THESIS 1983 W246M

Wang, James Yunn-Jia, 1956-

Page Number 91

- ☐ Images
- ☐ Foldouts
- ☐ Maps

- ☐ Scanned
- ☐ Clean-up
- ☐ PDF

☐ Archive MSF ☐ Projects

Verified

Date

A METHOD FOR THE SYNTHESIS OF OPTIMAL TRAJECTORY OF ROBOTS

By

JAMES YUNN-JIA WANG

Bachelor of Science

National Tsing Hua University

Hsin Chu, Taiwan

Republic of China

1978

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
July, 1983

A METHOD FOR THE SYNTHESIS

OF OPTIMAL TRAJECTORY

OF ROBOTS

Thesis Adviser

Dean of the Graduate College

ACKNOWLEDGMENTS

I wish to express my appreciation to my major adviser Dr. A. H. Soni for his patience, supervision and encouragement.

Appreciation is also expressed to my committee members: Dr. R. L. Lowery and Dr. Gary Young for their time and comments to prepare this thesis.

It is a great pleasure to extend special thanks to Dr. V. Srinivasan for his invaluable instruction, discussion and encouragement. As a matter of fact, this work would never be done without his assistance.

I would also like to express my sincere appreciation to Dr. Gerald Brusewitz for the financial support, understanding and friendship throughout my study at Oklahoma State University.

Special credit is due to Ms. Neisa Lock for her excellent and untiring efforts in typing this thesis.

Finally, I would like to owe special gratitude to my parents, Mr. Chiuen Wang and Mrs. Min-Lou Yi for their love and moral support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Today, Robotics is one of the hottest topics in the United States. People, especially engineers and scientists, are talking about robots almost everywhere at any time. Economic experts believe that an industrial robot is one of the major advancements in improving the U.S. productivity and may provide an avenue to America to survive from current economic depression.

The Robot Institute of America (RIA) defines a robot as a reprogrammable multi-functional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks [28].

Japan is the only country to officially define a robot as a mechanical system which has flexible motion functions analogous to the motion functions of living organisms or combines such motion functions with intelligent functions, and which acts in response to the human will. In this context, intelligent functions mean the ability to perform at least one of the following: judgement, recognition, adaptation or learning [34].

However, from the viewpoint of applied mechanics [24] a robot or manipulator is usually defined as an open-loop chain of binary links as shown in Figure 1, where every link has one degree-of-freedom (DOF) relative to each of its neighbors. One end of the chain is fixed to the

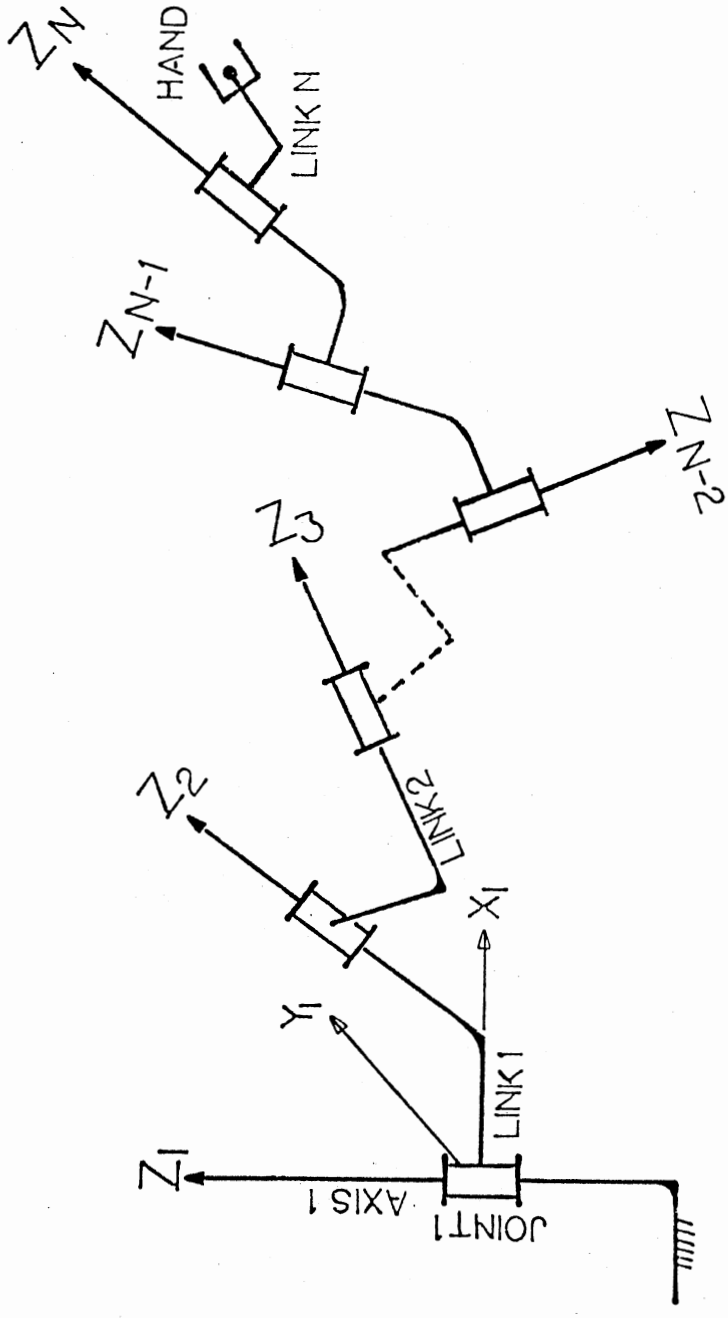

Figure 1. A General n-Link Robot

ground while the other end is free to move in the space. In this report, we will adopt this definition throughout this study.

1.1 Background Review

The origin of mechanical manipulators in America can be traced to the servoed electrical-powered teleoperator developed by Argonne National Laboratory in 1947 for the purpose of handling radio-active materials [28]. This system is essentially a "master-slave" manipulator which could duplicate the hand motions of a person stationed at a remote site. Thirteen years later, in 1960, George Devol demonstrated the first industrial robot under the teaching mode. Since then the manipulator could be taught to perform any simple job by driving it by hand through the sequence of task positions, which were recorded in memory. Task execution consisted in replaying these recorded positions by servoing the individual joint axes of the robot.

Pieper [19], in 1968, guided a programmable computer-controlled manipulator from an initial position to a final position through a space containing obstacles by using a heuristic control algorithm and 4 x 4 transformation matrix to map hand coordinates to joint coordinates. Almost at the same time of Pieper's finishing his work, Whitney [32] initiated the idea "resolved motion rate control" and found that the relatinship between the rate of hand coordinates and that of joint coordinates is linear. Through his work, an operator of a mechanical manipulator can, in real time, command motion control rates of the arm's hand along coordinates axes which are convenient and visible to the operator.

Paul [16], in 1975, proposed a preplanned path control scheme: if the traveling path of a manipulator is preplanned, then one knows exactly the values of desired position, linear velocity, and linear acceleration of the hand with respect to the fixed base coordinates on the entire path. In his scheme, paths were made up of straight line segments connected by smooth transitions with controlled acceleration. Luh [12] combined the idea of "resolved motion rate control" and "preplanned path control" and proposed a "resolved acceleration control" algorithm in which acceleration program of an end-effector is specified along the path.

However, in Paul's and Luh's control schemes, the hand passes approximately through the presecribed points with certain error. Milenkovic [14] proposed another path control scheme using a special interpolation technique in which the hand can be driven continuously and smoothly to pass exactly through those prescribed points. He also pointed out that real-time processing for a continuous path control is important due to the fact that far more control data will be generated and the necessary processing rate is higher than that called for in point-to-point operation. The "inverse problem", is encountered whenever converting position, orientation, velocity, acceleration, or force in hand coordinates into local joint coordinates, and was a bottle-neck for real-time control of robots. In the past five years, many investigations were conducted by Paul, Luh, Walker, Vokubratovic, Hollerbach [8], and Silver [25] on the efficiency of solving the inverse problem. Finally, Silver concluded that the method (Newton-Euler formulation) presented by Luh, Walker and Paul [12], in practice, is the most efficient. Thus far, there seems to be little difficulty in guiding a

robot hand to move along a given, or preplanned path in real time. Furthermore, Luh [13] and Vokubratovic [31], respectively, presented the methodology of finding the optimal velocity distribution along a given path. Luh applied the modified "method of approximate programming (MAP)" to obtain a time schedule of velocity and accleration along the path such that the traveling time is minimal. On the other hand, Vokubratovic indicated that due to the complexity and nonlinear dynamics of a robot system it is difficult to apply classical optimal theory to solve optimization problems for a robot. Hence, "Dynamic Programming" was applied in his algorithm to determine the optimal velocity distribution along a given path for a 3R robot, under the criterion of minimal energy consumption.

Presently, as we can see, from the foregoing review, both the theory and practice have been well developed for the problem of preplanned path control. In other words, as long as the user specifies the traveling path of the hand of any specific robot with no more than 6 DOF, there is little difficulty to drive the hand to move along the specified path. Besides, certain optimal criteria, such as the minimal traveling time or the minimal energy consumption, can be imposed onto the motion of the hand. Consequently, the next interesting question is: for given initial and final locations of the robot in a free working space (in which no obstacle exists), can we find an optimal trajectory as our preplanned path? This study attempts to conduct a preliminary investigation on this problem.

Before answering the above question, a certain criterion of performance index should be selected in advance. From the standpoint of the efficiency of manipulator use, two aspects can be distinguished:

augmentation of work speed and diminishing of energy consumption. Therefore, either of the following criteria can be considered as the performance index for the optimal trajectory of manipulators:

(1) minimum traveling time,

(2) minimum energy consumption,

(3) mixed criteria.

In this study, the criterion of minimum energy consumption is used as the performance index during the optimization procedure.

Thus far, three different expressions, listed below, are widely adapted to represent the energy dissipated by a robot or a mechanical manipulator. The first and the second expressions stand for the energy and power, respectively, which are very straightforward and common expressions. The third expression was used by Chow [6] in his research on the human locomotion to account for the fact that energy can be consumed even when no motion is executed in a human-limb. In his research, the quadratic criterion ($\alpha = 2$) is used. In this study, the second and third expressions are used as the performance index in the illustrative examples.

(1) $\int M d\theta$

(2) $\int M d\dot{\theta}$

(3) $\int M^{\alpha} dt$

In above expressions M is the generalized force, θ is the generalized displacement and t is time.

The configuration of most common industrial robots can be kinematically decomposed into two sub-configurations: "regional structure" and "orientational structure". "Regional structure" usually refers to the first three joints and dominates the position of the end-effector

and the energy dissipated during the motion of the robot. "Orientational structure" refers to the last three joints and dominates the orientation of the end-effector, but has minor effect on the energy consumption.

As a matter of fact, all the energy consumed by the robot is supplied by the actuators of the robot. It is very obvious that the energy consumption is a function of the characteristics of the actuators used in the robot system. Actuators are assumed to be ideal in this study. We assume that there is no energy loss during transmission from the power source of the actuator to the robot system. However, the viscous effect is considered during the modeling of actuators in this study.

During the procedure of optimization, particular constraints as

(1) reachability: the geometrical condition that confines the manipulator's ability of reaching some specific position or orientation,

(2) durability: the condition that the stresses in the manipulator members do not exceed certain values,

(3) driving ability of actuators: the condition that the driving forces or torques do not exceed the values which can be generated by the actuators.

(4) stability: the condition that sustains the stability of a robot, especially for the robots having the ability of locomotion.

may be imposed onto the optimization. The constraints of (1) and (3) are discussed in the sense of formulation of performance index and the solution procedure, however, in the illustrative example we consider only the unconstrainted cases.

1.2 Proposed Research

The overall aim of this research is to develop a methodology of obtaining the optimal trajectory for an end-effector of any n DOF manipulator which moves in a free space. Minimum energy consumption is used as the performance index during the optimization procedure. Two constraints, reachability and driving ability of actuators, are considered along with the optimization procedure.

A brief review and comparison is made for the methods of optimizing a functional, such as the variational calculus, Pontryagin's maximum principle, Bellman's principle of optimality. A general methodology based on these methods is developed for the optimal trajectory of any n DOF robot.

The most popular 3R robot as shown in Figure 2 is illustrated to demonstrate the methodology developed. Two forms of performance index are used and discussed in the illustrative example. It is intended to develop an algorithm and an efficient computer program coded in FORTRAN for the most popular 3R robot to find the path of the end-effector in the fixed base coordinate frames in terms of a series of discrete points, and determining the joint angles and actuating forces, or torques, in local joint coordinate frame, with the initial and the final displacements of each joint as inputs. SAS/GRAPH is applied to generate a 3-dimensional plot on an IBM 3279 graphic terminal or Tektronix 4662 digital plotter such that the optimal trajectory is visible to the user.

The main purpose of this study is to conduct a preliminary investigation of the problem of optimal trajectory of a robot, and hopefully to provide some tools and fundamentals for future investigations. From

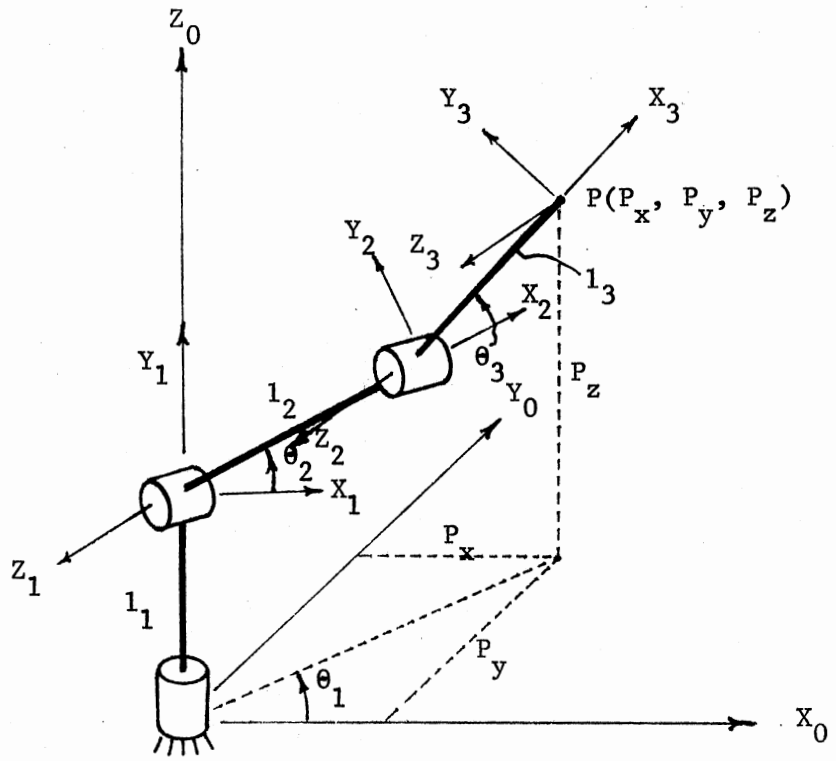

Figure 2. The Most Popular 3R Robot

the viewpoint of practice, the results of this study probably will be helpful to the industry in the sense of energy conservation, especially in the applications of heavy-duty pick/place type of robots.

1.3 Organization

Some background material has been reviewed briefly and the scope of study has been declared in present chapter. In Chapter II, the affine transformation is introduced and some useful properties are derived. The Denavit and Hartenberg's method is applied in conjunction with the Newton-Euler formulation to derive the explicit form of the driving torques and forces for any n DOF robot. The complete equations of motion in addition to the viscous effect are involved in our modeling.

Chapter III explains the fundamental theories of variational calculus, Pontryagin's maximum principle, Bellman's Principle of Optimality (Dynamic Programming) and Hooke and Jeeves pattern search. Two representative examples are illustrated to expose the spirit of the principle of optimality. The maximum principle is well interpreted in terms of two important rules of itself. A general algorithm based on above theories and the pattern search is proposed to determine the optimal trajectory of any n DOF robot.

In Chapter IV, the most popular 3R robot is illustrated to demonstrate the proposed algorithm. Two kinds of expressions are used as the performance index in the example, and three cases are investigated for this particular configuration of robot.

The final chapter summarizes what has been presented, discusses the possible extention of the proposed algorithm, and makes suggestions for future study and research.

CHAPTER II

FORMULATION OF PERFORMANCE INDEX FOR AN N DOF ROBOT

In deriving the general formulation of performance index, the first problem that we will encounter is how to obtain the explicit form of the driving force and toques at each joint. Several approaches, such as Newton-Euler method [12], Lagrange's method [8] and the quaternion method [18], have been investigated in the past few years. Among these approaches the Newton-Euler method is proven to be the most efficient in the sense of practical computation. However, we like to emphasize that the advantage of Newton-Euler metod is available only when applied in conjunction with the 3 x 3 rotation matrix instead of the 4 x 4 transformation matrix.

In this chapter, the affine transformation [1] is introduced to express the general relationship between two coordinate frames, and some important properties of the affine transformation are derived. Denvait and Hartenberg's method (DHM) are applied to represent the geometry of a link in space. A general formulation of driving toques is derived using the method of moving coordinates, Newton-Euler equation and 3 x 3 rotation matrix.

A notation $\{{}^{r}_{m}v\}$ or ${}^{r}_{m}\underline{v}$ will be used, throughout this study, to represent a vector v which is expressed in frame r but measured from the origin of frame m. Sometimes m may be omitted if frame m is the

base frame, i.e., m = 0. By using this notation, the kinematic relationships of moving coordinate systems may be expressed more concisely and precisely. For an easy understanding, we will use the most popular 3R robot as an illustrative example at proper situation.

