ P, where $J(\alpha)$ is given by (4.2) for all $\alpha \subset R^m$.

In view of the above discussion, it is clear that a very natural approach for solving the identification problem is to consider it as a control problem. In fact this could be considered as a special case of the general control problem (see chapter II) where the parameter α can be regarded as the control vector. Using the minimum principle as given in theorem 2.1 (see chapter II), we can derive a set of necessary conditions of optimality for the identification problem. This is given below.

Theorem 4.1(Necessary Conditions of Optimality for Identification)

In order that the $\alpha^* \in$ P or equivalently the pair $\{\alpha^*, x^*\}$ to be optimal it is necessary that there exists $\psi \in AC(\mathrm{I}, R^n)$ I=[0,T], (absolutely continuous function on I with values in R^n) such that the triple $\{\alpha^*, x^*, \psi^*\}$ satisfies the following differential equations and inequalities :

(i) $\dot{x}(t) = f(t, x(t), \alpha)$ for a.e. $t \in (0, T]$

$x(0) = x_0$

(ii) $\dot{\psi}(t) = -f_x{}'(t, x(t), \alpha)\psi(t) + (y(t) - x(t))$

$\psi(T) = 0$ (4.3)

and the inequality

(iii) $\int_0^T H(t, x^*(t), \psi^*(t), \alpha^*)dt \leq \int_0^T H(t, x^*(t), \psi^*(t), \alpha)dt$

for all $\alpha \in P$

where

$$H(t, x, \psi, \alpha) = (f(t, x, \alpha), \psi) + (1/2)\|x(t, \alpha) - y(t)\|^2 \qquad (4.4)$$

4.3 ALGORITHM and FLOWCHART

4.3.1 Algorithm

Based on the necessary conditions of optimality presented in the preceding section, an iterative algorithm can be developed for the determination of the optimal parameter α^* using the gradient technique as discussed in the chapter II. The time optimal control algorithm can be easily modified to be used in this case. Indeed, for the gradient method applied to identification problems, we have the following algorithm :

Step 1 : Guess $\alpha_1 \in$ P and set n=1.

Step 2 : Solve the initial value problem (4.1) with $\alpha = \alpha_1$ giving $x_n(t) = x(t, \alpha_n)$.

Step 3 : Using the data x_n and α_n, solve the adjoint system

$$\begin{aligned} \dot{\psi}(t) &= -f_x{}'(x_n(t), \alpha_n)\psi(t) + (y(t) - x_n(t)) \\ \psi(T) &= 0 \end{aligned} \tag{4.5}$$

backward in time to determine ψ_n.

Step 4 : Compute the gradient vector

$$g_n = \int_0^T (\partial H/\partial \alpha)(t, x_n, \psi_n, \alpha_n)dt \tag{4.6}$$

where H is the Hamiltonian (see equation 4.4)

Step 5 :

(i) If $n = 1$ set $s_1 = -g_n$

(ii) If $n \neq 1$ set $s_n = -g_n + (\|g_n\|^2/\|g_{n-1}\|^2)\ s_{n-1}$

Step 6 :

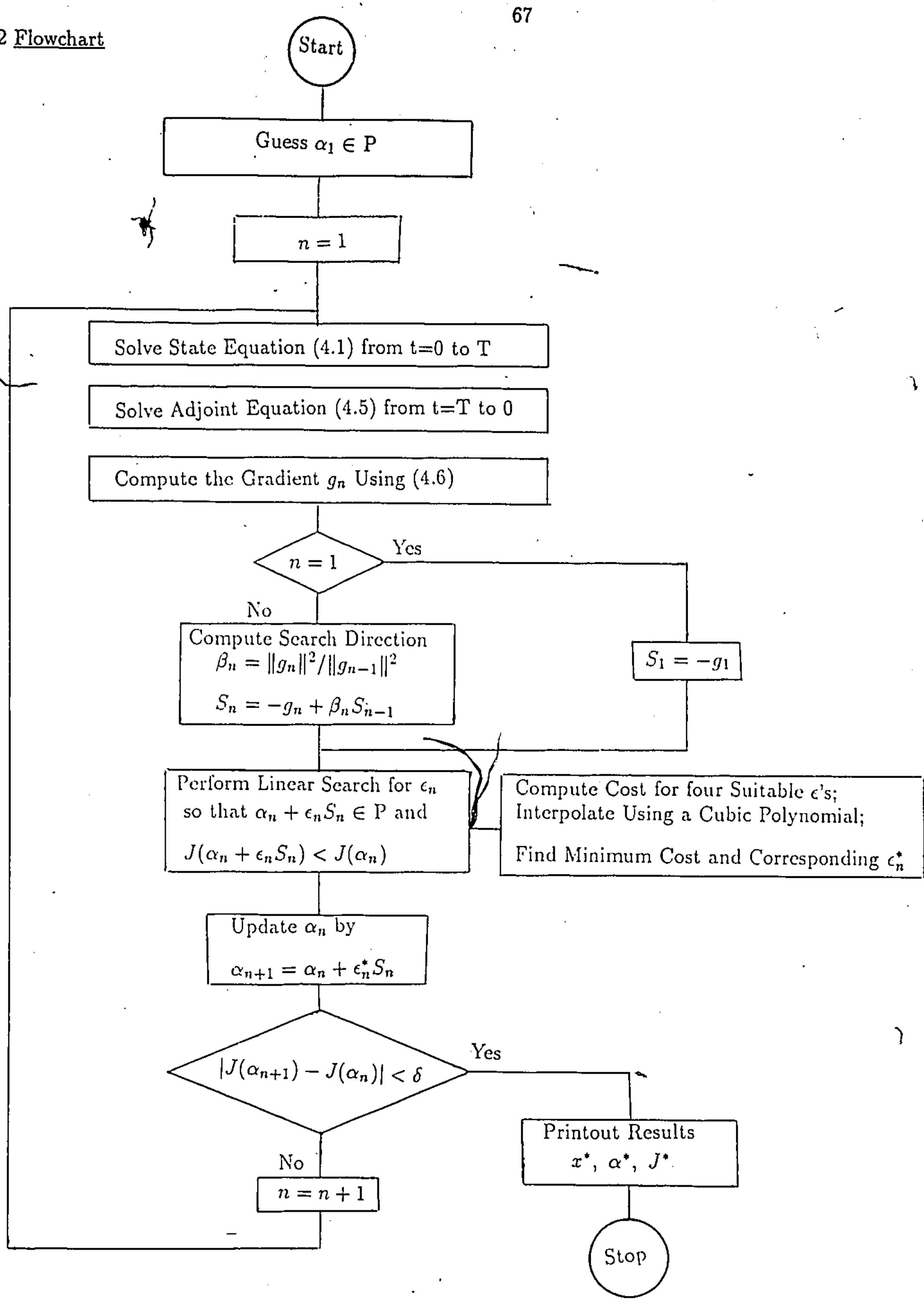
a. If $g_n \neq 0$, then modify α_n to $\alpha_{n+1} = \alpha_n + \epsilon s_n$ by choosing $\epsilon > 0$ sufficiently small so that , $\alpha_{n+1} \in P$ and $J(\alpha_{n+1}) \leq J(\alpha_n)$. A stopping criterion is used at this stage . If $|J(\alpha_{n+1}) - J(\alpha_n)| \leq \delta$, for small $\delta > 0$, then stop, otherwise set $\alpha_n = \alpha_{n+1}, n = n + 1$, and go to step 2.

b. If at the n^{th} stage $g_n = 0$ then α_n is a local minimizing element of $J(\alpha)$.

<u>Remark 4.2</u>

As in any gradient method the iterated parameter would tend to seek a local minimum. This could be considered by repeating the procedure for different initial guesses for the unknown parameter.

4.3.2 Flowchart



4.4 NUMERICAL SIMULATION

The observation data for the actual system i.e. y(t) were generated with known values of the parameter in the equations which, in the sequel, will be referred to as the "true parameter". In order to illustate the effectiveness of our proposed scheme, we consider the identification problem for the three models that have been presented in chapter III.

Example 1: (Automelec ACR robot)

The identification problem arises naturally in the modelling of a robot manipulator. The system model given in chapter III contains certain unknown parameters such as

a. moment of inertia

b. masses of the arm and the load

and

c. lengths of the manipulator links

The true parameter is taken as

$$\alpha^* = [m_k, m_l, d_0, M_z, M_c]' \in R^5$$
$$= [3.7,\ 4.6,\ 0.37,\ 0.28,\ 0.09]'$$

In table (4.1.a)-(4.1.d), we summarize the convergence of the parameter α corresponding to different initial guesses of the unknown parameter.

(i)

Iter.	α_1	α_2	α_3	α_4	α_5	cost
0	1.5	3.1	1.6	1.3	1.0	$.283\times10^{1}$
30	2.978	3.818	0.1811	0.9675×10^{-1}	0.6255×10^{-1}	$.134\times10^{-2}$
80	3.729	4.322	0.3984	0.1781	0.1439	$.268\times10^{-2}$
120	3.849	4.393	0.3884	0.1864	0.1522×10^{-1}	$.138\times10^{-5}$
200	3.885	4.414	0.3855	0.1888	0.1547	$.637\times10^{-10}$

Table 4.1.a.

The initial, computed, and true system trajectories are shown in figure (4.1.a)-(4.1.d).

(ii)

Iter.	α_1	α_2	α_3	α_4	α_5	cost
0	6.0	7.0	0.2	1.5	2.0	$.256\times10^{1}$
10	4.006	6.490	0.1122	0.4301	0.1302	$.274\times10^{-1}$
20	2.667	5.732	0.3261	0.3587	0.5875×10^{-1}	$.102\times10^{-2}$
30	2.569	5.662	0.3075	0.3919	0.9190×10^{-1}	$.431\times10^{-4}$
90	2.614	5.686	0.2993	0.3951	0.9513×10^{-1}	$.359\times10^{-13}$

Table 4.1.b.