2.1 Affine Transformation

An affine transformation is a general method describing the relationship between two coordinate systems related by a change of coordinates involving a translation of the origin and a rotation of axes. Let x_1, y_1, z_1 and x_2, y_2, z_2 be two sets of coordinates related by an affine transformation as shon in Figure 3.

The position vector of any point p, referred to coordinates 1 and 2, can be expressed as

$$\{{}^1_1p\} = \{{}^1_1r_{12}\} + [{}^1Q_2] \{{}^2_2p\} \tag{2.1}$$

where $\{{}^1_1r_{12}\}$ is the translation vector from O_1 to O_2 and $[{}^1Q_2]$ is the rotation matrix from frame 2 to frame 1, both are expressed with respect to frame 1 while $\{{}^2_2p\}$ is the position vector of point p measured in frame 2 and expressed in frame 2. Equation (2.1) indicates the general form of an affine transformation. Symbolically, the transformation of Equation (2.1) canbe written as

$$\{{}^1_2p\} = [{}^1T_2] \{{}^2_2p\} \tag{2.2}$$

Similarly, $[{}^2T_3]$ is given through vector $\{{}^2_2r_{23}\}$ and matrix $[{}^2Q_3]$ as

$$\{{}^2_2p\} = \{{}^2_2r_{23}\} + [{}^2Q_3] \{{}^3_3p\} \tag{2.3}$$

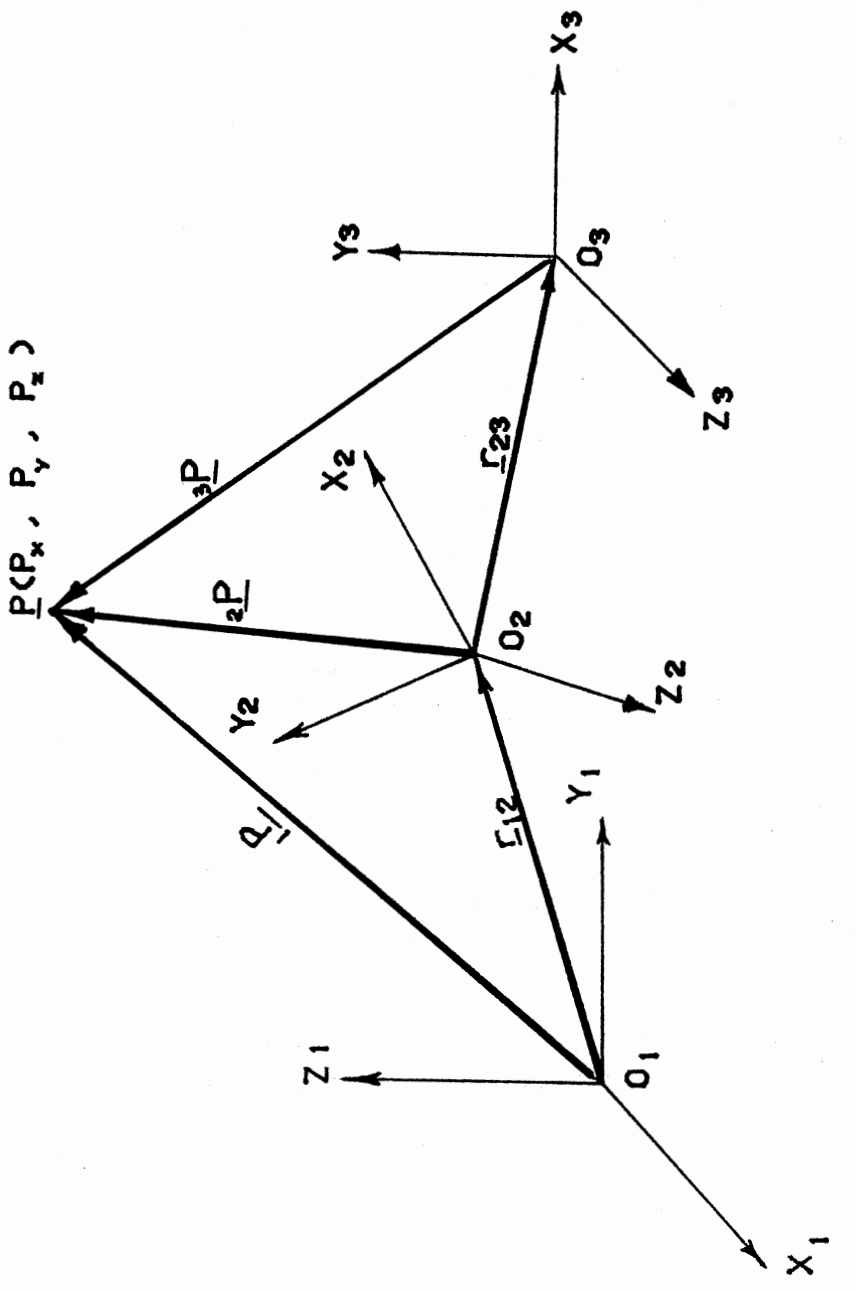

Figure 3. The Affine Transformation

and $[{}^1T_3]$ is given through $\{{}^1_1r_{13}\}$ and $[{}^1Q_3]$, correspondingly, as

$$\{{}^1_1p\} = \{{}^1_1r_{13}\} + [{}^1Q_3]\ \{{}^3_3p\} \tag{2.4}$$

Substitution of Equation (2.3) into (2.1) yields

$$\begin{aligned}\{{}^1_1p\} &= \{{}^1_1r_{12}\} + [{}^1Q_2](\{{}^2_2r_{23}\} + [{}^2Q_3]\ \{{}^3_3p\}\ ) \\ &= \{{}^1_1r_{12}\} + [{}^1Q_2]\ \{{}^2_2r_{23}\} + [{}^1Q_2]\ [{}^2Q_3]\ \{{}^3_3p\}\end{aligned} \tag{2.5}$$

By comparing Equation (2.5) with (2.4) we obtain

$$\{{}^1_1r_{13}\} = \{{}^1_1r_{12}\} + [{}^1Q_2]\ \{{}^2_2r_{23}\} \tag{2.6}$$

which, alternatively, can be written as

$$\{{}^1_1r_{13}\} = \{{}^1_1r_{12}\} + \{{}^1_2r_{23}\} \tag{2.7}$$

Equation (2.7) is in agreement with the geometrical meaning, that is to say, the vector connecting the origin O_1 to the origin O_3 equals to the sum of that connecting O_1 to O_2 plus that connecting O_2 to O_3. Also, from Equation (2.5) we have

$$[{}^1Q_3] = [{}^1Q_2]\ [{}^2Q_3] \tag{2.8}$$

thereby showing that the composition of two affine transformations is also affine.

Let the "identity affine transformation" $[{}^iT_i]$ be defined as

$$\{{}^i_ix\} = [{}^iT_i]\ \{{}^i_ix\} \tag{2.9}$$

i.e., $[{}^iT_i]$ is the coordinate transformation from coordinates i into themselves. Clearly, its translation vector is the zero vector and

the rotation matrix is the identity matrix.

Now, let's consider the "inverse transformation" $[{}^{i}T_{j}]^{-1}$ such that

$$[{}^{i}T_{j}]^{-1}\,[{}^{i}T_{j}]\,\{{}^{j}_{j}x\} = [{}^{j}T_{i}][{}^{j}T_{i}]^{-1}\,\{{}^{j}_{j}x\} = \{{}^{j}_{j}x\}\quad [I]\{{}^{j}_{j}x\} \tag{2.10}$$

where

$$[{}^{i}T_{j}]\,\{{}^{j}_{j}x\} = \{{}^{i}_{i}r_{ij}\} + [{}^{i}Q_{j}]\,\{{}^{j}_{j}x\} \tag{2.11}$$

and

$$[{}^{i}T_{j}]^{-1}\,\{{}^{i}_{i}x\} = [{}^{j}T_{i}]\,\{{}^{i}_{i}x\} = \{{}^{j}_{j}r_{ji}\} + [{}^{j}Q_{i}]\,\{{}^{i}_{i}x\} \tag{2.12}$$

Thus,

$$\begin{aligned}[{}^{i}T_{j}]^{-1}\,[{}^{i}T_{j}]\,\{{}^{j}_{j}x\} &= \{{}^{j}_{j}r_{ji}\} + [{}^{j}Q_{i}](\{{}^{i}_{i}r_{ij}\} + [{}^{i}Q_{j}]\,\{{}^{j}_{j}x\}\,)\\ &= \{{}^{j}_{j}r_{ji}\} + [{}^{j}Q_{i}]\,\{{}^{i}_{i}r_{ij}\}\\ &\quad + [{}^{j}Q_{i}][{}^{i}Q_{j}]\,\{{}^{j}_{j}x\}\end{aligned} \tag{2.13}$$

Substituting Equation (2.13) into (2.10) we obtain

$$\{{}^{j}_{j}r_{ji}\} + [{}^{j}Q_{i}]\{{}^{i}_{i}r_{ij}\} = \underline{0} \tag{2.14}$$

and

$$[{}^{j}Q_{i}][{}^{i}Q_{j}] = [I] \tag{2.15}$$

Hence,

$$[{}^{j}Q_{i}] = [{}^{i}Q_{j}]^{-1} = [{}^{i}Q_{j}]^{T} \tag{2.16}$$

and

$$\{{}^{j}_{j}r_{ji}\} = -[{}^{i}Q_{j}]^{T}\ \{{}^{i}_{i}r_{ij}\} \tag{2.17}$$

Note that the rotation matrix is orthonormal. This property is very useful while dealing with inverse problems.

Next, a general composition law for n transformations is derived. Assuming that an expression similar to Equation (2.7) and (2.8) hold for k transformations, it will be shown that they hold also for k + 1, thereby obtaining general relationships by induction. Thus,

$$\{{}^{1}_{1}r_{1k}\} = \{{}^{1}_{1}r_{12}\} + \{{}^{1}_{2}r_{23}\} + \ldots\ldots + \{{}^{1}_{k-1}r_{k-1,k}\} \tag{2.18}$$

$$[{}^{1}Q_{k}] = [{}^{1}Q_{2}][{}^{2}Q_{3}] \ldots\ldots [{}^{k-1}Q_{k}] \tag{2.19}$$

then,

$$\{{}^{1}_{1}r_{1,k+1}\} = \{{}^{1}_{1}r_{1k}\} + \{{}^{1}_{k}r_{k,k+1}\} \tag{2.20}$$

and

$$[{}^{1}Q_{1,k+1}] = [{}^{1}Q_{k,k+1}][{}^{1}Q_{1k}] \tag{2.21}$$

Introducing a similarity transformation to refer ${}^{k}Q_{k,k+1}$ to k-coordinates,

$$\begin{aligned}[{}^{1}Q_{k+1}] &= ([{}^{1}Q_{k}][{}^{k}Q_{k,k+1}][{}^{1}Q_{k}]^{T})[{}^{1}Q_{k}] \\ &= [{}^{1}Q_{k}][{}^{k}Q_{k+1}]\end{aligned} \tag{2.22}$$

hence, in general

$$\{{}^{1}_{1}r_{1n}\} = \sum_{i=1}^{n-1} \{{}^{1}_{i}r_{i,i+1}\} \tag{2.23}$$

$$[{}^{1}Q_{n}] = [{}^{1}Q_{2}][{}^{2}Q_{3}] \ldots\ldots [{}^{n-1}Q_{n}] \tag{2.24}$$

which are very useful while formulating the kinematic equations for n DOF robots. Usually, however, $\{{}^{1}_{i}r_{i,i+1}\}$ will be more readily referred to coordinate system i instead of 1, and the transformation to system 1 is easily performed as

$$\{{}^{i}_{i}f_{i,i+1}\} = [{}^{i}Q_{1}]\ \{{}^{1}_{i}r_{i,i+1}\} \tag{2.25}$$

where $[{}^{i}Q_{1}]$ can be obtained from Equation (2.24) with n = i.

A practical application of affine transformation is the method of Denavit and Hartenburg (MDH), which is applied widely in the analysis and synthesis of rigid body motion. Essentially, MDH is based on the loop-closure equation

$$[{}^{1}T_{n}][{}^{n}T_{n-1}] \ldots\ldots [{}^{3}T_{2}][{}^{2}T_{1}] = \underline{I} \tag{2.26}$$

or, equivalently,

$$\{{}^{1}_{1}r_{12}\} + \{{}^{1}_{2}r_{23}\} + \ldots\ldots + \{{}^{1}_{n-1}r_{n-1,n}\} + \{{}^{1}_{n}r_{n}\} = \underline{0} \tag{2.27}$$

together with

$$[{}^{1}Q_{n}][{}^{n}Q_{n-1}] \ldots\ldots [{}^{3}Q_{2}][{}^{2}Q_{1}] = \underline{I} \tag{2.28}$$

where

$$[{}^{i}T_{i+1}] = \left[\begin{array}{c|c} {}^{i}\underline{Q}_{i+1} & {}^{i}_{i}\underline{r}_{i,i+1} \\ \hline 0 & 1 \end{array}\right] \tag{2.29}$$

A general link in the space can be represented very easily and clearly in terms of Denavit and Hartenberg's link parameters: a_i, θ_i, s_i, and

α_i as shown in Figure 4; where

i : number of particular joint or link

z_i : characteristic axis of motion for the joint involved

x_{i+1} : axis formed by the common normal directed from z_i to z_{i+1}.

y_i : axis implicitly determined by x_i and z_i to form a right-handed Cartesian coordinate system $x_i y_i z_i$.

a_i : link length measured positive along x_{i+1}.

θ_i : angle between x_i and x_{i+1}; measured positive counter-clockwise about z_i.

s_i : offset distance measured along z_i.

α_i : twist angle defining the relative orientation of z_i and z_{i+1}; measured positive counter-clockwise about x_{i+1}.

The link parameters of the most popular 3R robot are listed in Table I, and the transformation matrix then can be written as

$$[{}^{i}T_{i+1}] = \begin{bmatrix} \text{Cos}\theta_i & -\text{Sin}\theta_i\text{Cos}\alpha_i & \text{Sin}\theta_i\text{Sin}\alpha_i & a_i\text{Cos}\theta_i \\ \text{Sin}\theta_i & \text{Cos}\theta_i\text{Cos}\alpha_i & -\text{Cos}\theta_i\text{Sin}\alpha_i & a_i\text{Sin}\theta_i \\ 0 & \text{Sin}\theta_i & \text{Cos}\alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.30}$$

Comparing Equation (2.30) with (2.29), we have explicit forms of rotation matrix $[{}^{i}Q_{i+1}]$ and translation vector $\{{}^{i}_{i}r_{i,i+1}\}$, in terms of MDH. Besides, the sum total of n transformations can be expressed as

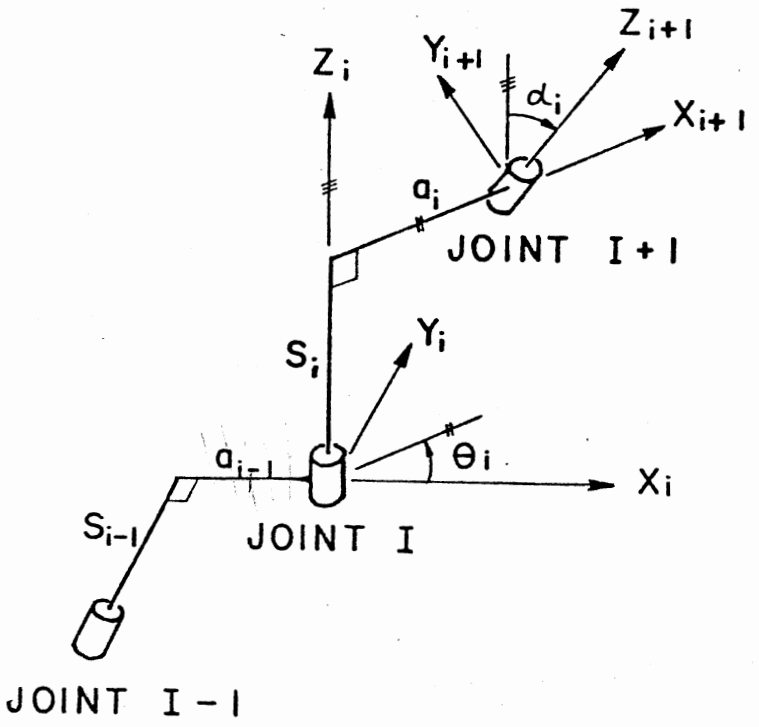

Figure 4. Denavit and Hartenberg's Link Parameters.

TABLE I. LINK PARAMETERS FOR THE MOST POPULAR 3R ROBOT

i	a_i	s_i	α_i	θ_i
1	0	ℓ_1	90°	θ_1
2	ℓ_2	0	0°	θ_2
3	ℓ_3	0	0°	θ_3

$$[{}^{1}T_{n}] = [{}^{1}T_{2}][{}^{2}T_{3}] \ . \ . \ . \ . \ . \ [{}^{n-1}T_{n}] \qquad (2.31)$$

2.2 A Systematic Approach to the Explicit Form of Driving Toques and Forces

The general equation of motion for an n DOF robot can be expressed as

$$\underline{J}(\underline{q})\ddot{\underline{q}} + \underline{V}\dot{\underline{q}} + \underline{f}(\dot{\underline{q}}_i\dot{\underline{q}}_j,\underline{q};\ i, j=1, 2, \ . \ . \ . \ . \ . \ n) + \underline{g}(\underline{q}) = \underline{\tau} \qquad (2.32)$$

where

$\underline{q}$: n x 1 vector defining the joint displacements

$\underline{J}(\underline{q})$: n x n inertia matrix

$\underline{V}$: n x n viscous friction matrix

$f(q_i q_j, q)$: n x 1 vector corresponding to Coriolis and centrifugal effects

$\underline{g}(\underline{g})$: n x 1 vector defining the gravity terms

$\underline{\tau}$: n x 1 vector of input generalized forces

A systematic approach for deriving the explicit form of the generalized driving forces is stated as below. The same approach can also be implemented as a computer program for the kinematic analysis for any n DOF robot.