(iii)

Iter.	α_1	α_2	α_3	α_4	α_5	cost
0	7.0	1.0	3.0	1.0	1.0	$.279\times10^{1}$
20	4.359	3.496	0.4932	0.2270	0.0	$.243\times10^{-3}$
50	4.489	3.605	0.4927	0.1158	0.274×10^{-2}	$.150\times10^{-4}$
120	4.591	3.707	0.4591	0.1656	0.526×10^{-1}	$.113\times10^{-9}$
140	4.592	3.708	0.4590	0.1657	0.527×10^{-1}	$.193\times10^{-14}$

Table 4.1.c.

It is clear from the above tables, that there exists more than one optimal parameter that provide the desired trajectory.

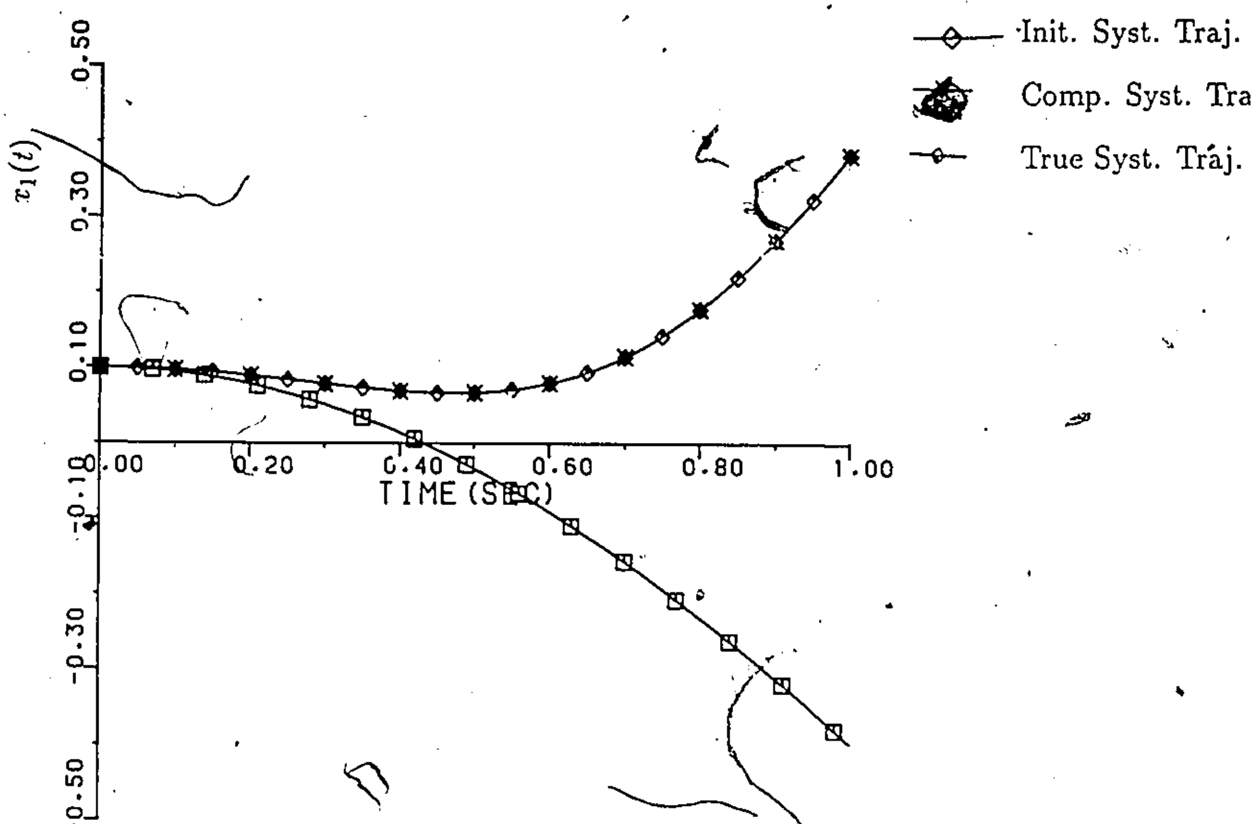


Fig. 4.1.a.

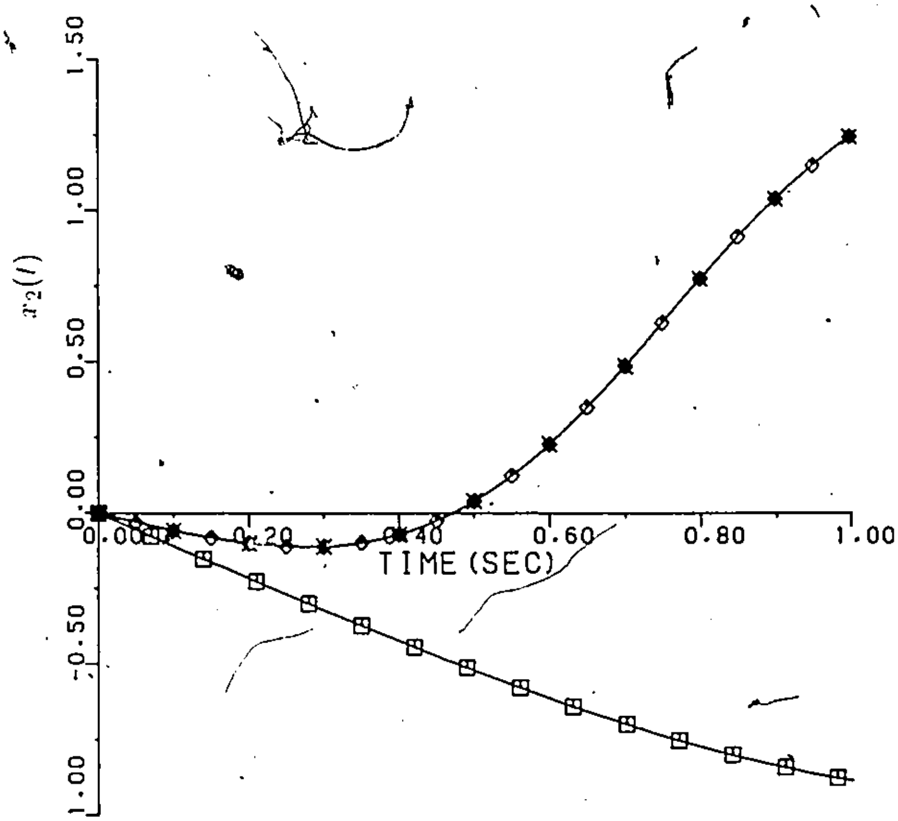


Fig. 4.1.b.

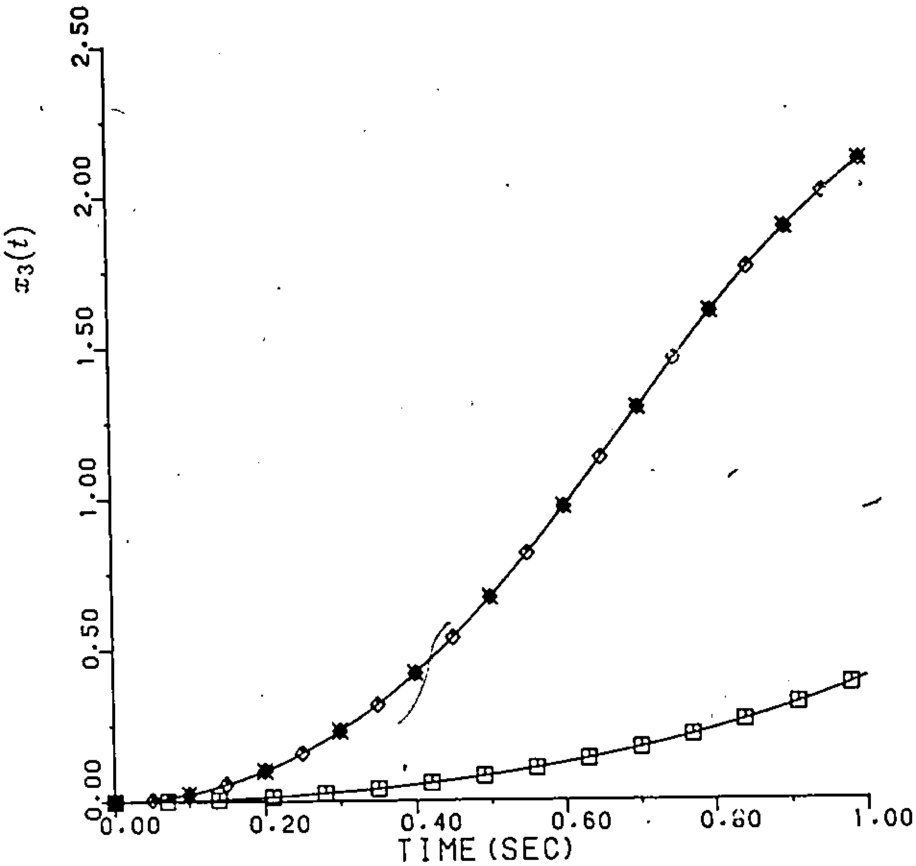


Fig. 4.1.c.