The explicit form of driving forces or toques can be obtained by the successive transformation of velocities and accelerations from the base of the manipulator out to the end-effector, link by link, using the relationship of moving coordinate systems. Forces and torques are then transformed backwardly from the end-effector to the base.

Considering the coordinate systems as shown in Figure 5, the base frame (x_0, y_0, z_0) attached to link 0, the frame (x_i, y_i, z_i) attached to link i, and the frame $(x_{i-1}, y_{i-1}, z_{i-1})$ attached to link i-1, thus the frames move together with the links. Let ${}^{r}\underline{w}_i$ denote the angular velocity of link i, measured in base frame and represented in frame r which is attached to link i. From the general applied mechanics [9], we have following recursive relationships to evaluate the angular velocity of link i with respect to the base frame.

$$ {}^{r-1}\underline{\omega}_i = \begin{cases} {}^{r-1}\underline{\omega}_{i-1} + \underline{z}\dot{q}_i, & \text{for revolute pairs} \\ {}^{r-1}\underline{\omega}_{i-1}, & \text{for prismatic pairs} \end{cases} \tag{2.33} $$

where

$$ \underline{z} = (\,0,\ 0,\ 1)^T $$

$$ {}^{r}\underline{\omega}_i = [{}^{r}Q_{r-1}]\ {}^{r-1}\underline{\omega}_i \tag{2.34} $$

Based on the above equations, the following procedures can be written as the scheme of derivation or computation.

Step 1: set i = 1 and r = 1.

Step 2: use Equation (2.33) to obtain ${}^{r-1}\underline{\omega}_i$.

Step 3: use Equation (2.34) to obtain ${}^{r}\underline{\omega}_i$.

Step 4: if i = n then stop.

Step 5: set i = i + 1 and r = r + 1, go to Step 1.

The angular velocity of each link for the most 3R robot can be obtained

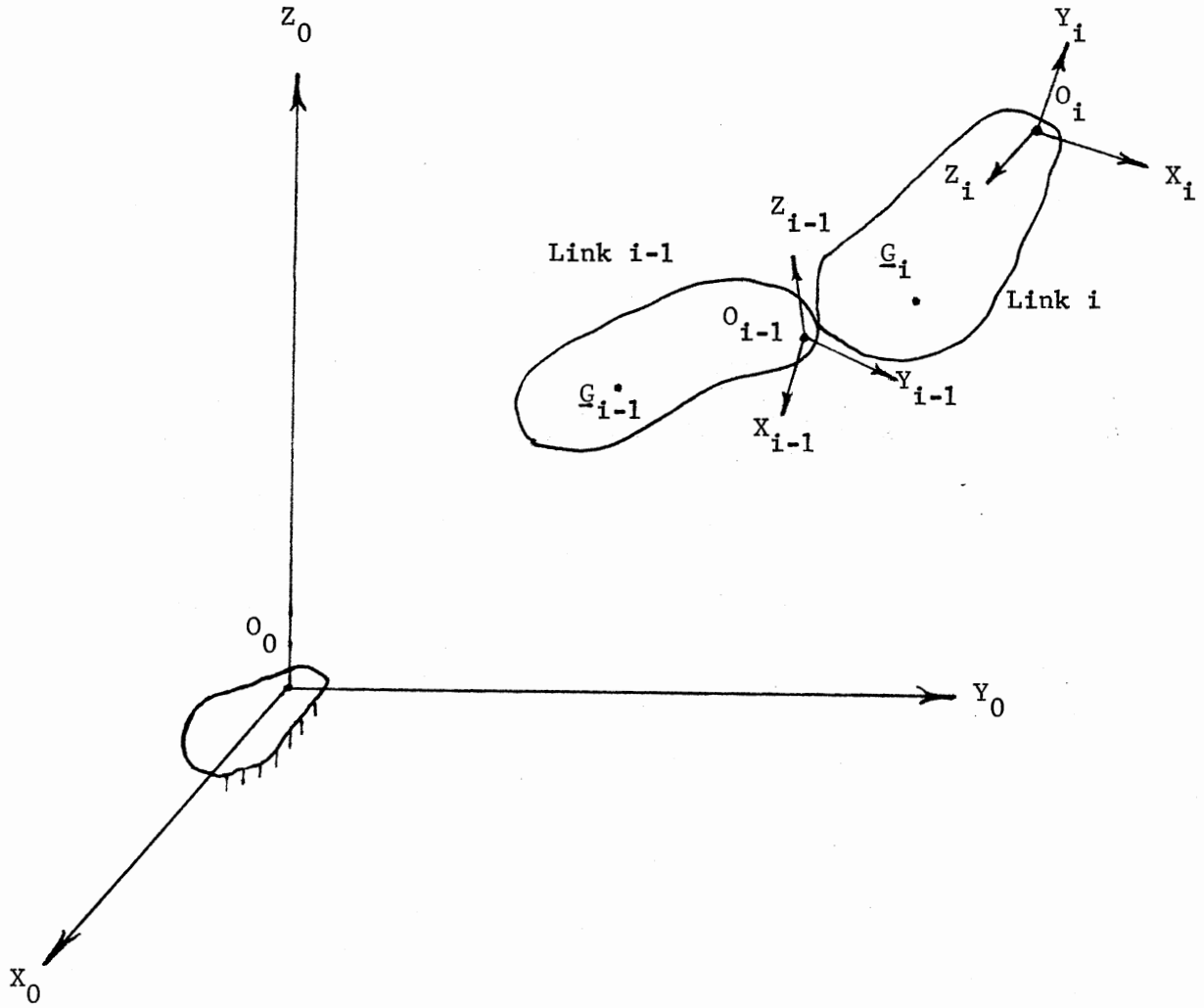

Figure 5. Motion of a Rigid-Body w.r.t a Moving Coordinate System

by following the above procedures and is listed as below.

$$ {}^{0}\underline{\omega}_{1} = \underline{z}_{0}\dot{q} = \begin{Bmatrix} 0 \\ 0 \\ \dot{\theta}_{1} \end{Bmatrix} \tag{2.35} $$

$$ {}^{1}\underline{\omega}_{1} = [{}^{1}Q_{0}]\,{}^{0}\underline{\omega}_{1} = \begin{Bmatrix} 0 \\ \dot{\theta}_{1} \\ 0 \end{Bmatrix} \tag{2.36} $$

$$ {}^{1}\underline{\omega}_{2} = {}^{1}\underline{\omega}_{1} + \underline{z}\dot{q}_{2} = \begin{Bmatrix} 0 \\ \dot{\theta}_{1} \\ \dot{\theta}_{2} \end{Bmatrix} \tag{2.37} $$

$$ {}^{2}\underline{\omega}_{2} = [{}^{2}Q_{1}]\,{}^{1}\underline{\omega}_{2} = \begin{Bmatrix} \dot{\theta}_{1}\, S_{2} \\ \dot{\theta}_{1}\, C_{2} \\ \dot{\theta}_{2} \end{Bmatrix} \tag{2.38} $$

$$ {}^{2}\underline{\omega}_{3} = {}^{2}\underline{\omega}_{2} + \underline{z}\dot{q}_{3} = \begin{Bmatrix} \dot{\theta}_{1} & S_{2} \\ \dot{\theta}_{1} & C_{2} \\ \dot{\theta}_{2} + \dot{\theta}_{3} \end{Bmatrix} \tag{2.39} $$

$$ {}^{3}\underline{\omega}_{3} = [{}^{3}Q_{2}]\,{}^{2}\underline{\omega}_{3} = \begin{Bmatrix} \dot{\theta}_{1} & S_{23} \\ \dot{\theta}_{1} & C_{23} \\ \dot{\theta}_{2} + \dot{\theta}_{3} \end{Bmatrix} \tag{2.40} $$

where S_i, C_i denotes $\mathrm{Sin}\theta_i$, $\mathrm{Cos}\theta_i$, respectively, and S_{ij}, C_{ij} denotes $\mathrm{Sin}(\theta_i + \theta_j)$, $\mathrm{Cos}\,(\theta_i + \theta_j)$, respectively, and so on. Besides, θ_i, $\dot{\theta}_i$ and $\ddot{\theta}_i$ denote the angular displacement, velocity and acceleration,

respectively, of link in terms of local coordinates.

Similarly, let ${}^{r}\dot{\underline{\omega}}_i$ denote the angular acceleration of link i, measured in the base frame and represented in frame r. By applying the similar procedures as used in the derivation of angular velocity and following recursive relationships we can obtain the angular acceleration of each link.

$$ {}^{r-1}\dot{\underline{\omega}}_i = \begin{cases} {}^{r-1}\dot{\underline{\omega}}_{i-1} + \ddot{\underline{zq}}_i + {}^{r-1}\underline{\omega}_{i-1} \times (\dot{\underline{zq}}_i) & \text{for revolute pairs} \\ {}^{r-1}\dot{\underline{\omega}}_{i-1} & \text{, for prismatic pairs} \end{cases} \tag{2.41} $$

$$ {}^{r}\dot{\underline{\omega}}_i = [{}^{r}Q_{r-1}]\,{}^{r-1}\dot{\underline{\omega}}_i \tag{2.42} $$

Followings are the angular accelerations of each link for the most popular 3R robot.

$$ {}^{0}\dot{\underline{\omega}}_1 = \ddot{\underline{zq}}_1 = \begin{Bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{Bmatrix} \tag{2.43} $$

$$ {}^{1}\dot{\underline{\omega}}_1 = [{}^{1}Q_0]\,{}^{0}\dot{\underline{\omega}}_1 = \begin{Bmatrix} 0 \\ \ddot{\theta}_1 \\ 0 \end{Bmatrix} \tag{2.44} $$

$$ {}^{1}\dot{\underline{\omega}}_2 = {}^{1}\dot{\underline{\omega}}_1 + \ddot{\underline{zq}}_2 + {}^{1}\underline{\omega}_1 \times (\dot{\underline{zq}}_2) = \begin{Bmatrix} \dot{\theta}_1\,\dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} \tag{2.45} $$

$$ {}^{2}\dot{\underline{\omega}}_2 = [{}^{2}Q_1]^{1}\ \dot{\underline{\omega}}_2 = \begin{Bmatrix} \dot{\theta}_1\ \dot{\theta}_2\ C_2 + \ddot{\theta}_1\ S_2 \\ -\dot{\theta}_1\ \dot{\theta}_2\ S_2 + \ddot{\theta}_1\ C_2 \\ \ddot{\theta}_2 \end{Bmatrix} \tag{2.46} $$

$$ {}^{2}\dot{\underline{\omega}}_3 = {}^{2}\dot{\underline{\omega}}_2 + \ddot{\underline{z}}\underline{q}_3 + {}^{2}\underline{\omega}_2 \times (\dot{\underline{z}}\underline{q}_3) = \begin{Bmatrix} \dot{\theta}_1\ C_2(\dot{\theta}_2 + \dot{\theta}_3) + \ddot{\theta}_1\ S_2 \\ -\dot{\theta}_1\ S_2(\dot{\theta}_2 + \dot{\theta}_3) + \ddot{\theta}_1\ C_2 \\ \ddot{\theta}_2 + \ddot{\theta}_3 \end{Bmatrix} \tag{2.47} $$

$$ {}^{3}\dot{\underline{\omega}}_3 = [{}^{3}Q_2]^{2}\dot{\underline{\omega}}_3 = \begin{Bmatrix} \dot{\theta}_1\ (\dot{\theta}_2 + \dot{\theta}_3)C_{23} + \ddot{\theta}_1 S_{23} \\ -\dot{\theta}_1\ (\dot{\theta}_2 + \dot{\theta}_3)S_{23} + \ddot{\theta}_1 C_{23} \\ \ddot{\theta}_2 + \ddot{\theta}_3 \end{Bmatrix} \tag{2.48} $$

The linear velocity of link i with respect to the base frame can also be obtained in the similar manner by using the following recursive relationships.

$$ {}^{r-1}\underline{v}_i = \begin{cases} {}^{r-1}\underline{v}_{i-1} + {}^{r-1}\underline{\omega}_i \times ({}^{r-1}_{i-1}\underline{O}_i), & \text{for revolute pairs} \\ {}^{r-1}\underline{v}_{i-1} + {}^{r-1}\underline{\omega}_i \times {}^{r-1}_{i-1}\underline{O}_i + \dot{z}q_i, & \text{for prismatic pairs} \end{cases} \tag{2.49} $$

$$ {}^{r}\underline{v}_i = [{}^{r}Q_{r-1}]\ {}^{r-1}\underline{v}_i \tag{2.50} $$

$$ {}^{r}\underline{v}_i^{*} = {}^{r}\underline{v}_i + {}^{r}\underline{\omega}_i \times {}^{r}_{i}\underline{G}_i \tag{2.51} $$

where ${}^{r}\underline{v}_i$ is the linear velocity at the tip of link i while ${}^{r}\underline{v}_i^*$ is at the center of mass, and $\underline{G}_i$ is the position vector of the center of mass of link i while G_i^* is a scalar describing the center of mass of link i measured from the origin of frame i-1. Followings are the linear velocities of each link for the most popular 3R robot.

$$ {}^{1}\underline{v}_1 = [{}^{1}Q_0]({}^{0}\underline{\omega}_1 \times {}^{0}_{0}\underline{O}_1) = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{2.52} $$

$$ {}^{1}\underline{v}_1^* = {}^{1}\underline{v}_1 + {}^{1}\underline{\omega}_1 \times {}^{1}_{1}\underline{G}_1 = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{2.53} $$

$$ {}^{2}\underline{v}_2 = [{}^{2}Q_1]\,({}^{1}\underline{v}_1 + {}^{1}\underline{\omega}_2 \times {}^{1}_{1}\underline{O}_2) = \begin{Bmatrix} 0 \\ \dot{\theta}_2\, \ell_2 \\ -\dot{\theta}_1\, \ell_2\, C_2 \end{Bmatrix} \tag{2.54} $$

$$ {}^{2}\underline{v}_2^* = {}^{2}\underline{v}_2 + {}^{2}\underline{\omega}_2 \times {}^{2}_{2}\underline{G}_2 = \begin{Bmatrix} 0 \\ \dot{\theta}_2\, G_2^* \\ -\dot{\theta}_1\, G_2^*\, C_2 \end{Bmatrix} \tag{2.55} $$

$$ {}^{3}\underline{v}_3 = [{}^{3}Q_2]({}^{2}\underline{v}_2 + {}^{2}\underline{\omega}_3 \times {}^{2}_{2}\underline{O}_3) = \begin{Bmatrix} \dot{\theta}_2\, \ell_2\, S_3 \\ \ell_3\,(\dot{\theta}_2 + \dot{\theta}_3) + \dot{\theta}_2\, \ell_2\, C_3 \\ -\dot{\theta}_1\ \ell_2\, C_2 - \dot{\theta}_1\, \ell_3\, C_{23} \end{Bmatrix} \tag{2.56} $$

$$ {}^{3}\underline{v}_{3}^{*} = {}^{3}\underline{v}_{3} + {}^{3}\underline{\omega}_{3} \times {}^{3}_{3}\underline{G}_{3} = \begin{Bmatrix} \dot{\theta}_2 \, \ell_2 \, S_3 \\ G_3^{*} \, (\dot{\theta}_2 + \dot{\theta}_3) + \dot{\theta}_2 \ell_2 \, C_3 \\ -\dot{\theta}_1 \, \ell_2 \, C_2 - \dot{\theta}_1 \, G_3^{*} \, C_{23} \end{Bmatrix} \tag{2.57} $$

The acceleration can also be obtained by using the following recursive relationships.

$$ {}^{r-1}\dot{\underline{v}}_i = \begin{cases} {}^{r-1}\dot{\underline{v}}_{i-1} + {}^{r-1}\dot{\underline{\omega}}_i \times ({}^{r-1}_{i-1}\underline{O}_i) \\ \quad + {}^{r-1}\underline{\omega}_i \times ({}^{r-1}\underline{\omega}_i \times {}^{r-1}_{i-1}\underline{O}_i), \text{ for revolute pairs} \\ {}^{r-1}\dot{\underline{v}}_{i-1} + {}^{r-1}\dot{\underline{\omega}}_i \times {}^{r-1}_{i-1}\underline{O}_i + \ddot{\underline{z}q}_i \\ \quad + {}^{r-1}\underline{\omega}_i \times ({}^{r-1}\underline{\omega}_i \times {}^{r-1}_{i-1}\underline{O}_i) \\ \quad + 2\,{}^{r-1}\underline{\omega}_i \times (\dot{\underline{z}q}_i), \text{ for prismatic pairs} \end{cases} \tag{2.58} $$

$$ {}^{r}\dot{\underline{v}}_i = [{}^{r}Q_{r-1}] \, {}^{r-1}\dot{\underline{v}}_i \tag{2.59} $$

$$ {}^{r}\dot{\underline{v}}_i^{*} = {}^{r}\dot{\underline{v}}_i + {}^{r}\underline{\omega}_i \times {}^{r}_{i}\underline{G}_i + {}^{r}\underline{\omega}_i \times ({}^{r}\underline{\omega}_i \times {}^{r}_{i}\underline{G}_i) \tag{2.60} $$

where ${}^{r}\dot{\underline{v}}_i$ is the linear acceleration of each link at the tip of link i while ${}^{r}\dot{\underline{v}}_i^{*}$ is at the center of mass. Followings are the expressions of linear acceleration of each link for the most popular 3R robot.