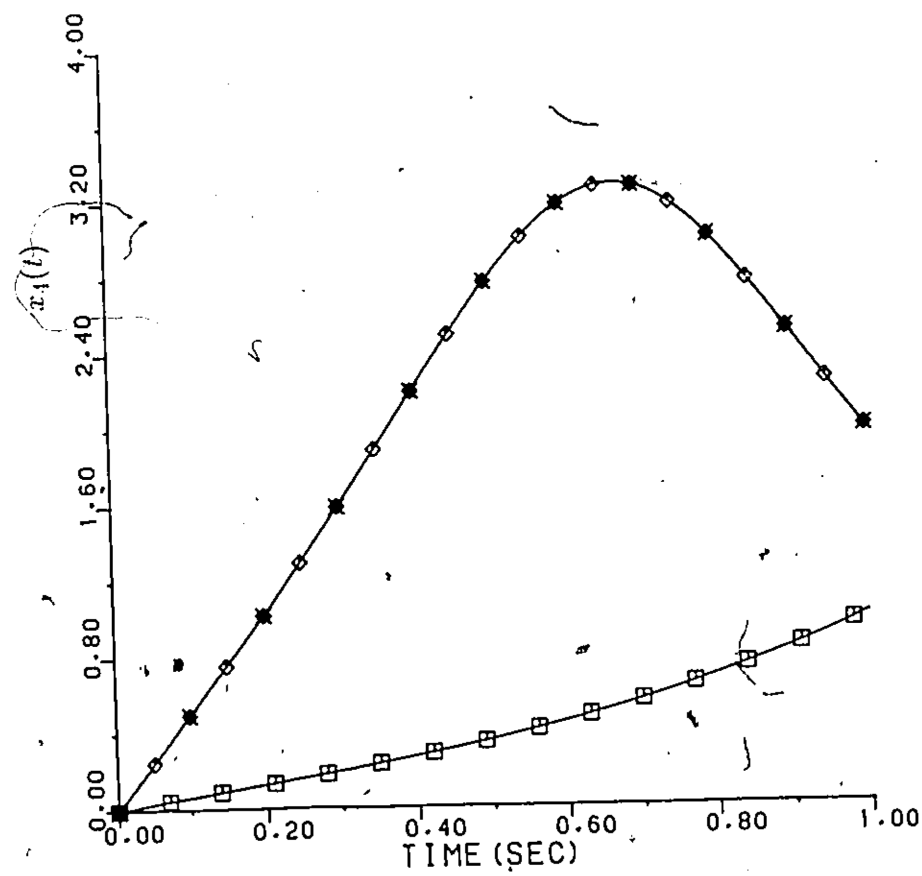


Fig. 4.1.d.

Example 2: (IBM Articulated Arm Robot with two Links)

The true parameter is taken as

$$\alpha^* = [L_1, L_2, d_s, m_s, m_h, M_f, M_s, M_h]' \in R^8$$
$$= [0.4,\ 0.25,\ 0.125,\ 15.0,\ 6.0,\ 1.6,\ 0.43,\ 0.01]'$$

In the following, the unknown vector is computed for two differerent initial guesses as shown in table (4.2.a)-(4.2.b).

CASE A

Iter.	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	cost
0	.15	.30	.35	.10	.20	1.60	.75	.85	$.181\times10^{1}$
10	.2679	.3032	.3332	.5076	.6507	1.929	.5310	.6457	$.559\times10^{-2}$
20	.2761	.3029	.3329	.5362	.6798	1.962	.5133	.6321	$.157\times10^{-7}$
55	.2760	.3029	.3329	.5361	.6798	1.962	.5132	.6320	$.108\times10^{-16}$

Table 4.2.a.

The initial, computed and true system trajectories are shown in figure (4.2.a)-(4.2.d).

CASE B

Iter.	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	cost
0	1.0	.70	.40	2.0	.80	1.0	.20	.70	$.344\times10^{1}$
10	1.153	.5534	.2534	1.981	.8634	1.067	.0905	.5397	$.592\times10^{-2}$
60	1.175	.5536	.2535	1.996	.8802	1.147	.0942	.5483	$.425\times10^{-7}$
70	1.176	.5535	.2535	1.995	.8803	1.147	.0941	.5482	$.153\times10^{-9}$

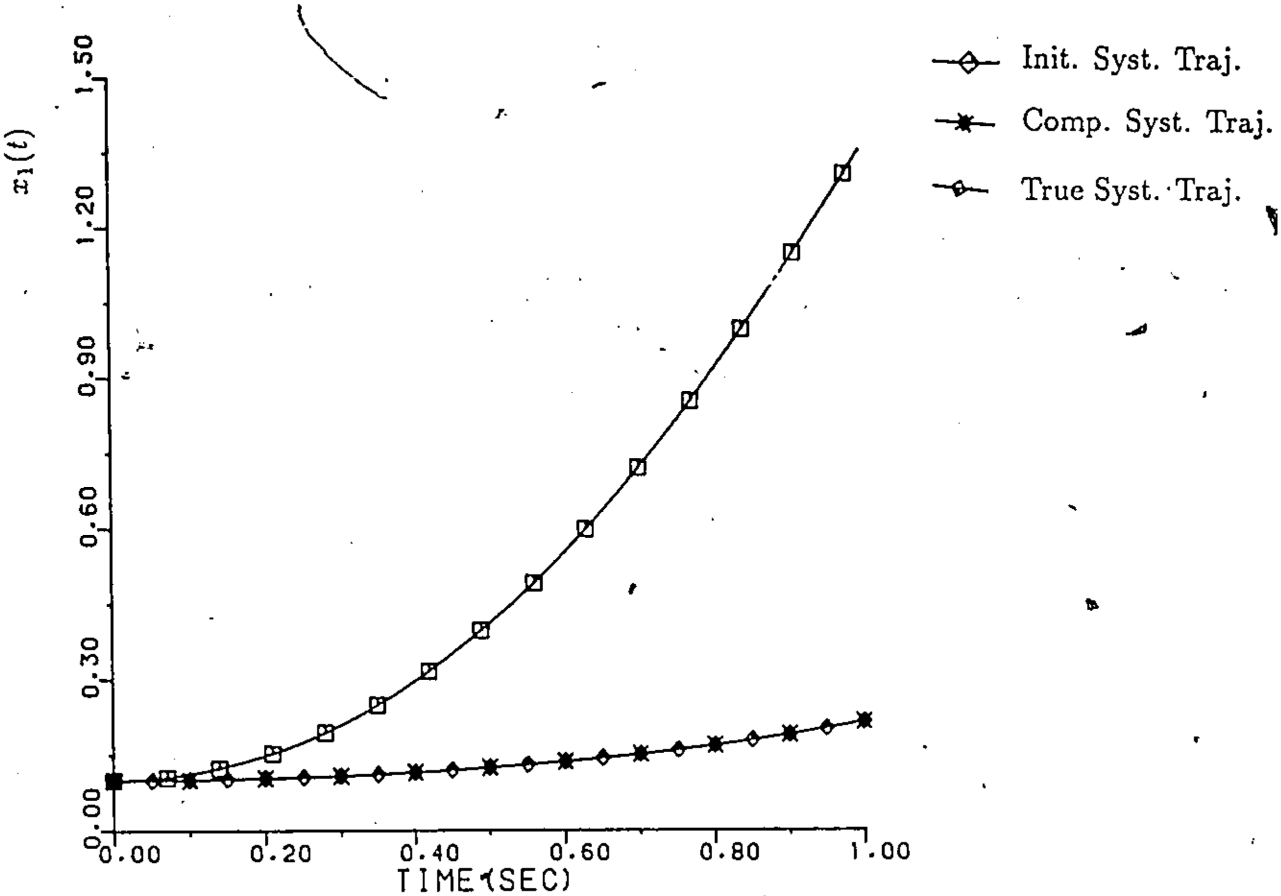


Fig.4.2.a

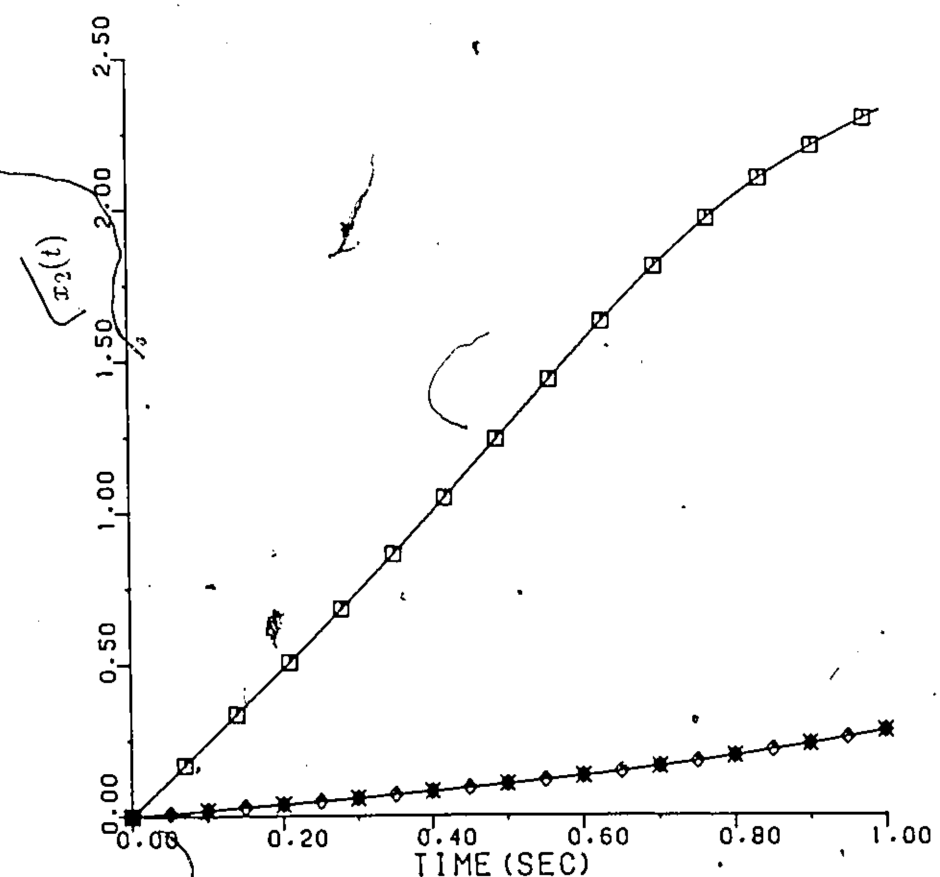


Fig.4.2.b

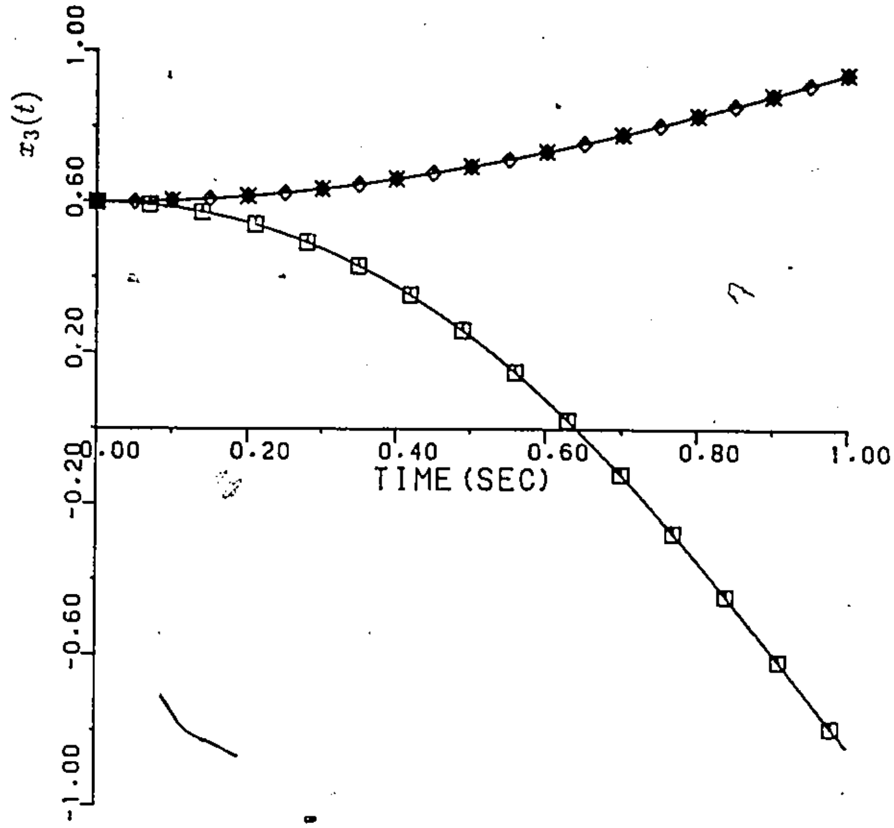


Fig.4.2.c

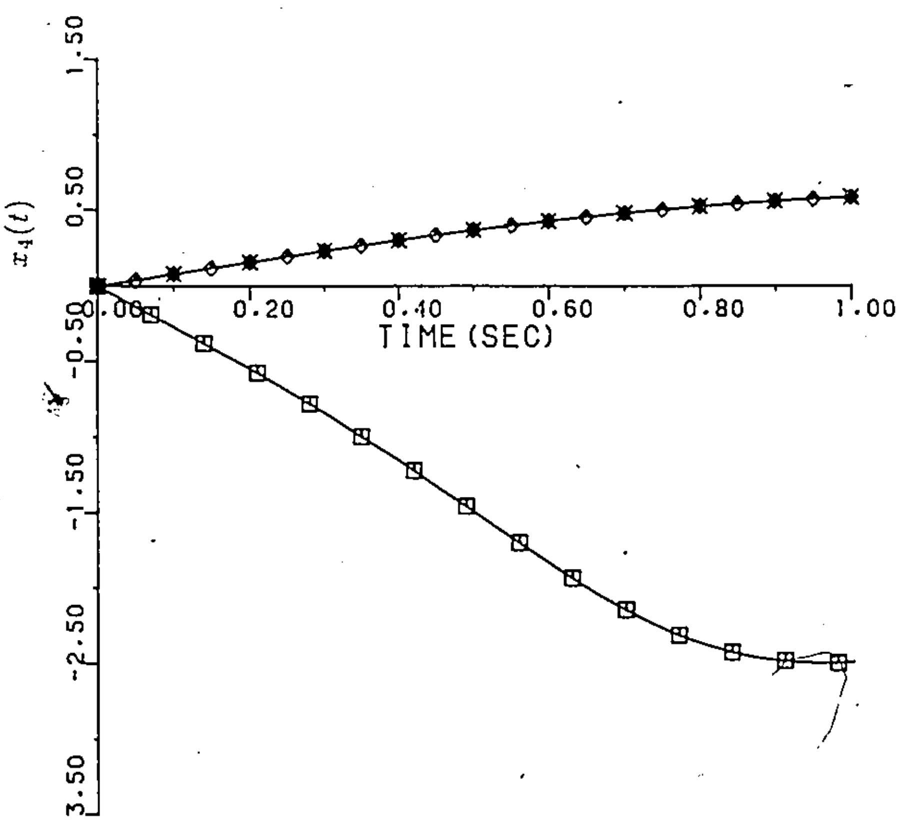


Fig.4.2.d

<u>Example 3:</u> (Ecological System)

<u>CASE A:</u> The true parameter is taken as

$$\alpha^* = [a, b, c, d, \lambda, \beta, \gamma, \delta]' \in R^8$$
$$= [0.5, 300, 10.5, 100, 1, 10^{-3}, 1, 3 \times 10^4]'$$

The initial guess of the unknown parameter is taken as

α_g=[1, 20, 7, 11, 2, 1, 6, 900]'

The computed parameter is

α_c=[.565, 3.8, 2.5, 1.0, 1.23, 4.13, 2.21, 667.17]'

The initial, computed, and true system trajectories are shown in figure (4.3.a)-4.3.c).

<u>CASE B:</u> The true parameter is taken as

$$\alpha^* = [a, \beta, \gamma]' \in R^3$$
$$= [0.5, 10^{-3}, 1.0]'$$

The remaining terms are taken constant.

The computed parameter is

$\alpha_c = [0.50001, 10^{-3}, 1.00003]'$.

In table (4.3), the error function is given as a function of the number of iterations.

Iter.	cost
0	$.243\times10^{6}$
10	.197
20	$.522\times10^{-2}$
30	$.931\times10^{-2}$
40	$.300\times10^{-4}$
50	$.376\times10^{-5}$
80	$.144\times10^{-11}$

Table 4.3.

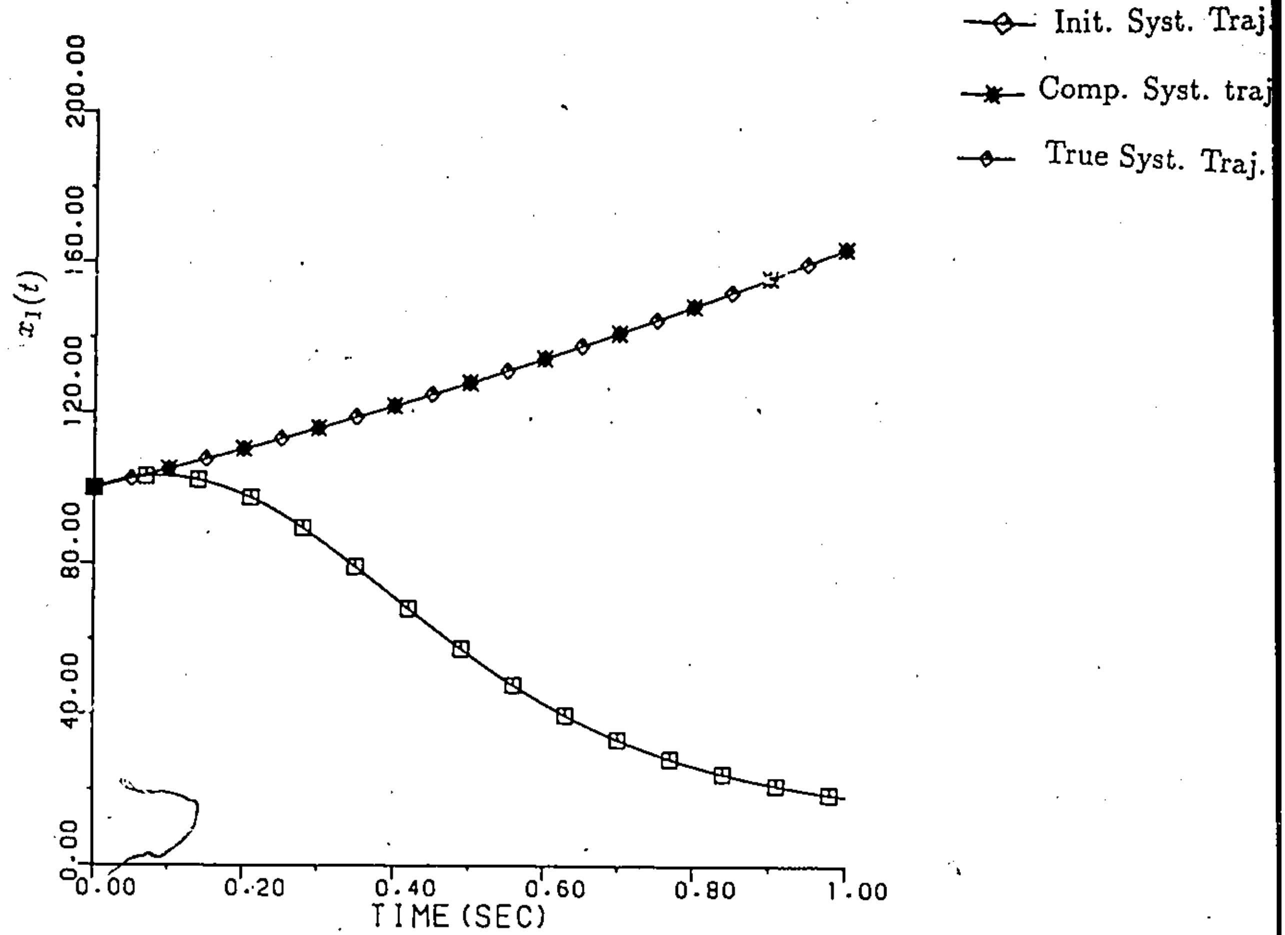


Fig. 4.3.a.