$$ {}^{1}\dot{\underline{v}}_i = [{}^{1}Q_0]({}^{0}\dot{\underline{v}}_0 + {}^{0}\dot{\underline{\omega}}_1 \times {}^{0}_{0}\underline{O}_1 + {}^{0}\underline{\omega}_1 \times ({}^{0}\underline{\omega}_1 \times {}^{0}_{0}\underline{O}_1)) = \begin{Bmatrix} 0 \\ g \\ 0 \end{Bmatrix} \tag{2.61} $$

$$ {}^{1}\dot{\underline{v}}^{*}_{1} = {}^{1}\dot{\underline{v}}_{1} + {}^{1}\dot{\underline{\omega}}_{1} \times {}^{1}\underline{G}_{1} + {}^{1}\underline{\omega}_{1} \times ({}^{1}\underline{\omega}_{1} \times {}^{1}\underline{G}_{1}) = \begin{Bmatrix} 0 \\ g \\ 0 \end{Bmatrix} \tag{2.62} $$

$$ {}^{2}\dot{\underline{v}}_{2} = [{}^{2}Q_{1}]({}^{1}\dot{\underline{v}}_{1} + {}^{1}\dot{\underline{\omega}}_{2} \times {}^{1}\underline{O}_{2} + {}^{1}\omega_{2} \times ({}^{1}\underline{\omega}_{2} \times {}^{1}\underline{O}_{2})) = $$

$$ \begin{Bmatrix} -\dot{\theta}_2^2 \ell_2 - \ell_2 C_2^2 \dot{\theta}_1^2 + g S_2 \\ \ddot{\theta}_2 \ell_2 + \dot{\theta}_1^2 \ell_2 g_2 S_2 + g C_2 \\ 2\dot{\theta}_1 \dot{\theta}_2 \ell_2 S_2 - \ddot{\theta}_1 \ell_2 C_2 \end{Bmatrix} \tag{2.63} $$

$$ {}^{2}\dot{\underline{v}}^{*}_{2} = {}^{2}\dot{\underline{v}}_{2} + {}^{2}\dot{\underline{\omega}}_{2} \times {}^{2}\underline{G}_{2} + {}^{2}\underline{\omega}_{2} \times ({}^{2}\underline{\omega}_{2} \times {}^{2}\underline{G}_{2}) = $$

$$ \begin{Bmatrix} -\dot{\theta}_2^2 G_2^* - \dot{\theta}_1^2 C_2^2 G_2^* + g S_2 \\ \ddot{\theta}_2 G_2^* + \dot{\theta}_1^2 S_2 C_2 G_2^* + g C_2 \\ 2\dot{\theta}_1 \dot{\theta}_2 S_2 G_2^* - \ddot{\theta}_1 C_2 G_2^* \end{Bmatrix} \tag{2.64} $$

$$ {}^{3}\dot{\underline{v}}_{3} = [{}^{3}Q_{2}]({}^{2}\dot{\underline{v}}_{2} + {}^{2}\dot{\underline{\omega}}_{3} \times {}^{2}\underline{O}_{3} + {}^{2}\underline{\omega}_{3} \times ({}^{2}\underline{\omega}_{3} \times {}^{2}\underline{O}_{3})) = $$

$$ \begin{Bmatrix} -\dot{\theta}_1^2 \ell_2 C_2 C_{23} - \dot{\theta}_2^2 \ell_2 C_3 + \ddot{\theta}_2 \ell_2 S_3 + g S_{23} - \dot{\theta}_1^2 \ell_3 C_{23}^2 - (\dot{\theta}_1 + \dot{\theta}_3)^2 \ell_3 \\ \dot{\theta}_1^2 \ell_2 C_2 S_{23} + \dot{\theta}_2^2 \ell_2 S_3 + \ddot{\theta}_2 \ell_2 C_3 + g C_{23} + (\ddot{\theta}_2 + \ddot{\theta}_3) \ell_3 + \dot{\theta}_1^2 \ell_3 S_{23} C_{23} \\ 2\dot{\theta}_1 \dot{\theta}_2 \ell_2 S_2 - \ddot{\theta}_1 \ell_2 C_2 + 2\dot{\theta}_1 (\dot{\theta}_2 + \dot{\theta}_3) \ell_3 S_{23} - \ddot{\theta}_1 \ell_3 C_{23} \end{Bmatrix} \tag{2.65} $$

$$ {}^{3}\dot{\underline{v}}^{*}_{3} = {}^{3}\dot{\underline{v}}_{3} + {}^{3}\underline{\omega}_{3} \times {}^{3}\underline{G}_{3} \times {}^{3}\underline{\omega}_{3} \times ({}^{3}\underline{\omega}_{3} \times {}^{3}\underline{G}_{3}) = $$

$$ \begin{Bmatrix} -\dot{\theta}_1^2 \ell_2 C_2 C_{23} - \dot{\theta}_2^2 \ell_2 C_3 + \ddot{\theta}_2 \ell_2 S_3 + g S_{23} - \dot{\theta}_1^2 G_3^* C_{23}^2 - (\dot{\theta}_2 + \dot{\theta}_3)^2 G_3^* \\ \dot{\theta}_1^2 \ell_2 C_2 S_{23} + \dot{\theta}_2^2 \ell_2 S_3 + \ddot{\theta}_2 \ell_2 C_3 + g C_{23} + \dot{\theta}_1^2 G_3^* S_{23} C_{23} + (\ddot{\theta}_2 + \ddot{\theta}_3) G_3^* \\ 2\dot{\theta}_1 \dot{\theta}_2 \ell_2 S_2 - \ddot{\theta}_1 \ell_2 C_2 + 2\dot{\theta}_1 (\dot{\theta}_2 + \dot{\theta}_3) G_3^* S_{23} - \ddot{\theta}_1 G_3^* C_{23} \end{Bmatrix} \tag{2.66} $$

where $g = 9.8\ m/sec^2$

Now, we have obtained the angular velocity, accleration and the linear velocity, acceleration of each link. Following is the procedure to obtain the expression of the driving force or torque at each joint.

With reference to Figure 6, by applying D'Alembert's principle and Newton-Euler formulation to link i of the manipulator we can obtain

$$ \underline{f}_i = \underline{f}_{i+1} + \underline{F}_i \tag{2.67} $$

$$ \underline{m}_i = \underline{m}_{i+1} + (s_i \underline{z}_i + a_i \underline{x}_{i+1}) \times \underline{f}_{i+1} + \underline{G}_i \times \underline{F}_i + \underline{M}_i \tag{2.68} $$

where

$\underline{F}_i$: body force of link i with respect to the coordinate frame i+1

$\underline{G}_i$: position vector of the center of mass of link i with respect to the coordinate frame i+1

$\underline{f}_i$: force exerted on link i at joint i

$\underline{f}_{i+1}$: force exerted on link i at joint i+1

$\underline{m}_i$: moment exerted on link i at joint i

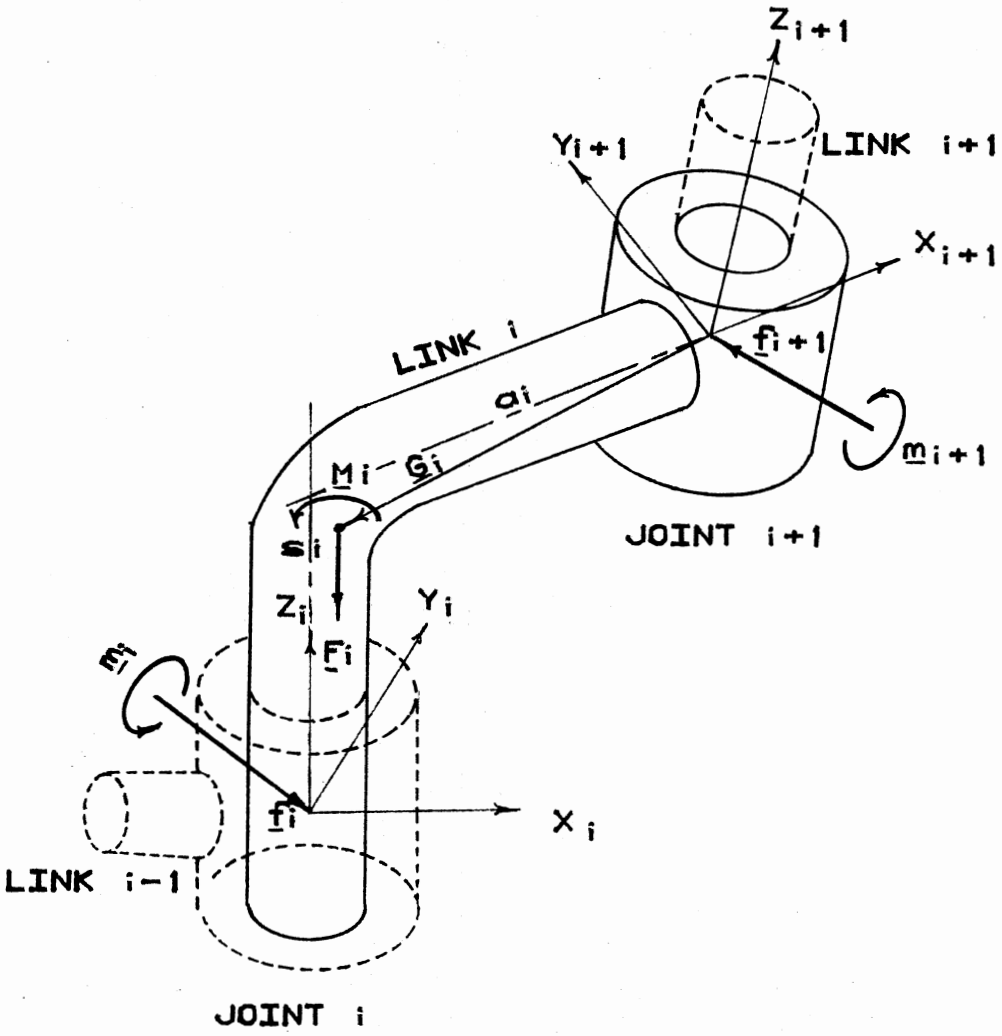

Figure 6. The Free-Body Diagram of Link i

$\underline{m}_{i+1}$: moment exerted on link i at joint i

$\underline{M}_i$: external moment applied to link i

In which $\underline{F}_i$ and $\underline{M}_i$ can be obtained from the Newton's and Euler's equations as following.

$$\underline{F}_i = W_i \underline{v}_i^* \tag{2.69}$$

$$\underline{M}_i = [J_i]\, \dot{\underline{\omega}}_i + \underline{\omega}_i \times ([J_i]\, \underline{\omega}_i) \tag{2.70}$$

where

W_i : the mass of link i

$[J_i]$: the 3 x 3 inertial matrix of link i

Note that the off-diagonal terms of J matrix may be vanished if the link is assumed to be a circular column.

Based on Equations (2.67) to (2.70) we can designate the following recursive relationships in terms of the same notation used as previous procedures for evaluating velocities and accelerations.

$$^{r}\underline{F}_i = W_i\, ^{r}\dot{\underline{v}}_i^* \tag{2.71}$$

$$^{r}\underline{M}_i = [J_i]\, ^{r}\dot{\underline{\omega}}_i + ^{r}\dot{\underline{\omega}}_i \times ([J_i]\, ^{r}\dot{\underline{\omega}}_i) \tag{2.72}$$

$$^{r}\underline{f}_i = ^{r}\underline{F}_i + ^{r}\underline{f}_{i+1} \tag{2.73}$$

$$^{r}\underline{m}_i = ^{r}\underline{m}_{i+1} + ^{r}_{i}\underline{O}_{i+1} \times ^{r}\underline{f}_{i+1} + ^{r}_{i}\underline{G}_i \times ^{r}\underline{F}_i + ^{r}\underline{M}_i \tag{2.74}$$

The procedure for evaluating the forces and moments at each joint can be stated as follows:

Step 1: obtain $^{3}\underline{f}_4$ by $f = [^{3}Q_0]\, (\underline{z}W_4)$, where W_4 is the load on the end-effector.

Step 2: set $i = n$ and $r = n$, where n is the number of DOF.

Step 3: obtain ${}^{r}\underline{F}_i$ using Equation (2.71)

Step 4: obtain ${}^{r}\underline{M}_i$ using Equation (2.72)

Step 5: obtain ${}^{r}\underline{f}_i$ using Equation (2.73)

Step 6: obtain ${}^{r}\underline{m}_i$ using Equation (2.74)

Step 7: set $i = i - 1$ and $r = r - 1$.

Step 8: if $i = 0$ then stop, otherwise go to Step 3.

Once we have the joint force or moment at hands, the driving force and torque is the projection of that force or moment on the characteristic axis, which is

$$T_i = \underline{z} \cdot ([{}^{r-1}Q_r]{}^{r}\underline{m}_i), \text{ for revolute pairs} \tag{2.75}$$

or

$$F_i = \underline{z} \cdot ([{}^{r-1}Q_r]{}^{r}\underline{f}_i), \text{ for prismatic pairs} \tag{2.76}$$

The damping effects of the actuators or joints may be included simply by adding one term, $b_i\dot{\theta}_i$, in above equations, where b_i is the viscous coefficient at joint i.

For the most popular 3R robot, the driving torques at each joint can be obtained by following the procedures stated above and are listed as below. Note that the links considered here are assumed to be circular columns.

$$\begin{aligned} T_3 = {} & W_4 g_3 \ell_3 C_{23} + G_3^* W_3 (\dot{\theta}_1^2 \ell_2 C_2 S_{23} + \dot{\theta}_2^2 \ell_2 S_3 + \ddot{\theta}_2 \ell_2 C_3 + g C_{23} \\ & + \dot{\theta}_1^2 G_3^* C_{23} S_{23} + (\ddot{\theta}_2 + \ddot{\theta}_3) G_3^*) + I_{3z}(\ddot{\theta}_2 + \ddot{\theta}_3) - (I_{3x} - I_{3y}) \dot{\theta}_1^2 \\ & C_{23} S_{23} + b_3 \dot{\theta}_3 \end{aligned} \tag{2.77}$$

$$T_2 = T_3 + I_{2z}\ddot{\theta}_2 - (I_{2x} - I_{2y})\dot{\theta}_1^2 S_2 C_2 + \ell_2 W_4 g C_2 + W_3 \ell_2 [\dot{\theta}_1^2 \ell_2 C_2 S_2$$

$$+ \ddot{\theta}_2 \ell_2 + g C_2 + \dot{\theta}_1^2 G_3^* C_{23} S_2 - (\dot{\theta}_2 + \dot{\theta}_3)^2 G_3^* S_3 + (\ddot{\theta}_2 + \ddot{\theta}_3) G_3^* C_3]$$

$$+ W_2 G_2^* (\theta_2 G_2^* + \dot{\theta}_1^2 G_2^* S_2 C_2 + g C_2] + b_2 \dot{\theta}_2 \qquad (2.78)$$

$$T_1 = I_{3x} S_{23} (\ddot{\theta}_1 S_{23} + \dot{\theta}_1 (\dot{\theta}_2 + \dot{\theta}_3) C_{23}) + I_{2x} (\ddot{\theta}_1 S_2^2 + \dot{\theta}_1 \dot{\theta}_2 C_2 S_2)$$

$$+ I_{3y} C_{23} [\ddot{\theta}_1 C_{23} - 2\dot{\theta}_1 (\dot{\theta}_2 + \dot{\theta}_3) S_{23}] + I_{2y} C_2 (\ddot{\theta}_1 - 2\dot{\theta}_1 \dot{\theta}_2 S_2)$$

$$- W_3 (G_3^* C_{23} + \ell_2 C_2) [2\ell_2 \dot{\theta}_1 \dot{\theta}_2 S_2 - \ddot{\theta}_1 \ell_2 C_2 + 2\dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_3) G_3^* S_{23}$$

$$- \ddot{\theta}_1 G_3^* C_{23}] - W_2 G_2^2 C_2 (2\dot{\theta}_1 \dot{\theta}_2 S_2 - \dot{\theta}_1^2 C_2) + I_{1y} \ddot{\theta}_1 + b_1 \dot{\theta}_1 \qquad (2.79)$$

where I_{ij} is the principal moment of inertial about j-axis of link i.

2.3 Formulation of Performance Index

Mathematical modeling of a robot system includes modelings of the mechanical part of the system (manipulator itself) and the actuators. In this study the actuators are assumed to be ideal, in other words, we consider only the mechanical part of the system. For the problem of optimal trajectory with the criterion of minimum energy consumption, the energy consumed during the entire motion can be devided into two parts: (1) the static portion, due to the potential energy change, and (3) the dynamic portion needed to drive the manipulator with certain velocity and acceleration within certain traveling or cycling time. In this study all dynamic effects in addition to the viscous effect are considered. As we mentioned in Chapter I, there are three expressions widely used to stand for the energy consumption in a robot system.