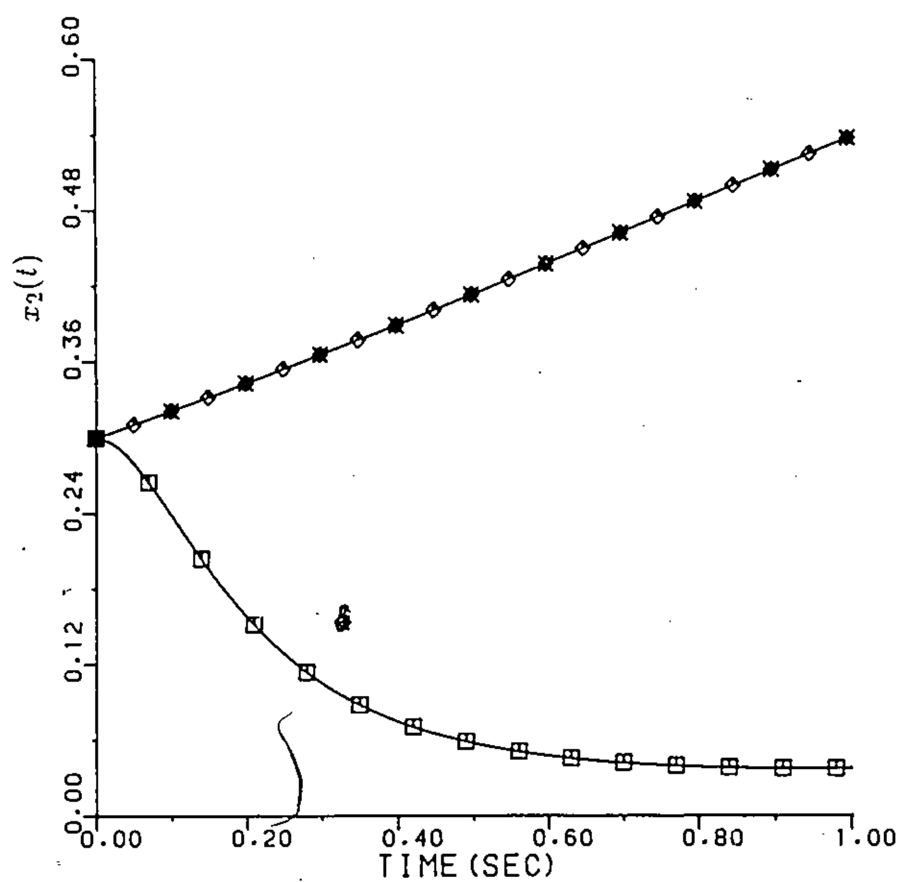


Fig. 4.3.b.

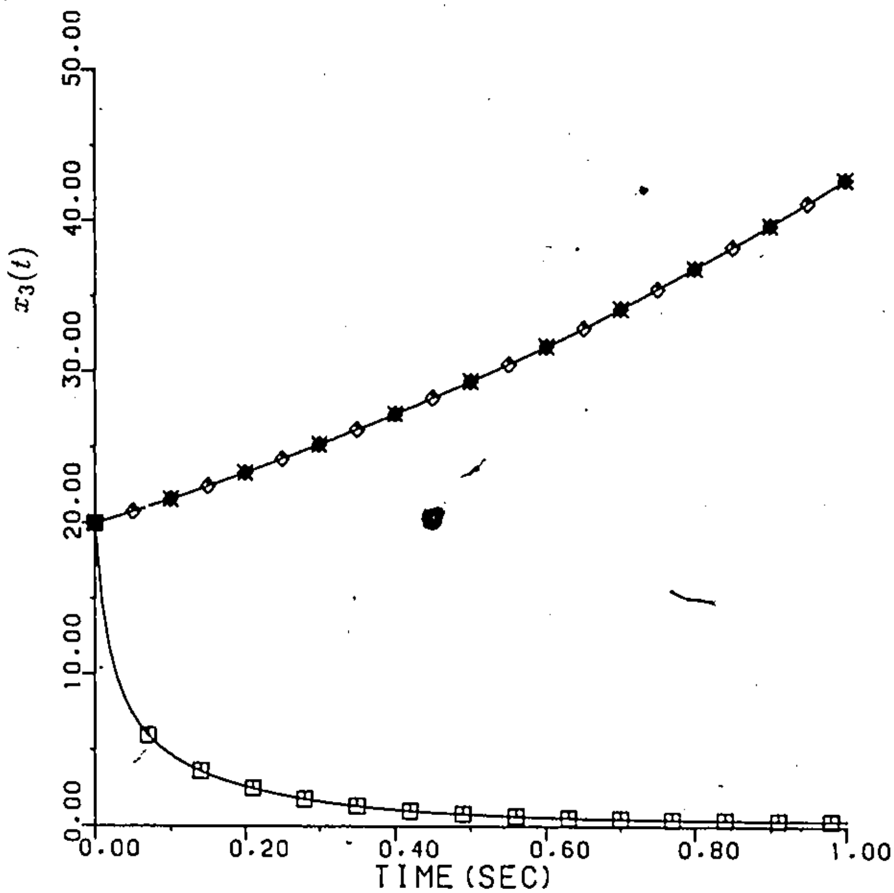

50.00
40.00
30.00
20.00
10.00
0.00
$x_3(t)$
0.00
0.20
0.40
0.60
0.80
1.00
TIME (SEC)

Fig. 4.3.c.

4.5 SUMMARY

In this chapter, we have considered the identification problem for systems described by a set of ordinary differential equations.

Utilizing optimal control theory, we have developed the necessary conditions of optimality on the basis of which the unknown parameter can be computed. It has been shown that the optimal parameter (minimizing the mean square difference between the observed data and the response of the model equation) is determined by simultaneous solution of the system equation, the corresponding adjoint equation, and an associated maximality condition.

Using the gradient technique, an algorithm for computing the unknown parameter has been developed. Finally, this algorithm has been used for identifying some of the parameters of two robots and the ecological system which have been discussed in chapter III.

CHAPTER V

CONCLUSIONS

In the first part of this thesis, we have presented an algorithm for computing the time optimal controls. The algorithm is based on approximation of time optimal controls by a sequence of controls which are optimal for a suitable class of terminal control problems. These later problems can be easily solved by conjugate gradient technique. The algorithm avoids the solution of a two point boundary value problem which is a difficult task. In order tc illustrate the usefulness of the proposed method, numerical simulation has been carried out for two robots (Automelec ACR robot and IBM articulated arm robot with two links) and an ecological system. For the Automelec ACR robot, the optimal time coincides with the results of Geering et al [13]. However, in the case of the IBM articulated arm robot, an improvement of the optimal time has been achieved (compared to that given in [13]).

In the second part of this thesis, we have considered the identification problem for systems described by a set of ordinary differential equations. In fact the algorithm that has been proposed for time optimal control problem was modified to be used for identification problem.

Utilizing optimal control theory, we have developed the necessary conditions of optimality on the basis of which the unknown parameter can be computed. It has been shown that the optimal parameter (minimizing the mean square difference between the observed data and the response of the model equation) is determined by simultaneous solution of the system equation, the corresponding adjoint equation, and an associated maximality condition.

In order to illustate the usefulness of this algorithm three numerical examples including two robots and an ecological system were considered. The numerical simulation has been carried out for different initial choices of the unknown parameter showing the existence of more than one optimal parameter for each initial choice.

APPENDIX A

LAGRANGE EXTRAPOLATION

To update the control in equation (2.18 chapter II), the step size (α) is determined using Lagrange Extrapolation. First, we consider that the cost function J(u) can be approximated by the following polonomial

$$J(u) = a\alpha^3 + b\alpha^2 + c\alpha + d \tag{A.1}$$

where (a,b,c,d)$\in R^4$.

Then four suitable values of α are chosen so as to compute a,b,c and d.

The best α that can give minimum cost in (A.1) is found by differentiating J(u) with respect to α and setting the result to zero. This is shown below

$$(\partial J(\alpha))/(\partial(\alpha)) = 3a\alpha^2 + 2b\alpha + c = 0 \tag{A.2}$$

From (A.2)