Here, two of them are used. One is the Chow's [6] quadratic criterion while the other is the actual energy dissipated. Let us denote the former one as J_1 while the later one as J_2, then we have

$$J_1 = \int_{t_0}^{t_1} (T_1^2 + T_2^2 + \ldots) \, dt \tag{2.80}$$

$$J_2 = \int_{t_0}^{t_1} (T_1\ddot{\theta}_1 + T_2\ddot{\theta}_2 + \ldots)dt \tag{2.81}$$

For the most popular 3R robot, J_1 and J_2 can be written, respectively, as

$$J_1 = \int_{t_0}^{t_1} (T_1^2 + T_2^2 + T_3^2)dt \tag{2.81}$$

$$J_2 = \int_{t_0}^{t_1} (T_1\ddot{\theta}_1 + T_2\ddot{\theta}_2 + T_3\ddot{\theta}_3)dt \tag{2.83}$$

Consequently, the optimal trajectory problem can be stated as a problem of minimizing the performance index J_1 or J_2, as expressed in Equation (2.80) and (2.81) with respect to the trajectories of joint displacement $\theta_i(t)$, i = 1, 2 . . ., n, under the constraints of reachability, as expressed in Equation (2.84), and driving ability of actuators, as expressed in Equation (2.85).

$$\begin{cases} \theta_{i,min} < \theta_i < \theta_{i,max}, \text{ for revolute joint i} \\ \\ s_{j,max} < s_j < s_{j,max}, \text{ for prismatic joint j} \end{cases} \tag{2.84}$$

$$T_{i,max} < |T_i| \quad \text{, for revolute joint } i$$

$$F_j, max < |F_j| \quad \text{, for prismatic joint } j \tag{2.85}$$

where

$\theta_{i,min}$: the lower limit of motion of revolute joint i

$s_{j,min}$: the lower limit of motion of prismatic joint j

$\theta_{i,max}$: the upper limit of motion of revolute joint i

$s_{j,max}$: the upper limit of motion of prismatic joint j

$T_{i,max}$: the maximal allowable actuating torque at revolute joint i

$F_{i,max}$: the maximal allowable actuating force at prismatic joint j

The problem we stated above is a typical variational problem of optimizing a functional. The solution to this problem will be discussed and developed in the next chapter.

CHAPTER III

PROPOSED ALGORITHM OF OPTIMIZATION

In the present chapter, we will, first, take a brief review on the fundamental theories of variational calculus, Bellman's Principle of Optimality (Dynamic Programming), Pontryagin's maximum principle, and then Hooke and Jeeves pattern search technique. An algorithm based on the principle of optimality, the maximum principle and the pattern search is developed for finding the optimal trjaectory with the criterion of foregoing derived performance index as the objective function.

3.1 Fundamental Theories of the Variational Calculus

The basic problem that the variational calculus attempts to solve is the problem of finding a function $\theta(t)$ which minimize the functional J

$$J = \int_{t_0}^{t_1} F(\theta(t), \dot{\theta}(t), t)dt \tag{3.1}$$

subjected to two-end boundary conditions, i.e.,

$$\theta(t_0) = \theta_0 \tag{3.2}$$

$$\theta(t_1) = \theta_1 \tag{3.3}$$

As a matter of fact, the fundamental equation of the variational calculus was derived by Euler by means of a passage to the limit from the finite to the infinite. Some classical results derived by Euler may be stated as follows.

Let $\theta(t)$ denote a minimizing function, which we consider to furnish a relative minimum. If $z(t)$ is any "nearby" function, we must then have

$$J(z) \geq J(\theta) \tag{3.4}$$

To represent the fact that $z(t)$ is a nearby function, let us write

$$z(t) = \theta(t) + \varepsilon g(t) \tag{3.5}$$

where ε is a small parameter and $g(t)$ may be any arbitrary function satisfying

$$g(t_0) = 0 \text{ and } g(t_1) = 0 \tag{3.6}$$

The inequality in Equation (3.4) then becomes

$$J(\theta + \varepsilon g) \geq J(\theta) \tag{3.7}$$

for all and $g(t)$, or, in explicit terms,

$$\int_{t_0}^{t_1} F(\theta + \varepsilon g, \dot{\theta} + \varepsilon \dot{g}, t)dt \geq \int_{t_0}^{t_1} F(\theta, \dot{\theta}, t)dt \tag{3.8}$$

From the condition that J has relative minimum, we have

$$\frac{\partial J}{\partial \varepsilon}\Big|_{\varepsilon = 0} = 0 \tag{3.9}$$

Expanding Equation (3.9) and substituting Equations (3.2) and (3.3) into it, then using the technique of integral by parts, we obtain

$$\int_{t_0}^{t_1} g(t)[F_\theta - \frac{d}{dt}(F_{\dot{\theta}})]dt = 0 \tag{3.10}$$

for all admissible g(t). Were this true for all g(t), it would be trivial that the coefficient function must be zero i.e.,

$$F_\theta - \frac{d}{dt}(F_{\dot{\theta}}) = 0 \tag{3.11}$$

which is called the Euler equation of the variational problem and is a necessary condition completely analogous to that derived in the finite dimensional case by setting the first partial derivatives equal to zero.

Since Equation (3.11) is a nonlinear partial differential equation, only in rare cases is an explicit analytical solution of this equation obtainable. Consequently, many attempts have been made to solve this variational problem by some other numerical approaches instead of the analytical approach. Some of the famous methods are Rayleigh-Ritz method, the principle of optimality and the maximum principle. Since Rayleigh-Ritz method has, inherently, the problem of convergence, in this study we attempt to apply the principle of optimality and the maximum principle to solve the problem stated in the previous chapter.

3.2 The Principle of Optimality

The optimization of sequential systems can be considerably simplified by the use of the principle of optimality first formally

stated by Bellman [3], who states the principle of optimality as "an optimality policy has the property that, whatever the initial state and the initial decisions are, the remaining decisions must constitute an optimal policty with regard to the state resulting from the first decision." Let us now sketch, mathematically, the dynamic programming approach to the variational problem.

Consider the problem of minimizing the functional J, expressed as Equation (3.1), subjected to the initial condition

$$\theta(t_0) = \theta_0 \tag{3.12}$$

The minimum value will then be a function of the initial t-value t_0, and the initial θ-value θ_0. Let us introduce the function

$$f(t_0, \theta_0) = \min_{\theta} J(\theta) \tag{3.13}$$

What we are trying to do is to imbed the particular problem posed above where t_0 and θ_0 are constants within the family of problems generated by allowing t_0 and θ_0 to be parameters with the range of variation $-\infty < t < \infty$ and $-\infty < \theta < \infty$. Since the integral has the requisite additivity property

$$\int_{t_0}^{t_1} = \int_{t_0}^{t_0+\Delta} + \int_{t_0+\Delta}^{t_1} \tag{3.14}$$

the fundamental mathematical expression for the principle of optimality stated previously then may be written as

$$f(t_0, \theta_0) = \min_{\theta[t_0, t_0+\Delta]} \left[\int_{t_0}^{t_0+\Delta} F(\theta, \dot{\theta}, t)dt + f(t_0 + \Delta, \theta_0(\theta)) \right] \tag{3.15}$$

where the minimization is over all functions defined over $t_0 < t < t_0+\Delta$, with $\theta(t_0) = \theta_0$ and $\theta_0(\theta) = \theta(t_0 + \Delta)$. Two examples known as "minimum chain problems" are investigated in order to expose the spirit behind the statement.

With reference to the chain network as shown in Figure 7, which consists of nodes and branches between nodes, a minimal length connected chain (or simply stated, a minimal chain) between A and B is desired. The "length" assigned to each branch could represent, say, a normalized cost of sending a message between nodes. Considering A as the origin and B as the point (0,6), we have the recurrence relation

$$f(x,y) = \min[d(x,y:x+1, y+1) + f(x+1,Y+1), \quad d(x,y:x-1,y-1) + f(x-1, y-1)] \tag{3.16}$$

where d(x,y:x+1, y+1) represents the length on the branch between the points (x,y) and (x+1, y+1). To solve, we first note that f(5,1)=4 and f(5,-1)=3. Using these values for f(5,y) we determine f(4,y). Thus f(4,2) = 1+4=5, f(4,0)=min[5+4,2+3]=5 and f(4,-2)=6+3=9. We also note that from f(4,0) our optimal choice is to go diagonally down. In an analogous manner we compute f(3,y), f(2,y), f(1,y), and finally f(0,0). The solution, with arrows indicating the optimal direction, is so called "backward solution" and summarized by the chart of values appearing in Figure 8. If we start the iteration from the initial point A instead of the end point B, we can also obtain the same solution by applying the same algorithm and it is called "forward solution" as shown in Figure 9.

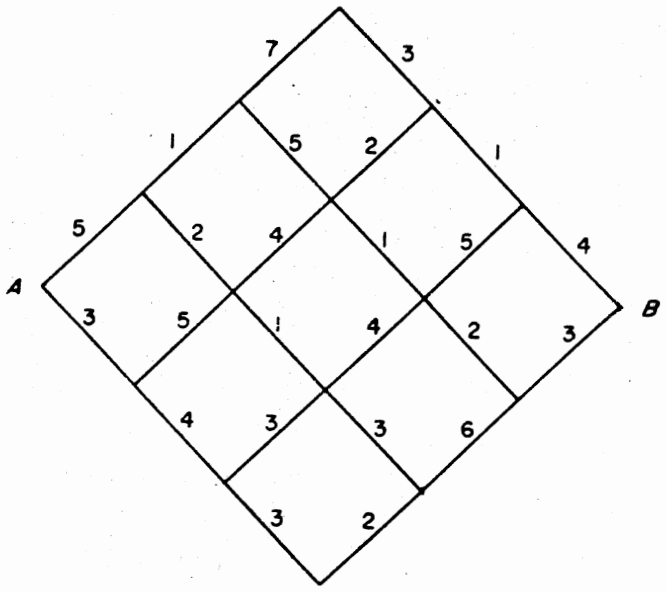

Figure 7. The Chain Network #1

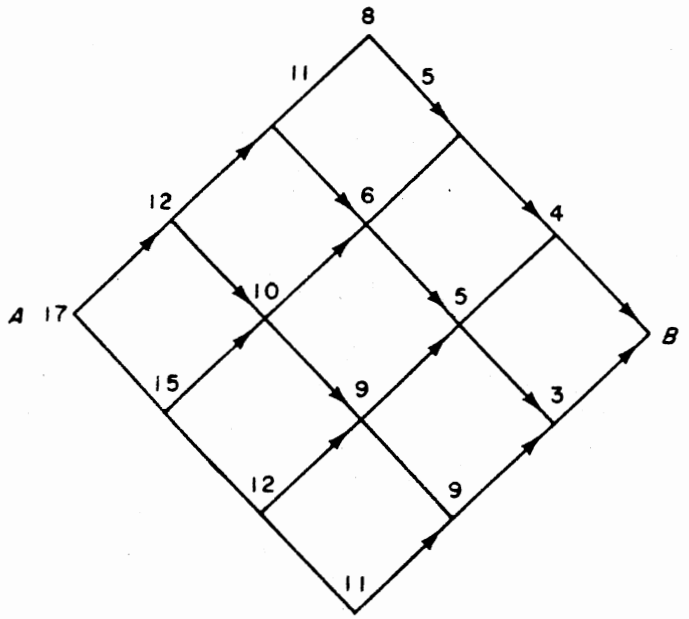

Figure 8. The Backward Solution

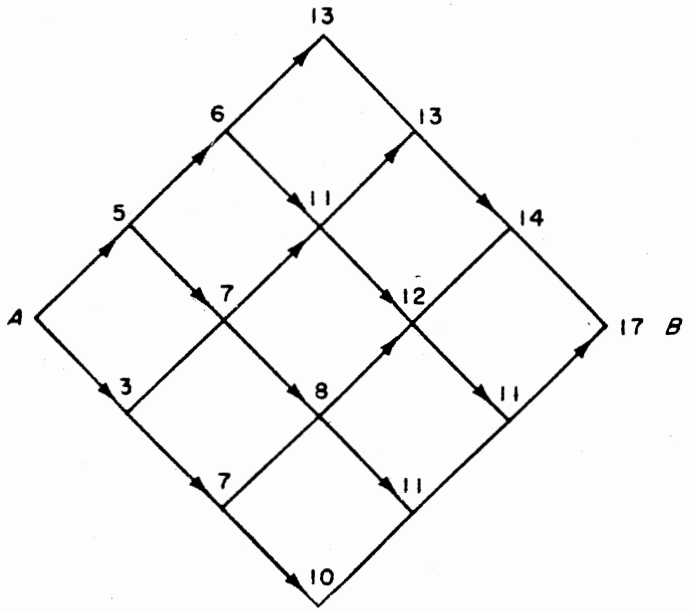

Figure 9. The Forward Solution

For the cases of both the initial and ending points are fixed either approach can be applied to find the optimal solution. However, if, for instance, the initial point is floating then only backward approach can be applied, and vice versa.

Let's consider another similar but more straight forward and simpler case. Referring to Figure 10, there are many (may be infinitive) branches between two neighbor nodes. An optimal polciy of solution is easy to set up in this case. "Either starting from A or B find the minimum branch between any two neighbor stations" gives us the minimal chain indicated by arrows as shown in Figure 10.

As we can see from the preceding example, the principle of optimality is suggestive of the fact that a given problem to be solved is placed in a dynamic frame work; the problem is imbedded in a class of similar problems, the situation of which are solvable by using dynamic programming. The phrase implies that a problem which is amenable to dynamic programming can be viewed as a succession of decision problems, each one building on the last, until the problem is solved.

However, we have to mention here that Bellman's approach yields a good heuristic method rather than a mathematical solution to the problem and is very valid while dealing with discrete problems with finite decisions to be made at each stage. For amulti-variable continuous problem, like the one we are trying to solve in this study, this method will inevitably encounter the difficulty of dimentionality in addition to that a huge number of grids has to be assigned for an acceptable result with certain accuracy. Nevertheless, the huristic concept of solving the variation problem using the physical phenomenon imbedded in that problem is adapted in our proposed algorithm in section 3.5.

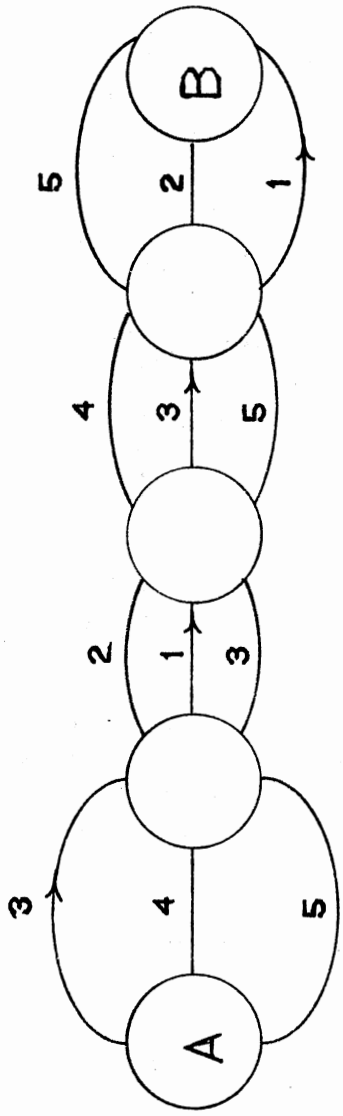

Figure 10. The Chain Network #2

3.3 The Maximum Principle

The maximum principle stated by Pontryagin [21] permits the solution of various problems of a mathematical and applied nature, problems which are variational but which do not fit into the classical arrangement of variational calculus. One may observe that this principle plays an important role on the theoretical basis for the algorithm we proposed in this study. The maximum principle, actually, is composed by two rules which we will introduce and interpret as below.

Consider the functional given Equation (3.1), in which if F does not depend explicitly on the variable t, then F is called autonomous. With reference to Figure 11, the first rule of the maximum principle can be stated in a following manner: for an autonomous system, the properties of the trajectories $\theta(t)$ do not change with displacment along t-axis. In other words, if the trajectory $\theta(t)$ having an optimal value of J and satisfying the two-end boundary condition, then the trajectory $\theta(t+h)$, $t_0 - h < t < t_1 - h$, also has the same optimal value J. If we turn our attention to our objective functional expressed by Equations (2.80) and (2.81), we find that the first rule of the maximum principle is applicable to our cases. Actually, in our case, we indeed care only about the total traveling time, which is the difference between the initial time t_0 and the final time t_1, instead of the real t_0 and t_1.