$$\alpha_{\min} = (-b \pm \sqrt{b^2 - 3ac})/3a$$

```
                   APPENDIX B



                   FORTRAN CODES

C     IN THE FOLLOWING THE FORTRAN CODES IN CHAPTER III ARE EXIBITED.
C
C     A PROGRAM FOR TIME OPTIMAL CONTROL PROBLEM
C
C     MAXIMUM PRINCIPLE HAS BEEN USED
C
C     ITERATION USING CONJUGATE GRADIENT METHOD
C
C     USER SUPPLIIED SUBROUTINES ARE
C
C     INTCON, SYSTEM, COSTATE, INTIAL, TERNAL, CONTRL,
C
C     COST, GRAD, OUTP
C
C     NDIM=NUMBER OF EQUATIONS
C
C     NCTL=NUMBER OF THE UNKNOWN PARAMETERS
C
C     NPTS=NUMBER OF POINTS IN THE TIME INTERVAL
C
C     TIME(1)=INITIAL TIME
C
C     TIME(2)=TERMINAL TIME
C
C     TIME(3)=STEP SIZE
C
C     MAX=MAXIMUM NUMBER OF ITERATIONS
C
C     DELTA=STEP SIZE TO UPDATE THE FINAL TIME
C
C     STOP=STOPPING CRITERION ON COST DIFFERENCE
C
C     ALPHA=STEP SIZE IN THE SEARCH DIRECTION FOR UPDATING THE UNKNOWN
C     VECTOR
C
C
C     MAIN ROUTINE
C
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION YY(4,1501),BNEW(3002),YD(4,1501),B(3002),GRD(3002)
      DIMENSION Y(4),DERY(4),AUX(4,4),YX(4),X(4),YT(4)
      DIMENSION SRH(3002),TIME(3),BT(3002)
      TIME(1)=0.0
      TIME(2)=0.20
      TIME(3)=.001
      DELTA=0.1
      M=1
      NDIM=4
```

```
      MAX=450
      STOP=1.0D-30
      ALPHA=5.0D-3
      NPTS=(TIME(2)/TIME(3))+1
      NCTL=2*NPTS
      CALL INTCON(NCTL,B)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,B,YY)
      CALL COST(NDIM,NCTL,NPTS,TIME,YY,YD,B,CST)
      CALL COGRAD(NDIM,NCTL,NPTS,MAX,TIME,Y,X,YT,DERY,AUX,YX,B,BNEW,YY
     1,Y,SRH,GRD,CST,ALPHA,STOP,BT,DELTA)
      CALL TEMP(NDIM,NCTL,NPTS,MAX,TIME,Y,X,YT,DERY,AUX,YX,B,BNEW,YY,
     1Y,SRH,GRD,CST,ALPHA,STOP,BT)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,B,YY)
        CALL OUTP(NDIM,NPTS,NCTL,TIME,Y,YY,B,M,CST)
      STOP
      END
C
      SUBROUTINE INTCON(NCTL,B)
C
C     INITIALIZATION OF THE CONTROL
C
      DIMENSION B(NCTL)
      DO 12 I=1,NCTL
12    B(I)=0.0
      RETURN
      END
C
      SUBROUTINE NUMBER(NPTS,NCTL,TIME)
C
C     DETERMINATION OF THE NUMBER OF POINTS IN THE TIME INTERVAL
C
      DIMENSION TIME(3)
      NPTS=(TIME(2)/TIME(3))+1
      NCTL=2*NPTS
      RETURN
      END
C
      SUBROUTINE SYSTEM (NDIM,NCTL,T,Y,U,DERY)
C
C     SYSTEM DYNAMICS
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION Y(NDIM),DERY(NDIM),U(2)
      DERY(1)=Y(2)
      DERY(2)=(U(1)+3.7*Y(1)*(Y(4)**2)+4.6*(.37+Y(1))*(Y(4)**2))/
     &(3.7+4.6)
      DERY(3)=Y(4)
      DERY(4)=(U(2)-2*3.7*Y(1)*Y(2)*Y(4)-2*4.6*(.37+Y(1))*Y(2)*Y
     &(4))/(.28+.09+3.7*(Y(1)**2)+4.6*((.37+Y(1))**2))
      RETURN
      END
C
      SUBROUTINE COSTATE (NDIM,NCTL,T,X,Y,YT,U,DERX)
C
C     ADJOINT SYSTEM
```

```
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION X(NDIM),DERX(NDIM),Y(NDIM),U(2),YT(NDIM)
      DERX(1)=-(X(2)*(Y(4)**2)+X(4)*((-2*3.7*Y(2)*Y(4)-2*4.6*Y(2)
     &*Y(4))*(.28+.09+3.7*(Y(1)**2)+4.6*((.37+Y(1))**2))-(2*3.7*Y(1)
     &+2*4.6*(.37+Y(1)))*(U(2)-2*3.7*Y(1)*Y(2)*Y(4)-2*4.6*(.37+Y(1)
     &)*Y(2)*Y(4)))/((.28+.09+3.7*(Y(1)**2)+4.6*((.37+Y(1))**2))**2))
      DERX(2)=-(X(1)+X(4)*(-2*3.7*Y(1)*Y(4)-2*4.6*(.37+Y(1))*Y(4))/(
     &.28+.09+3.7*(Y(1)**2)+4.6*((.37+Y(1))**2)))
      DERX(3)=0.0
            DERX(4)=-(X(2)*(1/8.3)*(2*Y(1)*Y(4)*3.7+4.6*(.37+Y(1))*2*
     &Y(4))+X(3)+X(4)*(-2*3.7*Y(1)*Y(2)-2*4.6*(.37+Y(1))*Y(2))/(
     &.28+.09+3.7*(Y(1)**2)+4.6*((.37+Y(1))**2)))
      RETURN
      END
C
      SUBROUTINE INTIAL(NDIM,Y)
C
C     INITIAL CONDITIONS FOR THE DIFFERENTIAL EQUATIONS
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION Y(NDIM)
      Y(1)=0.15
      Y(2)=0.0
      Y(3)=0.0
      Y(4)=0.0
      RETURN
      END
C
      SUBROUTINE TERNAL(NDIM,NPTS,YY,X)
C
C     TERMINAL CONDITIONS FOR THE COSTATE EQUATIONS
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION X(NDIM),YY(NDIM,NPTS)
      X(1)=YY(1,NPTS)-.15
      X(2)=YY(2,NPTS)
      X(3)=YY(3,NPTS)-1.570796
      X(4)=YY(4,NPTS)
      RETURN
      END
C
      SUBROUTINE CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
C
C     CONTROL UPDATE
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION B(NCTL),BNEW(NCTL),SRH(NCTL)
      DO 1 I=1,NCTL
    1 BNEW(I)=B(I)+STEP*SRH(I)
      DO 2 I=1,NPTS
C
C     CONTROL CONSTRAINTS
C
      IF(BNEW(I).GT.15.0) BNEW(I)=15.0000
```

```
      IF(BNEW(I).LT.-15.0) BNEW(I)=-15.0
      IF(BNEW(I+NPTS).GT.5.0) BNEW(I+NPTS)=5.0
      IF(BNEW(I+NPTS).LT.-5.0) BNEW(I+NPTS)=-5.0
2     CONTINUE
      RETURN
      END
C
      SUBROUTINE COST(NDIM,NCTL,NPTS,TIME,YY,YD,B,CST)
C
C     COST FUNCTION
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION YY(NDIM,NPTS),B(NCTL),YD(NDIM,NPTS),TIME(3)
      CST=0.5*((YY(1,NPTS)-.15)**2+YY(2,NPTS)**2+(YY(3,NPTS)-1.570796)
     &**2+YY(4,NPTS)**2)
      RETURN
      END
C
      SUBROUTINE GRAD(NPTS,NDIM,NCTL,Y,X,B,GRD,K)
C
C     GRADIENT SUBROUTINE
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION Y(NDIM),X(NDIM),B(NCTL),GRD(NCTL)
      GRD(K)=X(2)/8.3
      GRD(K+NPTS)=X(4)/(.28+.09+3.7*(Y(1)**2)+4.6*((.37+Y(1))**2))
      RETURN
      END
C
      SUBROUTINE FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,B,YY)
C
C     RUNGA-KUTTA SUBROUTINE FOR FORWARD SOLUTION OF SYSTEM EQUATION
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION YY(NDIM,NPTS),B(NCTL)
      DIMENSION Y(NDIM),DERY(NDIM),AUX(4,NDIM),YX(NDIM)
      DIMENSION TIME(3),U(2)
      CALL INTIAL(NDIM,Y)
      X=TIME(1)
      H=TIME(3)
      K=1
      DO 1 I=1,NDIM
1     YY(I,K)=Y(I)
      DO 11 J=2,NPTS
      U(1)=B(K)
      U(2)=B(K+NPTS)
      XX=X
      CALL SYSTEM (NDIM,NCTL,XX,Y,U,DERY)
      DO 2 I=1,NDIM
      AUX(1,I)=H*DERY(I)
2     YX(I)=Y(I)+AUX(1,I)*0.5
      XX=X+0.5*H
      CALL SYSTEM (NDIM,NCTL,XX,YX,U,DERY)
      DO 3 I=1,NDIM
      AUX(2,I)=H*DERY(I)
```

```
3     YX(I)=Y(I)+AUX(2,I)*0.5
      XX=X+0.5*H
      CALL SYSTEM (NDIM,NCTL,XX,YX,U,DERY)
      DO 4 I=1,NDIM
      AUX(3,I)=H*DERY(I)
4     YX(I)=Y(I)+AUX(3,I)
      XX=X+H
      K=K+1
      U(1)=B(K)
      U(2)=B(K+NPTS)
      CALL SYSTEM (NDIM,NCTL,XX,YX,U,DERY)
      DO 5 I=1,NDIM
5     AUX(4,I)=H*DERY(I)
      DO 8 I=1,NDIM
      DY=(AUX(1,I)+2.0*AUX(2,I)+2.0*AUX(3,I)+AUX(4,I))/6.0
8     Y(I)=Y(I)+DY
      X=X+H
      DO 10 I=1,NDIM
10     YY(I,K)=Y(I)
11    CONTINUE
      RETURN
      END
C
      SUBROUTINE BACKWD(NDIM,NPTS,NCTL,TIME,Y,YT,DERX,AUX,YX,X,B,YY,YD
     1,GRD)
C
C     RUNGA-KUTTA SUBROUTINE FOR BACKWARD SOLUTION OF ADJOINT EQUATIONS
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION YY(NDIM,NPTS),GRD(NCTL),B(NCTL),YD(NDIM,NPTS)
      DIMENSION Y(NDIM),DERX(NDIM),AUX(4,NDIM),YX(NDIM),X(NDIM)
      DIMENSION TIME(3),YT(NDIM),U(2)
      CALL TERNAL(NDIM,NPTS,YY,X)
      T=TIME(2)
      H=-TIME(3)
      K=NPTS
      DO 11 J=2,NPTS
      U(1)=B(K)
      U(2)=B(K+NPTS)
      DO 9 I=1,NDIM
9     Y(I)=YY(I,K)
      CALL GRAD(NPTS,NDIM,NCTL,Y,X,B,GRD,K)
      TT=T
      CALL COSTATE (NDIM,NCTL,TT,X,Y,YT,U,DERX)
      DO 2 I=1,NDIM
      AUX(1,I)=H*DERX(I)
2     YX(I)=X(I)+AUX(1,I)*0.5
      TT=TT+0.5*H
      CALL COSTATE (NDIM,NCTL,TT,YX,Y,YT,U,DERX)
      DO 3 I=1,NDIM
      AUX(2,I)=H*DERX(I)
3     YX(I)=X(I)+AUX(2,I)*0.5
      TT=T+0.5*H
      CALL COSTATE (NDIM,NCTL,TT,YX,Y,YT,U,DERX)
      DO 4 I=1,NDIM
```

```
      AUX(3,I)=H*DERX(I)
4     YX(I)=X(I)+AUX(3,I)
      TT=T+H
      K=K-1
      DO 17 L=1,NDIM
17    Y(L)=YY(L,K)
      U(1)=B(K)
      U(2)=B(K+NPTS)
      CALL COSTATE (NDIM,NCTL,TT,YX,Y,YT,U,DERX)
      DO 5 I=1,NDIM
5     AUX(4,I)=H*DERX(I)
      DO 8 I=1,NDIM
      DX=(AUX(1,I)+2.0*AUX(2,I)+2.0*AUX(3,I)+AUX(4,I))/6.0
8     X(I)=X(I)+DX
      T=T+H
      CALL GRAD(NPTS,NDIM,NCTL,Y,X,B,GRD,K)
11    CONTINUE
      RETURN
      END
C
      SUBROUTINE SEARCH(NCTL,GRDN,SRH,VALUO)
C
C     COMPUTES SEARCH DIRECTION
C
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION GRDN(NCTL),SRH(NCTL)
      VALUN=0.0
      DO 2 I=1,NCTL
2     VALUN=VALUN+GRDN(I)*GRDN(I)
      BTA=VALUN/VALUO
      DO 3 I=1,NCTL
3     SRH(I)=-GRDN(I)+BTA*SRH(I)
      VALUO=VALUN
      RETURN
      END
C
      SUBROUTINE START(NCTL,GRDO,SRH,VALUO)
C
C     INITIALIZES THE SEARCH PROCESS
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION GRDO(NCTL),SRH(NCTL)
      VALUO=0.0
      DO 1 I=1,NCTL
      VALUO=VALUO+GRDO(I)*GRDO(I)
1     CONTINUE
      DO 2 I=1,NCTL
2     SRH(I)=-GRDO(I)
      RETURN
      END
C
      SUBROUTINE TEMP(NDIM,NCTL,NPTS,MAX,TIME,Y,X,YT,DERY,AUX,YX,B,
     1 BNEW,YY,Y,SRH,GRD,CST,ALPHA,STOP,BT,DELTA)
C
C     UPDATE THE FINAL TIME AND LOCATE THE OPTIMAL TIME
```

```
C
      DIMENSION TIME(3),Y(NDIM),X(NDIM),DERY(NDIM),YT(NDIM)
      DIMENSION AUX(4,4),YX(NDIM),B(NCTL),BNEW(NCTL),YY(NDIM,NPTS)
      DIMENSION Y(NDIM),SRH(NCTL),GRD(NCTL),BT(NCTL)
      T0=TIME(2)
      C1=CST
      T1=TIME(2)
      TIME(2)=TIME(2)+DELTA
      CALL NUMBER(NPTS.NCTL,TIME)
      CALL COGRAD(NDIM,NCTL,NPTS,MAX,TIME,Y,X,YT,DERY,AUX,YX,B,BNEW,
     1YY,Y,SRH,GRD,CST,ALPHA,STOP,BT)
      C2=CST
      T2=TIME(2)
      TIME(2)=T0-DELTA
      CALL NUMBER(NPTS,NCTL,TIME)
      CALL COGRAD(NDIM,NCTL,NPTS,MAX,TIME,Y,X,YT,DERY,AUX,YX,B,BNEW,
     1YY,Y,SRH,GRD,CST,ALPHA,STOP,BT)
      C3=CST
      T3=TIME(2)
      IF(C3-C1) 10,10,20
10    C2=C1
      C1=C3
      T2=T1
      T1=T3
      TIME(2)=T1-DELTA
      CALL SAVE(NCTL,B,BT)
      CALL NUMBER(NPTS,NCTL,TIME)
      CALL COGRAD(NDIM,NCTL,NPTS,MAX,TIME,Y,X,YT,DERY,AUX,YX,B,BNEW,
      YY,,Y,SRH,GRD,CST,ALPHA,STOP,BT)
      C3=CST
      T3=T1
      IF(C3.EQ.0) GO TO 98
      IF(C3-C1) 10,10,99
20    IF(C1-C2) 50,50,99
50    C3=C1
      T3=T1
      C1=C2
      T1=T2
      TIME(2)=T1+DELTA
      CALL SAVE(NCTL,B,BT)
      CALL NUMBER(NPTS,NCTL,TIME)
      CALL COGRAD(NDIM,NCTL,NPTS,MAX,TIME,Y,X,YT,DERY,AUX,YX,B,BNEW,
     1YY,Y,SRH,GRD,CST,ALPHA,STOP,BT)
      C2=CST
      T2=TIME(2)
      IF(C2.EQ.0) GO TO 98
      GO TO 20
99    TIME(2)=T3
      CST=C(3)
      CALL NUMBER(NPTS,NCTL,TIME)
      CALL SAVE(NCTL,BT,B)
98    RETURN
      END
C
      SUBROUTINE SAVE(NCTL,B,BT)
```

```
C
C     SAVE THE OPTIMAL CONTROL
C
      DIMENSION B(NCTL),BT(NCTL)
      DO 100 I=1,NCTL
100   BT(I)=B(I)
      RETURN
      END
C
      SUBROUTINE MINAP(NDIM,NCTL,NPTS,TIME,Y,DERY,AUX,YX,B,BNEW,YNEW,YD,
     1SRH,CSTOD;STEP,INN)
C
C     UPDATES THE PARAMETER VECTOR
C
C     LAGRANGE EXTRAPOLATION HAS BEEN USED
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION X(4),C(4),TIME(3)
      DIMENSION YNEW(NDIM,NPTS),SRH(NCTL),BNEW(NCTL),B(NCTL)
      DIMENSION Y(NDIM),DERY(NDIM),AUX(4,NDIM),YX(NDIM)
      DIMENSION YD(NDIM,NPTS)
      N1=1
      N2=1
      CALL CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,BNEW,YNEW)
      CALL COST(NDIM,NCTL,NPTS,TIME,YNEW,YD,BNEW,CST)
      C(1)=CST
      X(1)=STEP
      STEP=2.0*X(1)
      CALL CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,BNEW,YNEW)
      CALL COST(NDIM,NCTL,NPTS,TIME,YNEW,YD,BNEW,CST)
      C(2)=CST
      X(2)=STEP
      IF(C(1)-C(2)) 3,3,4
3     STEP=0.5*X(1)
21    CALL CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,BNEW,YNEW)
      CALL COST(NDIM,NCTL,NPTS,TIME,YNEW,YD,BNEW,CST)
      C(3)=C(2)
      X(3)=X(2)
      C(2)=C(1)
      X(2)=X(1)
      C(1)=CST
      X(1)=STEP
      IF(C(1)-C(2)) 9,9,10
9     STEP=0.5*X(1)
      N1=N1+1
      IF(N1.GT.50) GO TO 888
      CALL CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,BNEW,YNEW)
      CALL COST(NDIM,NCTL,NPTS,TIME,YNEW,YD,BNEW,CST)
      C(4)=C(3)
      X(4)=X(3)
      C(3)=C(2)
```

```
      X(3)=X(2)
      C(2)=C(1)
      X(2)=X(1)
      C(1)=CST
      X(1)=STEP
      IF(C(1)-C(2)) 9,9,100
4     STEP=2.0*X(2)
22    CALL CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,BNEW,YNEW)
      CALL COST(NDIM,NCTL,NPTS,TIME,YNEW,YD,BNEW,CST)
      C(3)=CST
      X(3)=STEP
      IF(C(2)-C(3)) 200,19,19
19    STEP=2.0*X(3)
      N2=N2+1
      IF(N2.GT.50) GO TO 888
      CALL CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,BNEW,YNEW)
      CALL COST(NDIM,NCTL,NPTS,TIME,YNEW,YD,BNEW,CST)
      C(4)=CST
      X(4)=STEP
      IF(C(3)-C(4)) 100,301,301
301   C(1)=C(2)
      X(1)=X(2)
      C(2)=C(3)
      X(2)=X(3)
      C(3)=C(4)
      X(3)=X(4)
      GO TO 19
10    STEP=(X(1)+X(2))*0.5
      CALL CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,BNEW,YNEW)
      CALL COST(NDIM,NCTL,NPTS,TIME,YNEW,YD,BNEW,CST)
      C(4)=C(3)
      X(4)=X(3)
      C(3)=C(2)
      X(3)=X(2)
      C(2)=CST
      X(2)=STEP
      GO TO 100
200   STEP=(X(2)+X(3))*0.5
      CALL CONTRL(NPTS,NCTL,STEP,B,BNEW,SRH)
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,BNEW,YNEW)
      CALL COST(NDIM,NCTL,NPTS,TIME,YNEW,YD,BNEW,CST)
      C(4)=C(3)
      X(4)=X(3)
      C(3)=CST
      X(3)=STEP
      GO TO 100
100   CALL POLYN(C,X,STEP,INN)
C
       WRITE(6,345)
345   FORMAT(10X,'LAGRANGE EXTRAPOLATION'/)
       WRITE(6,5) (X(I),I=1,4)
       WRITE(6,6) (C(I),I=1,4)
```

```
5       FORMAT(/ 8X,'ALPHA',4D15.5/)
6       FORMAT( 8X,'COST ',4D15.5)
        CALL CONTRL(NPTS,NCTL,STEP,B,B,SRH)
        GO TO 889
888     INN=2
889       RETURN
        END
C
        SUBROUTINE POLYN(C,X,ALPMIN,INN)
C
C       SUBROUTINE FOR LAGRANGE EXTRAPOLATION
C
        IMPLICIT REAL *8 (A-H,O-Z)
        DIMENSION C(4),X(4)
        D1=((X(1)-X(2))*(X(1)-X(3))*(X(1)-X(4)))
        IF(D1.EQ.0.0) GO TO 77
        F1=C(1)/D1
        D2=((X(2)-X(1))*(X(2)-X(3))*(X(2)-X(4)))
        IF(D2.EQ.0.0) GO TO 77
        F2=C(2)/D2
        D3=((X(3)-X(2))*(X(3)-X(1))*(X(3)-X(4)))
        IF(D3.EQ.0.0) GO TO 77
        F3=C(3)/D3
        D4=((X(4)-X(2))*(X(4)-X(3))*(X(4)-X(1)))
        IF(D4.EQ.0.0) GO TO 77
        F4=C(4)/D4
        A=F1+F2+F3+F4
        B=F1*(X(2)+X(3)+X(4))+F2*(X(1)+X(3)+X(4))+F3*(X(1)+X(2)+X(4))
       1+F4*(X(1)+X(2)+X(3))
        CC=F1*(X(2)*X(3)+X(3)*X(4)+X(2)*X(4))+F2*(X(1)*X(3)+X(1)*X(4)+
       1X(3)*X(4))+F3*(X(1)*X(2)+X(1)*X(4)+X(2)*X(4))+F4*(X(1)*X(2)
       1+X(1)*X(3)+X(2)*X(3))
        S=B*B-3.0*A*CC
        IF(A.EQ.0.0.OR.S.LT.0.0) GO TO 77
        ALPMIN=(B+DSQRT(B*B-3.0*A*CC))/(3.0*A)
        GO TO 78
77      INN=2
78        RETURN
        END
C
        SUBROUTINE COGRAD(NDIM,NCTL,NPTS,MAX,TIME,Y,X,YT,DERY,AUX,YX,B
       1,BNEW,YY,YD,SRH,GRD,CST,ALPHA,STOP,BT)
C
C       CONJUGATE GRADIENT ALGORITHM
C
        IMPLICIT REAL *8(A-H,O-Z)
        DIMENSION YY(NDIM,NPTS),YD(NDIM,NPTS),B(NCTL),BNEW(NCTL),TIME(3)
        DIMENSION Y(NDIM),DERY(NDIM),AUX(4,NDIM),YX(NDIM),SRH(NCTL)
        DIMENSION X(NDIM),GRD(NCTL),YT(NDIM),BT(NCTL)
        ITER=0
        INN=1
        CSTOD=CST
345     CALL BACKWD(NDIM,NPTS,NCTL,TIME,Y,YT,DERY,AUX,YX,X,B,YY,YD,GRD)
        CALL START(NCTL,GRD,SRH,VALU)
         WRITE(6,2) ITER,CSTOD,ALPHA
```

```
2     FORMAT(//5X,'ITER=',I3,3X,'COST=',E20.12,5X,'ALPHA=',E15.7/)
      WRITE(6,3) (B(I),I=1,NCTL)
3     FORMAT(3X,'UPDATED PARAMETERS',4E13.5/)
210   ITER=ITER+1
      DO 77 I=1,NCTL
77    BT(I)=B(I)
      CST1=CST
      KOUNT=0
      ALPHAO=ALPHA
      IF(ITER .GT. MAX) GO TO 999
67    CALL MINAP(NDIM,NCTL,NPTS,TIME,Y,DERY,AUX,YX,B,BNEW,YY,YD,SRH,
     1CSTOD,ALPHA,INN)
      IF(INN.EQ.2) GO TO 999
      CALL FORWRD(NDIM,NPTS,NCTL,TIME,Y,DERY,AUX,YX,B,YY)
      CALL COST(NDIM,NCTL,NPTS,TIME,YY,YD,B,CST)
       WRITE(6,2) ITER,CST,ALPHA
      WRITE(6,3) (B(I),I=1,NCTL)
      IF(ITER .GT. MAX) GO TO 999
       IF(ALPHA.GT.1.0E+04) GO TO 999
      CSTDIF=CSTOD-CST
C
C     STOPING CRITERION WITH COST DIFFERENCE
C
C     IF(CSTDIF .LT. STOP) GO TO 999
      IF(CSTOD - CST) 65,65,66
65    GO TO 999
C
      DO 76 I=1,NCTL
76    B(I)=B(I)-ALPHA*SRH(I)
      KOUNT=KOUNT+1
      IF(KOUNT - 3) 67,67,68
68    WRITE(6,69)
69    FORMAT(//10X,'COST DOES NOT DECREASE ANY MORE'//)
      GO TO 999
66    IF(CSTDIF.LT.STOP) GO TO 999
      CSTOD=CST
      IF(ITER .EQ. NCTL) GO TO 345
      CALL BACKWD(NDIM,NPTS,NCTL,TIME,Y,YT,DERY,AUX,YX,X,B,YY,YD,GRD)
      CALL SEARCH(NCTL,GRD,SRH,VALU)
      GO TO 210
999   DO 78 I=1,NCTL
78    B(I)=BT(I)
      CST=CST1
      RETURN
      END
C
      SUBROUTINE OUTP(NDIM,NPTS,NCTL,PRMT,Y,YY,B,M,CST)
C
C     OUTPUT SUBROUTINE FOR PRINTOUT OF SYSTEM TRAJECTORY
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION Y(NDIM),YY(NDIM,NPTS),B(NCTL),PRMT(3)
      WRITE(6,16) PRMT(2),CST
16    FORMAT(T5,F9.4,4X,E12.5)
      T=PRMT(1)
```

```
      DO 222 L=1,NPTS
      DO 111 I=1,NDIM
111      Y(I)=YY(I,L)
      WRITE(6,12) T,B(L),B(L+NPTS)
      TT=M
      T=T+TT*PRMT(3)
12    FORMAT(F8.3,F7.3,1X,F7.3)
222   CONTINUE
      RETURN
      END
```