As for the second rule of the maximum principle, let us denote the values of the integral as Equation (3.1), taken over the intervals $t_0 < t < t_a$, $t_a < t < t_b$, $t_b < t < t_1$, by J_1, J_2 and J_3, respectively, as shown in Figure 12. The total functional then has the value $J = J_1 + J_2 + J_3$. If the trajectory $\theta(t)$ considered on the interval

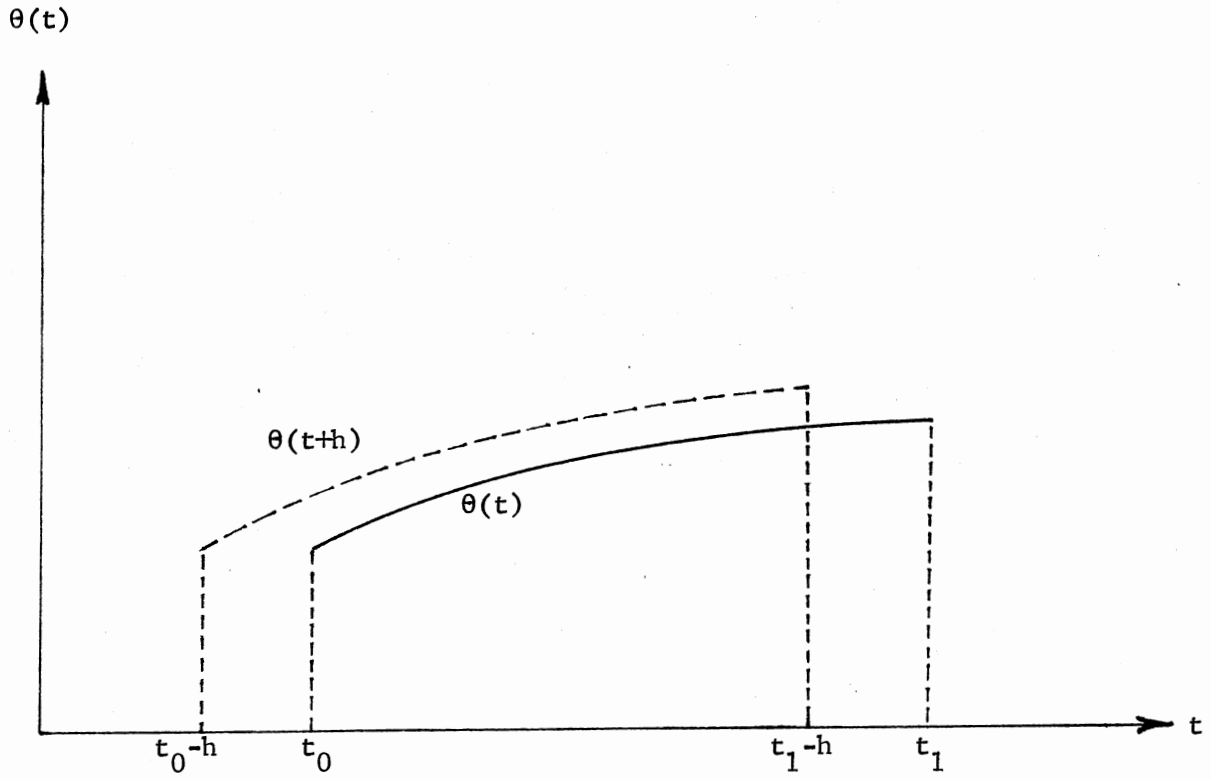

Figure 11. The First Rule of the Maximum Principle

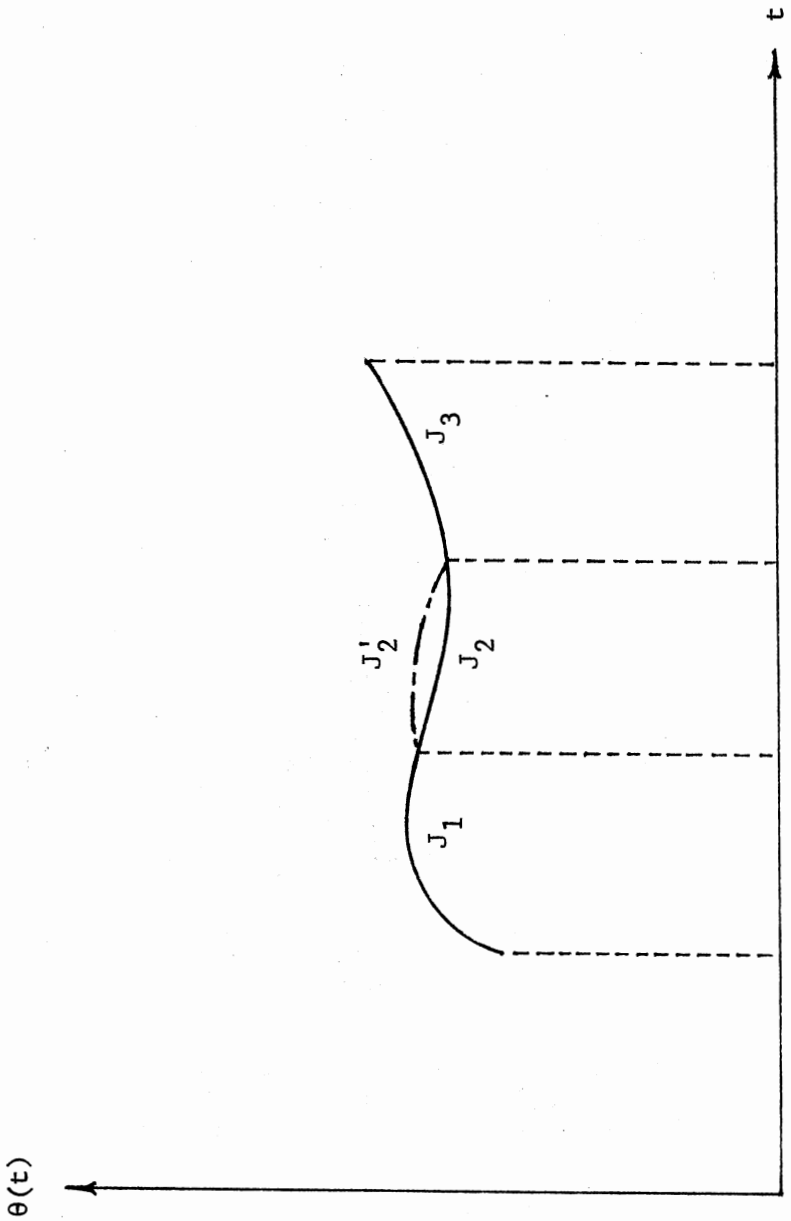

Figure 12. The Second Rule of the Maximum Principle

$t_a < t < t_b$ were not optimal, there would exist some other trajectory $\theta'(t)$ having the value $J' < J$, which contradicts the optimality of the trajectory $\theta(t)$, $t_a < t < t_b$. It readily follows what has been said above that every piece of an optimal trajectory is itself an optimal trajectory, which is the second rule of the maximum principle.

3.4 Pattern Search

The pattern search technique devised by Hooke and Jeeves [5] is one of direct optimum searching methods, and is based on the hopeful conjecture that any set of moves; that is, adjustments of the independent variables, which have been successful during early experiments will be worth trying again. The computation time was found empirically to be only the first power of the number of variables.

In visualizing what is mean by a "pattern" it is helpful to think of an arrow, with reference to Figure 13, its base at one end and its head at the other. The search begins at a base point $\underline{b}$ which may be chosen arbitrarily; as yet the pattern has not been established. The experimenter chooses a step size for each independent variable $x_i (i=1, 2, \ldots, k)$. Let $\underline{\delta}$ be the vector whose ith component is $\underline{\delta}_i$, all the rest being zero. After measuring the criterion at the initial base $\underline{b}_1$ one takes an observation at $\underline{b}_1 + \underline{\delta}_1$. If this new point is better than the base, we call $\underline{b}_1 + \underline{\delta}_1$ the temporary head $\underline{t}_{11}$, where the double subscript shows that we are developing the first pattern and that we have already perturbed the first variable x_1. Now $\underline{b}_1 + \underline{\delta}_1$ may not be as good as $\underline{b}_1$, in which case we discard $\underline{b}_1 + \underline{\delta}_1$ and try $\underline{b}_1 - \underline{\delta}_1$. If this new point is better than $\underline{b}_1$, we make $(\underline{b}_1 - \underline{\delta}_1)$ or $(\underline{b}_1 + \underline{\delta}_1)$ the temporary head; otherwise $\underline{b}_1$ is designated temporary head. In summary,

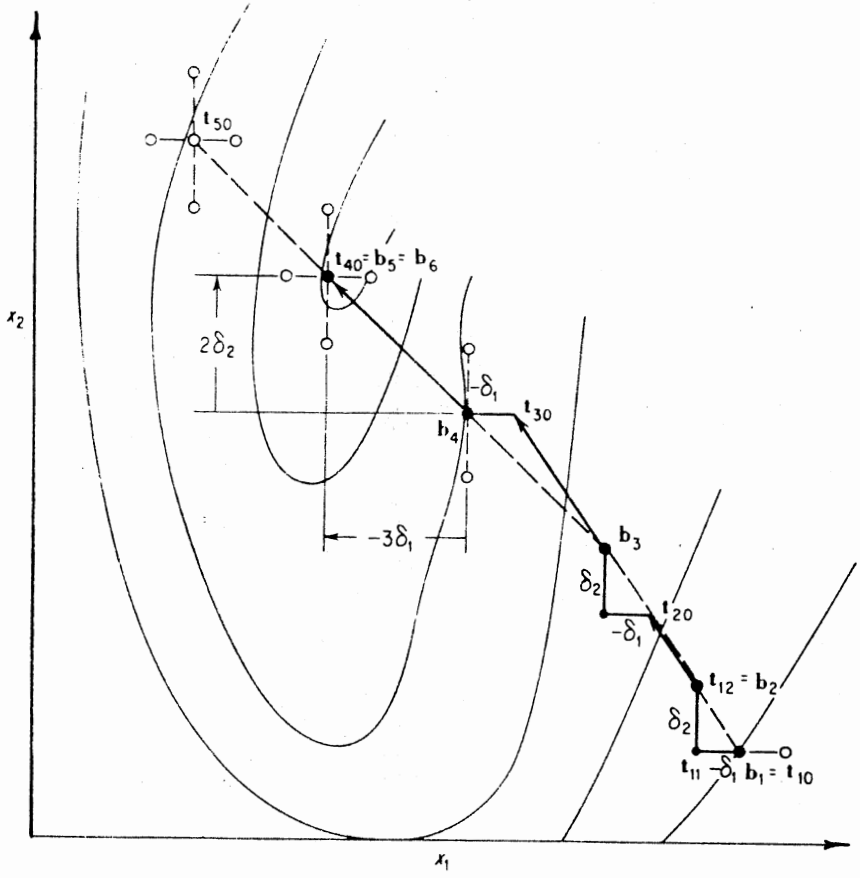

Figure 13. Pattern Search on a 2-Dimensional Plane

when we are minimizing,

$$\underline{t}_{11} = \begin{cases} \underline{b}_1 + \underline{\delta}_1 & \text{if } y(\underline{b}_1 + \underline{\delta}_1) < y(\underline{b}_1) \quad (3.17) \\ \underline{b}_1 - \underline{\delta}_1 & \text{if } y(\underline{b}_1 - \underline{\delta}_1) < y(\underline{b}_1) < y(\underline{b}_1 + \underline{\delta}_1) \quad (3.18) \\ \underline{b}_1 & \text{if } y(\underline{b}_1) < \min[y(\underline{b}_1 + \underline{\delta}_1), y(b_1 - \underline{\delta}_1)] \quad (3.19) \end{cases}$$

In Figure 13, Equation (3.18) governs.

Perturbation of x_2, the next independent variable, is now carried out in a similar manner, this time about the temporary head $\underline{t}_{11}$ instead of the original base $\underline{b}_1$. In general, the jth temporary head $\underline{t}_{1j}$ is obtained from the preceeding one $\underline{t}_{1,j-1}$ in the following manner:

$$\underline{t}_{ij} = \begin{cases} \underline{t}_{1,j-1} + \underline{\delta}_j & \text{if } y(\underline{t}_{1,j-1} + \underline{\delta}_j) < y(\underline{t}_{1,j-1}) \quad (3.20) \\ \underline{t}_{1,j-1} - \underline{\delta}_j & \text{if } y(\underline{t}_{1,j-1} - \underline{\delta}_j) < y(\underline{t}_{1,j-1}) < y(\underline{t}_{1,j-1} + \underline{\delta}_j) \quad (3.21) \\ \underline{t}_{1,j-1} & \text{if } y(\underline{t}_{1,j-1}) < \min[y(\underline{t}_{1,j-1} + \underline{\delta}_j), y(\underline{t}_{1,j-1} - \underline{\delta}_j)] \quad (3.22) \end{cases}$$

This expression covers all $1 < j < k$ if we adopt the convention that

$$\underline{t}_{10} = \underline{b}_1 \quad (3.23)$$

In Figure 13, Equation (3.20) applies for j=2. When all of the variables have been perturbed, the last temporary head point $\underline{t}_{1k}$ is designated as the second base point $\underline{b}_2$. That is,

$$\underline{t}_{1k} = \underline{b}_2 \quad (3.24)$$

The original base point $\underline{b}_1$ and the newly determined base point $\underline{b}_2$ together established the first pattern. Reasoning that if a similar exploration were conducted from $\underline{b}_1$ the results are likely to be the same, we skip the local excursions and extend the arrow from $\underline{b}_1$ to $\underline{b}_2$ immediately doubling its length. This establishes a new temporary head $\underline{t}_{20}$ for the second pattern based at $\underline{b}_2$. This initial temporary head is given by

$$\underline{t}_{20} = \underline{b}_1 + 2(\underline{b}_2 - \underline{b}_1)$$

$$= 2\underline{b}_2 - \underline{b}_1 \qquad (3.25)$$

The double subscript indicates that we are building a second pattern and that we not yet begun to perturb the variables. A local exploration about $\underline{t}_{20}$ is now carried out to correct the tentative second pattern if necessary as shown in Figure 13. The logical equations governing establishment of the new temporary heads $\underline{t}_{21}$, $\underline{t}_{22}$, . . . , $\underline{t}_{2k}$ will be similar to Equations (3.20), (3.21) and (3.22) the only difference being that the first subscript will be 2 instead of 1. The reconnaissance is completed when all of the variables have been perturbed, and the last temporary head $\underline{t}_2$, is designated the third base point $\underline{b}_3$, if, as in Figure 13, the outcome there is better than at $\underline{b}_2$.

As before, a new temporary head t is established by extrapolating from $\underline{b}_2$through $\underline{b}_3$.

$$\underline{t}_{30} = 2\underline{b}_3 - \underline{b}_2 \qquad (3.26)$$

We keep on searching by the same algorithm stated above until the moment that none of the temporary heads $\underline{t}_{50}$, $\underline{t}_{51}$, or $\underline{t}_{52}$ are any better

than the fifth base $\underline{b}_5$, as in Figure 13. Then $\underline{b}_6 = \underline{b}_5$ and the pattern is destroyed. By treating $\underline{b}_5$ as a new $\underline{b}_1$ and reducing the step size, we repeat the preceding search procedures until the step size falls below a preselected minimum. In other words, the search terminates and the last temporary head is the minimum we are searching for. The flowchart of the algorithm of pattern search is shownin Figure 14.

3.5 Proposed Algorithm

From the previous discussion on the approaches to the variational problem we can propose an algorithm to solve, numerically, the problem stated in Chapter II. Based on the theories of variational calculus, the principle of optimality, the maximum principle, and Hooke and Jeeves pattern search, the proposed algorithm can be stated as follows.

First, let us assign an initial feasible solution $\theta_i(t)$, i = 1, . . ., n, which satisfy the two-end boundary condition expressed as Equatins (3.2) and (3.3). For the convenience of programming the sinsoidal function is chosen for the case of the most popular 3R robot, which is also proper for most cases. For the problems we are dealing with, we can always set the initial time at zero value, and the final time at the total traveling time T. Considering the performance indices expressed as Equations (2.80) and (2.81), let us divide the whole time spectrum into N's small intervals as shown in Figure 15. The time span of each interval is ΔT. According to the maximum principle, the total performance index J equals to the sum of J_k's, which is the sub-performance index related to the ith interval, i.e.,