REFERENCES

[1] Kahn M.E. and Roth B., 1971, "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," Journal of Dynamic Systems, Measurement and Control, vol.93, No.3, Sept.1971, pp.164-172.

[2] G.N. Saridis and C.S.G.Lee, "Approximation Theory of Optimal Control for Trainable Manipulators," IEEE Trans. Syst. Man Cybern., vol. SMC-9, pp. 152-159, 1979.

[3] R. Gawronski, I.Pardyka, A.Pawlowski, and W.Wolski, " Time-Optimal Hierarchical Control System of Manipulator Movement with Adaptive On-Off Control," Syst. Sci., vol.7, pp.167-177, 1981.

[4] T.L.Johnson, "On Feedback Laws for Robotic Systems," Massachussetts Institute of Technology, Lab. Inform. Sci. and Syst., Cambridge, MA, Rep.AFOSRTR-81-0456, Apr. 1981.

[5] L.K.Huang, "Control of a Tendor Arm," Artificial Intelligence Lab., Mas. Inst. Technology, Rep. AI-M-617, Feb. 1981.

[6] M. Vukobratovic and M. Kircanski, "Method for Optimal Synthesis of Manipulator robot Trajectories," J. Dynamic. Syst. Meas. Control, vol.104, pp.188-193, 1982.

[7] K.G.Shin and N.D.Mckay, "Robot Path Planning Using Dynamic Programming, " in Proc. 23rd Conf. Decision and Control, Las Vegas, NV, Dec.1984, pp.1629-1635.

[8] J.E.Bobrow, S.Dubowsky, and J.S.Gibson, "On the Optimal Control of Robotic Manipulators with Actuator Constraints," in Proc. 1983 Amer. Contr. Conf., San Fransisco, CA, June 1983, pp.782-787.

[9] N.N.Bolotnik and A.A.Kaplunov, "Optimal Rectilinear Moving a Load with a Two-Link

Manipulator," Eng. Cybern., vol.20, pp.121-131, 1982.

[10] C.J.GOH and K.L.TEO, "Miser: An Optimal Control Software. Theory and User Manual." Department of Industrial and Systems Engineering, National University of Singapore, July 1987.

[11] P.Marinov and P.Kiriazow, "A Direct Method for Optimal Control Synthesis of Manipulator Point-to-Point Motion," in Proc. 9th IFAC World Congress, Budapest, Hungary, Aug.1984, paper 14.2/G-5.

[12] T.L. Turner and W.A.Gruver, "Viable Suboptimal Controller for Robotic Manipulators," in Proc. 19th IEEE conf. on Decision and control, Albuquerque, NM, Dec.1980, pp.83-87.

[13] Hans P. Geering, Lino Guzzella, Stephan A.R.Hepner, and Christopher H.Onder, "Time-Optimal Motions of Robots in Assembly Tasks," IEEE Transaction on Automatic Control, vol.AC-31, No.6, June 1986.

[14] N.U. Ahmed, "Elements of Finite Dimensional Systems and Control Theory," Longman Scientific and Technical U.K. England John Wiley and Sons Inc., New York, 1988.

[15] Kelley, H.J., "Gradient Theory of Optimal Flight Paths." ARSJ., 30, 1960, pp.947-954.

[16] Kelley, H.J., "Method of Gradients," G.Leitman, ed. Optimization Techniques. Academic Press, New York, 1962, Chapter 6.

[17] Bryson, A.E., and Denham, W.F., "A Steepest Ascent Method for Solving Optimum Programming Problems." J.Applied Mechanics, 29, 1962, pp.247-257.

[18] Bryson, A.E., et al., "Optimum Programming Problems with Inequality Constraints I: Necessary Conditions for Extermal Solutions." AIAA J., 1,1963, pp.2544-2550.

[19] Denham, W.F., and Bryson, A.E., "Optimal Programming Problems with Inequality Constraints II: Solution by Steepest Ascent." AIAA J., 2, 1964, pp.25-34.

[20] Stancil, R.T., "A New Approach to Steepest Ascent Trajectory Optimization." AIAA J., 2, 1964, pp.1365-1370.

[21] Goldstein, A.A., "On Steepest Descent." SIAM J. control, ser.A, 3, 1965, pp.147-151.

[22] McReynolds, S.R., and Bryson, A.E., "A Successive Sweep Method for Solving Optimal Programming Problems." Proceedings Joint Autom. Control Conf., August 1965, pp.551-555.

[23] Kopp, R.E., and McGill, R., "Several Trajectory Optimization Techniques; Part I: Discussion," A.V. Balakrishman and L.W. Newstadt, eds. Computing Methods in Optimization Problems. Academic Press, New York, 1964, pp.65-89.

[24] Moyer, H.G., and Pinkham, G., "Several Trajectory Optimization Techniques; Part II: Application." A.V. Balakrishman and L.W. Newstadt, eds. Computing Methods in Optimization Problems. Academic Press, New York, 1964, pp.91-105.

[25] Frank H. Clarke, "Optimization and Nonsmooth Analysis," John Wiley and Sons, 1983.

[26] Edward Beltrami, "Mathematics for Dynamic Modelling," Academic Press, New York, 1987.

[27] Joseph Frederic Bonnans, "On an Algorithm for Optimal Control Using Pontryagin's Maximum Principle." SIAM J. Control and Optimization, vol.24, No.3, May 1986, pp.579-588.

[28] Roberts S.M. and Shipman J.s., Two-Point Boundary Value Problems: Shooting Methods, Elsevier, New York, London, Amsterdam, 1972.

[29] Hermes H. and Lasalle J.P., Functional Analysis and Time Optimal Control, Academic Press, New York and London, 1969.

[30] Bang-Dar Cheng and William F.Powers, "Convergence of Gradient-Type Methods On Singular Parameter Optimization Problems." AIAA/AAS Astrodynamics Conference, San Diego Calif., Aug.18-20, 1976.

[31] Avner Friedman, "Foundations of Modern Analysis," Dover Publications, Inc., New York, 1982.