$$J = J_1 + J_2 + J_3 + . . . + J_N \tag{3.27}$$

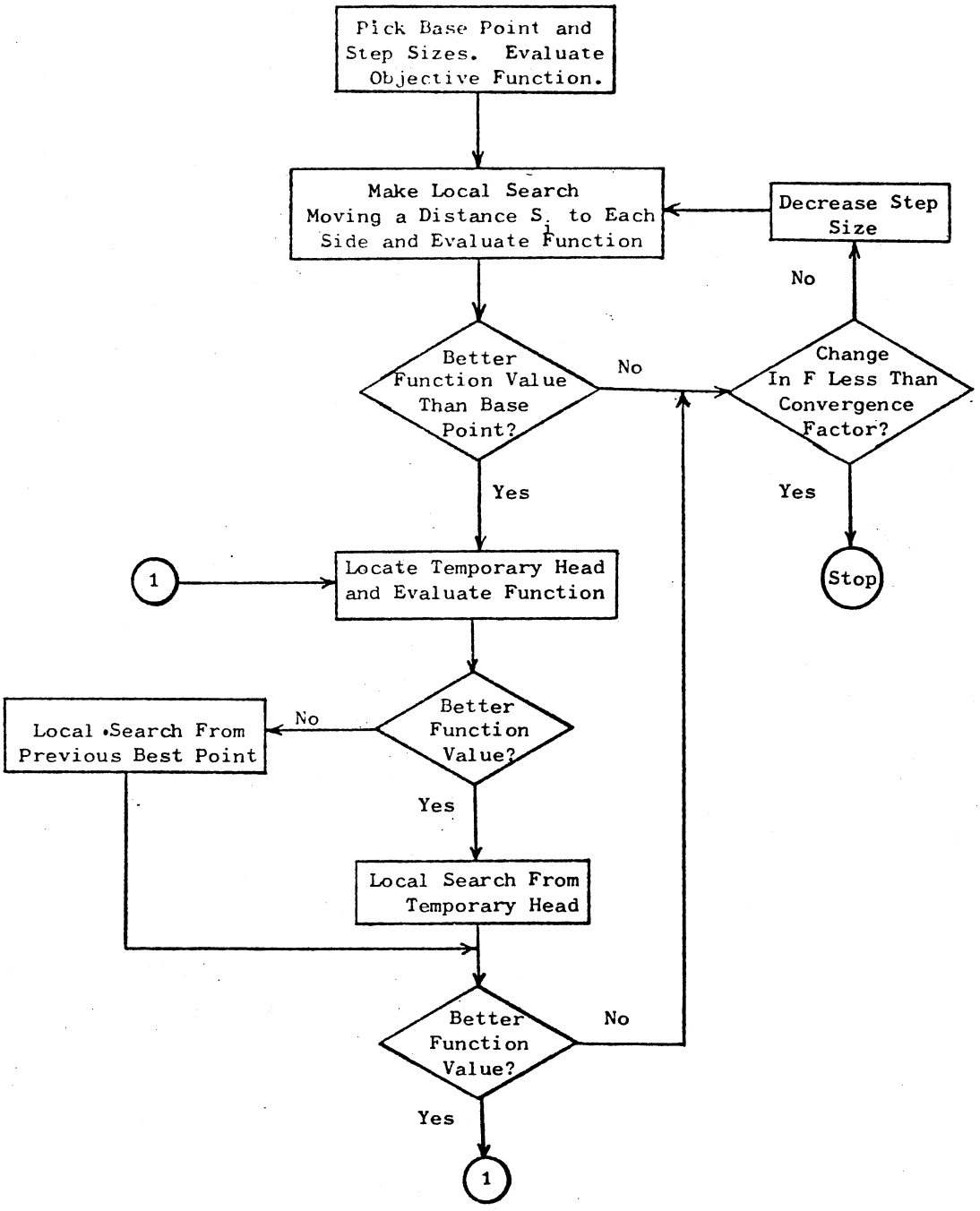

Figure 14. Flowchart of Pattern Search

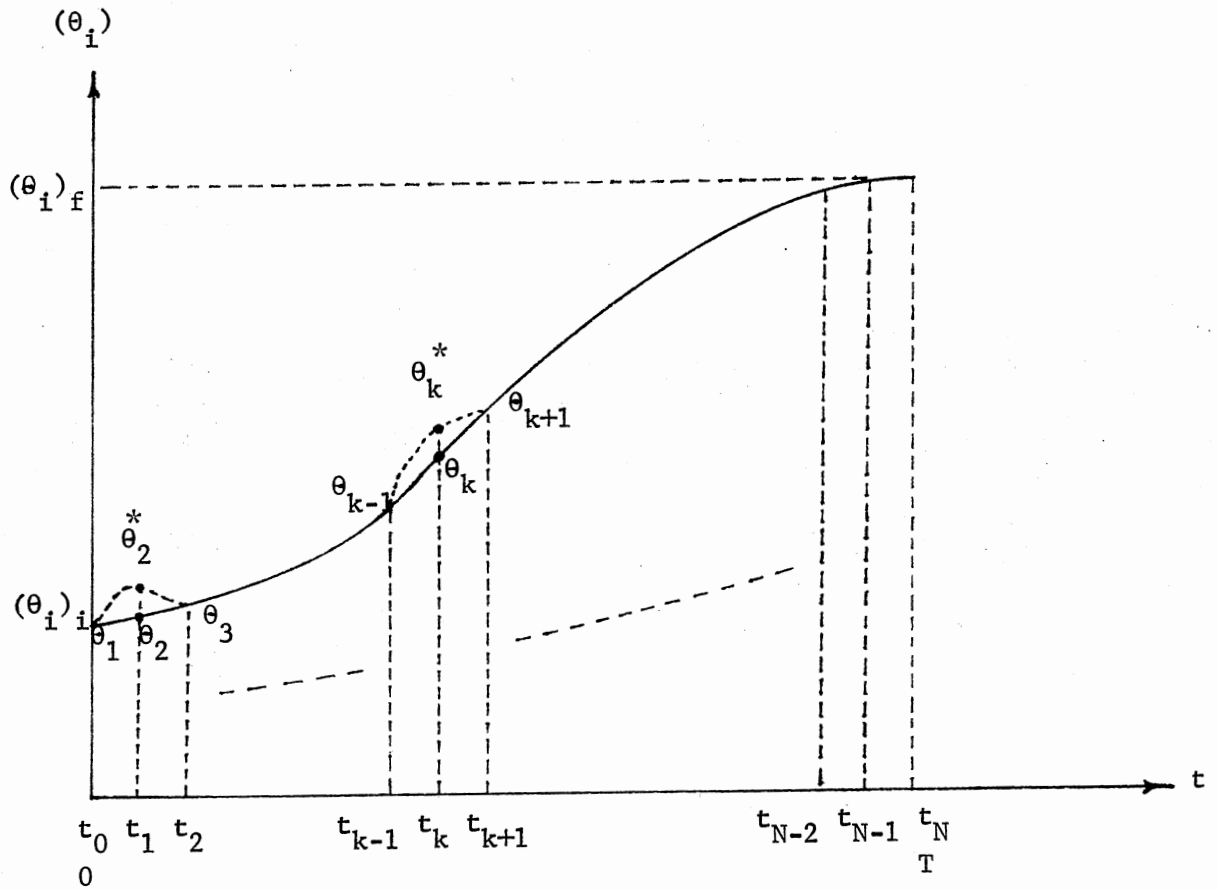

Figure 15. The Proposed Algorithm

Let us assume that in each time interval the actuators conduct a constant acceleration motion, i.e.,

$$\ddot{\theta}_i = \text{constant}, \quad i = 1, 2, \ldots, n \tag{3.28}$$

From the basic kinematic equations of constant acceleration motion, and given the values of θ_{k-1}, $\dot{\theta}_{k-1}$ and θ_k, we may have

$$\ddot{\theta}_k = 2[(\theta_k - \theta_{k-1}) - \dot{\theta}_{k-1} \Delta T]/(\Delta T)^2 \tag{3.29}$$

$$\dot{\theta}_k = \dot{\theta}_{k-1} + \ddot{\theta}_k \Delta T \tag{3.30}$$

Now, let us start with the first interval by setting the initial angular velocities to zero (may be any value as desired), in addition to the values of θ_1 and θ_2 obtained from the initially guessed feasible solution. We can calculate the initial accelerations and velocities at t_1 and t_2. Then the sub-performance indices J_1 and J_2 taken over the first and the second time interval, respectively, can be evaluated from Equations (2.80) and (2.81). Then we have a sub-total performance index J_{02}, taken over the interval $t_0 < t < t_2$, which is used as the performance index of the base for our pattern search. As aresult of the principle of optimality, we keep the θ_1 and θ_3 fixed and treat the θ_2 as a base, and then adjust the θ_2 by the technique of pattern search until an optimal θ_2^* is located. This optimal θ_2^* is recorded in the memory of computer for the use of the next iteration. Then we take the θ_2^* as a new θ_1 and θ_3 as a new θ_2, and proceed the same procedures until the whole time spectrum is completed. Thereafter, we repeat the iteration over the whole time spectrum until the criteria of convergence are satisfied.

During the process of optimization, if any constraint, such as reachability as Equation (2.84) or driving ability as Equation (2.85), needs to be considered, we need only to add several IF statements in the program of pattern search to check whether the constraints are satisfied. If any of the constraints is not satisfied, then we either artifically assign a very big number to the sub-performance index related to that interval, or skip this trial step.

As we can see, the algorithm stated above is very simple not only in the organization but also in the implementation on a computer. Besides, this algorithm can also be used for on-line real-time execution instead of off-line, since it has the nature of learning. In other words, as long as a feasible solution is assigned it will finally reach an optimal solution within a finite number of runs. Especially, if any obstacle, which might be very difficult to describe in mathematics, exists in the working space, the initial feasible solution may be assigned by means of teaching mode, which is available in most of the industrial robots. Figure 16 is the flow chart of the algorithm, in which k is the index of discrete time, n is the DOF of the robot system, N is the number of intervals divided, and i is the index of the joint.

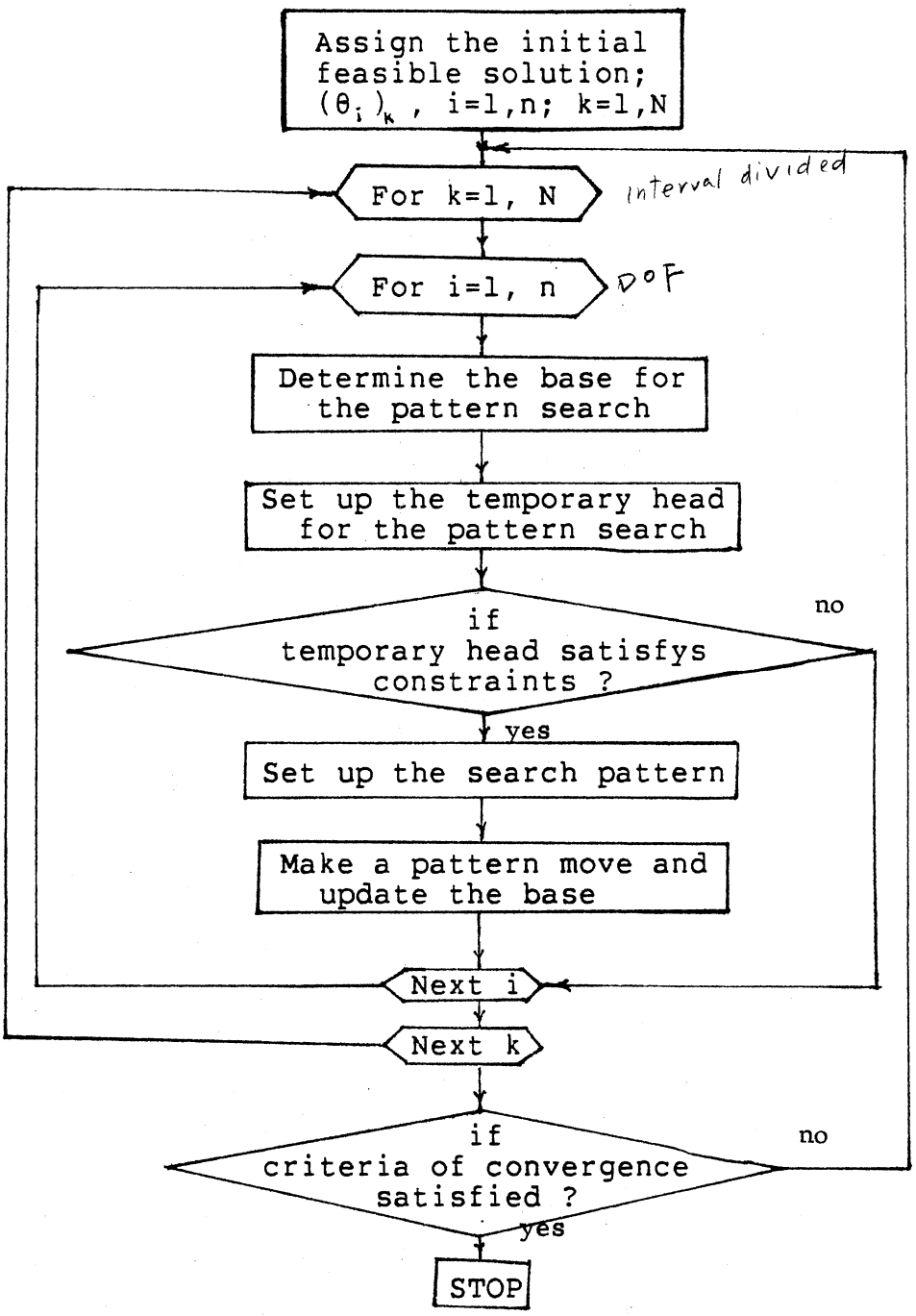

Figure 16. Flowchart of the Proposed Algorithm

CHAPTER IV

AN ILLUSTRATIVE EXAMPLE

In the present chapter, the "most popular 3R robot" as shown in Figure 2, is illustrated to demonstrate the methodology developed in the previous chapters. A computer program coded in FORTRAN is developed to find the optimal trajectory of this robot. The SAS/GRAPH is applied to show the trajectory in a 3-dimensional plot such that the user may have better feeling about the shape of the trajectory.

Two expressions of objective function as Equation (2.80) and (2.81) are used as the performance index, in which T_i's may be computed from Equations (2.77),(2.78) and (2.79). The tarapezoidal rule is applied to evaluate the integration numerically. Here, we like to call the first expression as J_1 and the second expression as J_2. Three cases will be investigated and discussed. The case (A) uses J_1 as the performance index, however, viscous effects are not considered. The case (B) also uses J_1 as the performance index, in addition to that viscous effects are involved. The case (C) uses J_2 as the performance index, in which viscous effects must be included. Since J_2 is the actual energy dissipated during the motion, if the viscous effects are neglected, then the robot system would be in a conservative force field. In other words,the energy consumed will be equal to the change of potential energy, which is a constant in all cases. Also, there is no physical meaning to the optimal trajectory, because any path will give

us the same value of performance index.

The initial and final joint angles in our cases are (0, 70, 210) and (60, 45, 315), respectively, which are arbitrarily assigned and measured in degrees. Correspondingly, the initial and final positions are (.314, .000, .582) and (.520, .901, 1.041), respectively, measured in meters. Other relevant data of this example are listed in Appendix.

The results for the input data as shown in Appendix A are presented in Figure 17 to 29, Table II and Table III. From Table II we find that there are 58.7% improvement, compared with the initial guess, on the performance index for case A, 58.2% for case B, and 59.9% for case C. The CPU time, on IBM 3081D computer, varies from 29.61 seconds, for case C, to 168.1 seconds, for case B, and depends on, according our experience, the initial guessed trajectory, the number of interval divided, the increment of search angle, and the criteria of convergence.

For all cases, as we can see from Figures 17 to 19 that no significant change was made on the displacement profiles of the first and third joints, since the second joint has much higher average torque and dominates the value of J_1 and J_2. Comparing the profiles of joint displacements and torques of case A with those of case B, as shown in Figures 17 to 25, we can find that they are almost the same. In other words, the viscous effects in this particular example have little impact to the performance index. Table III lists the peak values of torque $|T_i|$, $i = 1, 2, 3$, at each joint for all cases. Besides, the profiles of torques for the cases A and B are more smooth than case C is. Figures 26 to 29 shows the true shapes of the initial guessed trajectory and the optimal trajectories in a manner of 3-dimensional plot. Comparing the optimal trajectory of case A with that of case B as shown in

TABLE II. THE NUMERICAL RESULTS OF THE ILLUSTRATIVE EXAMPLE

Case	Initial Obj.	Optimal Obj.	CPU Time (sec)	No. of Interation
A	23873	15042	169.41	70
B	24085	15226	168.08	70
C	236.6	148.0	29.61	9

TABLE III. THE PEAK VALUES OF TORQUE OF EACH JOINT

CASE	1st Joint	2nd Joint	3rd Joint
A	85	205	80
B	85	215	80
C	120	250	140

Figure 27 and 28, we find that they are almost the similar, since both have similar profiles of joint displacements.

Here, we like to emphasize that even though we can determine the profiles of joint displacements and torques, it is difficult to make any general conclusion on what profile we should use for a certain robot with a certain objective function as its performance index. The performance index of a robot system is affected by too many characteristic factors such as the link lengths, the moments of inertia, the viscous coefficient, the configuration of the robot, etc. Any change on these characteristic factors will probably result into different profiles of joint displacements and torques.

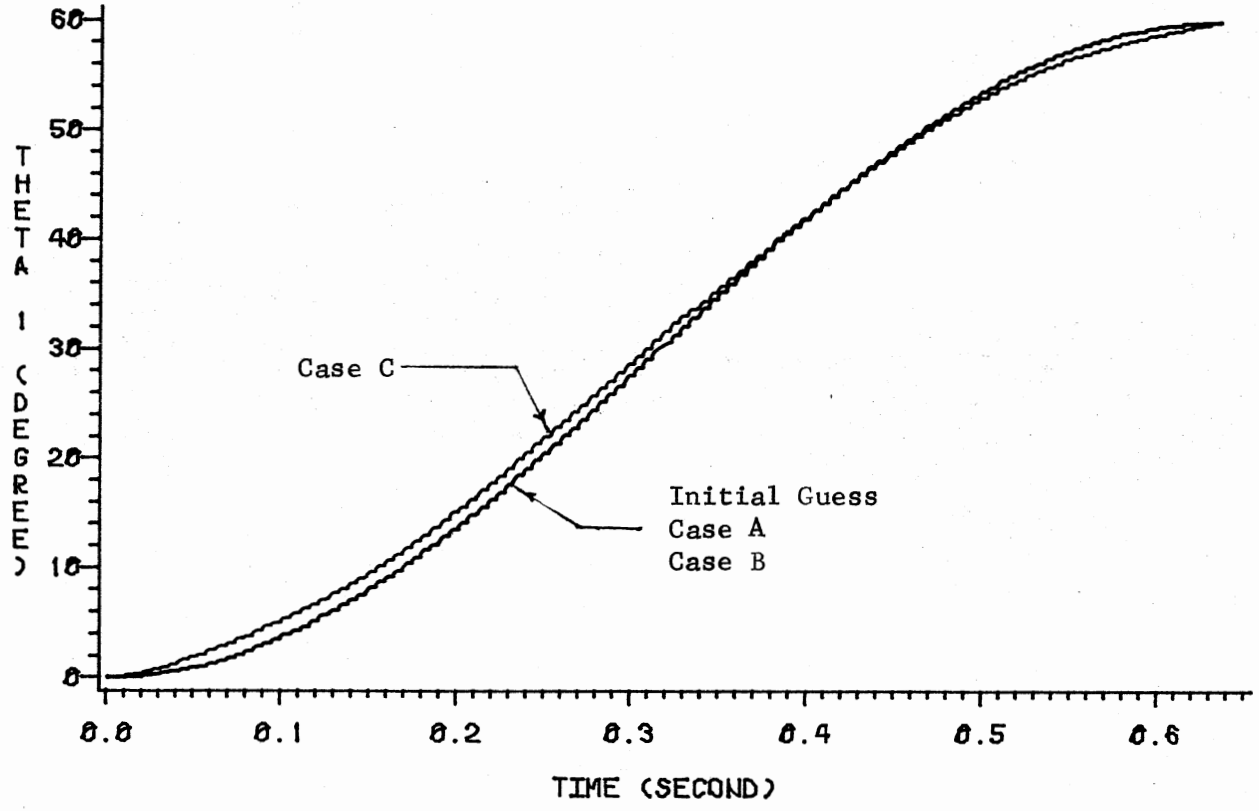

Figure 17. The Profiles of the Displacement of the First Joint

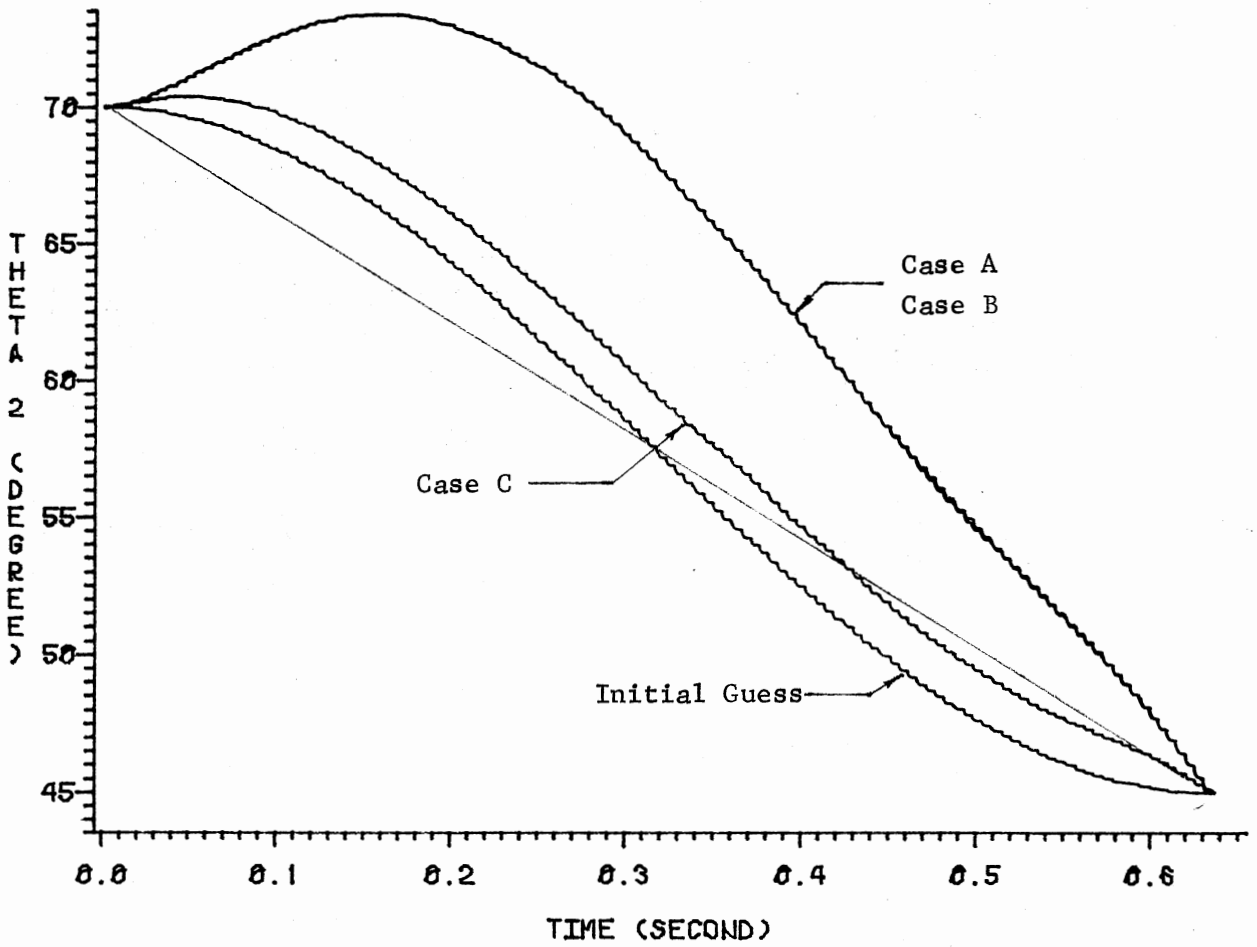

Figure 18. The Profiles of the Displacement of the Second Joint

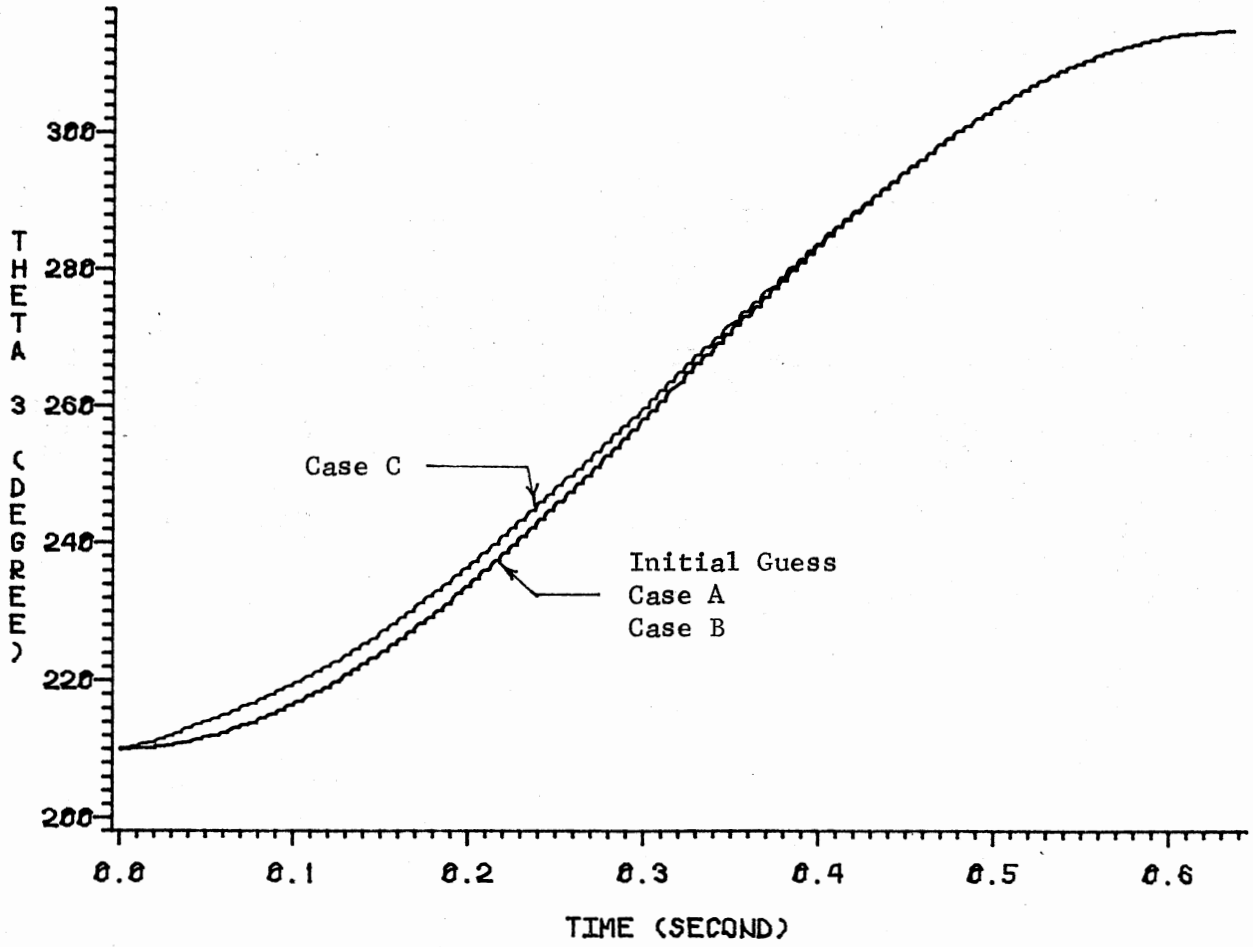

Figure 19. The Profiles of the Displacement of the Third Joint

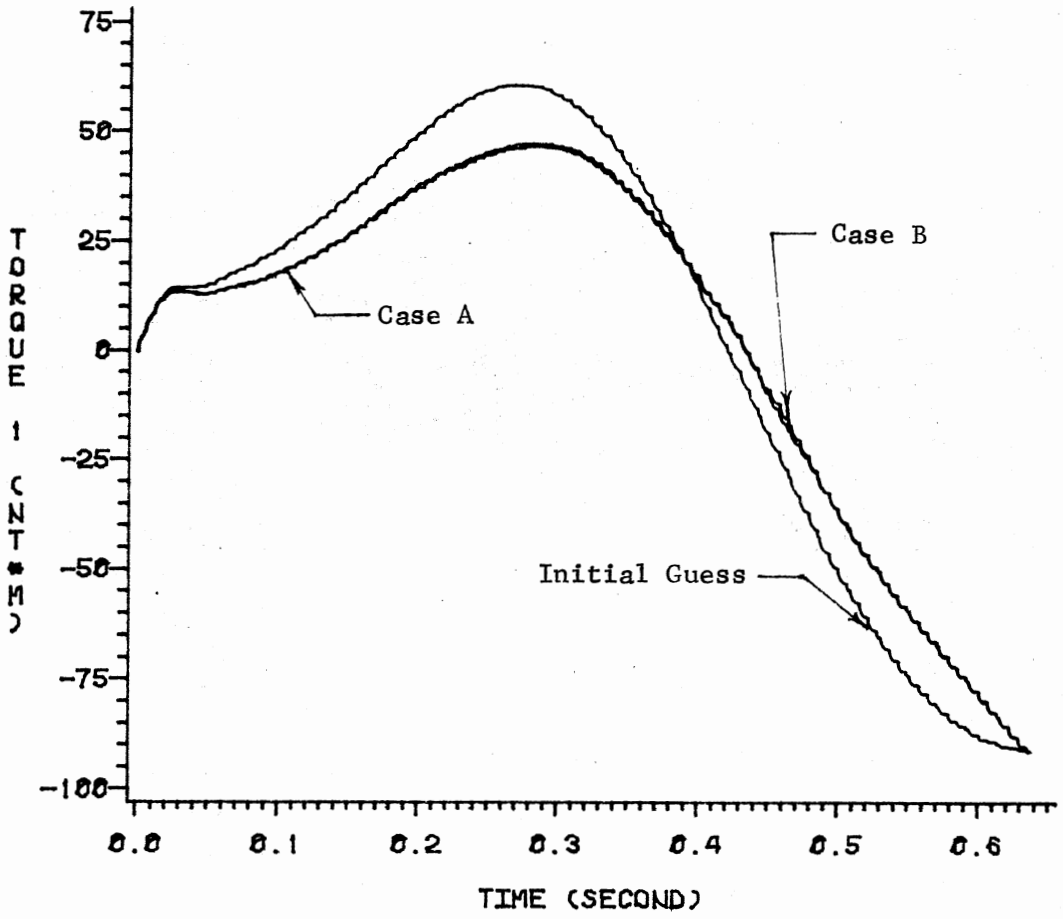

Figure 20. The Profiles of the Torques of the First Joint for the Cases of A and B

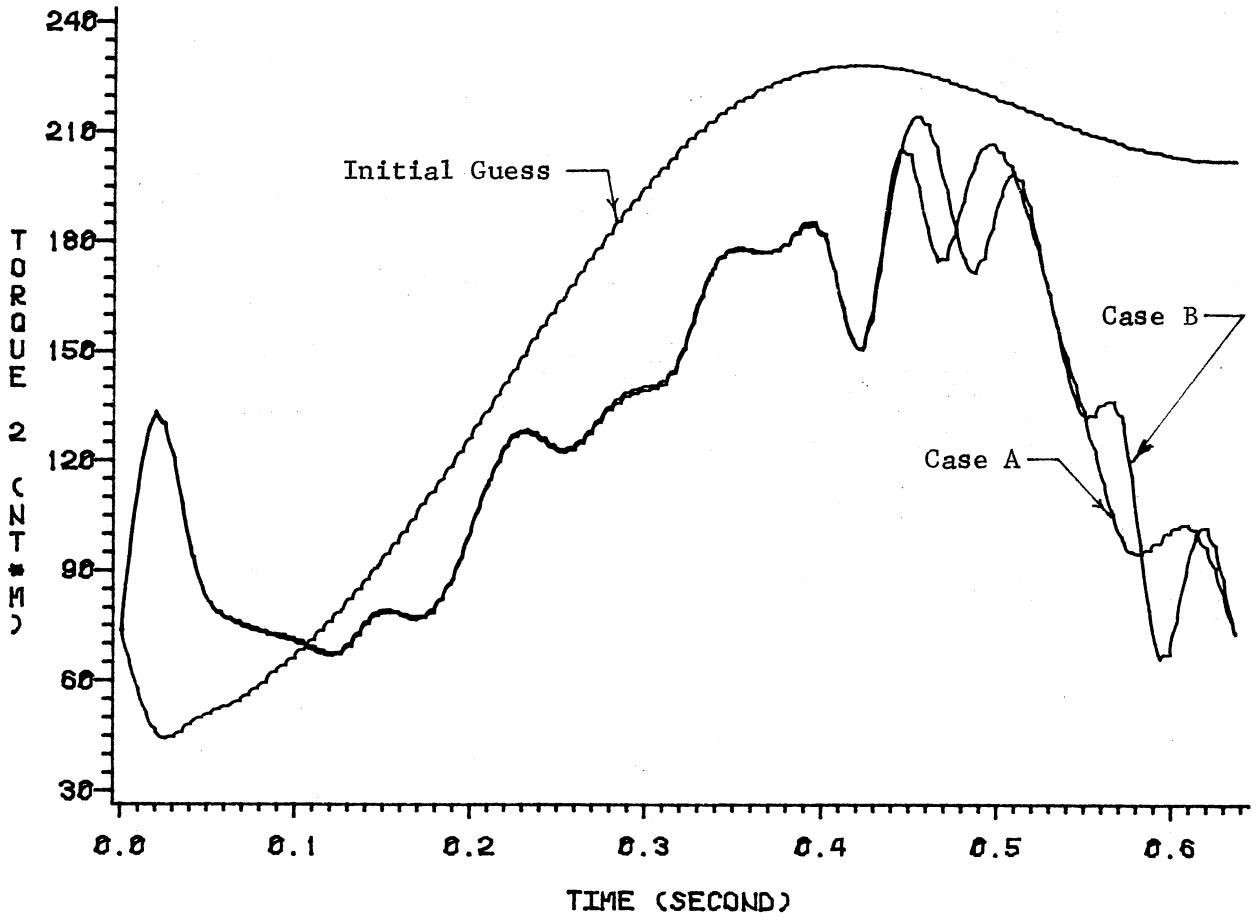

Figure 21. The Profiles of the Torques of the Second Joint for the Cases A and B

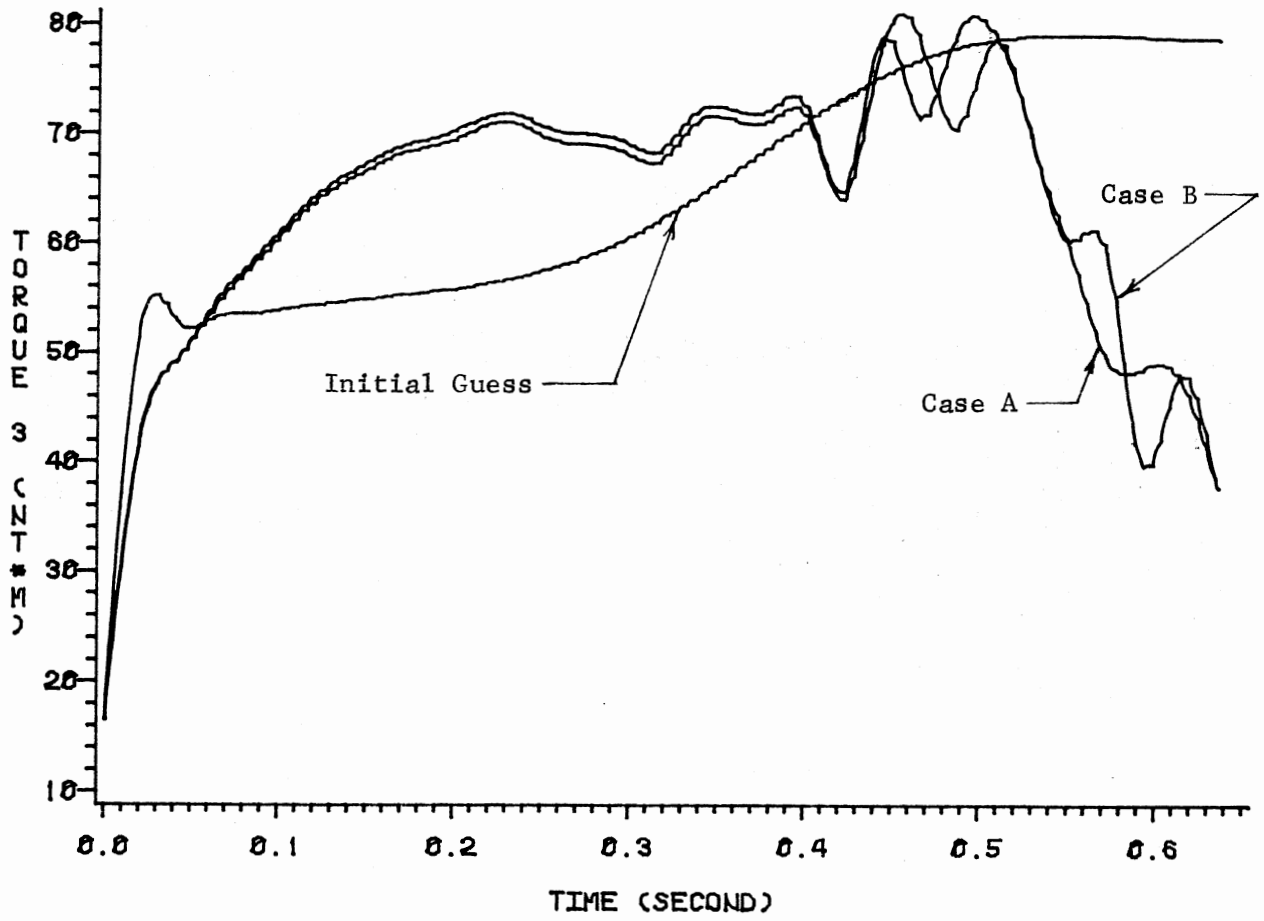

Figure 22. The Profiles of the Torques of the Third Joint for the Cases A and B

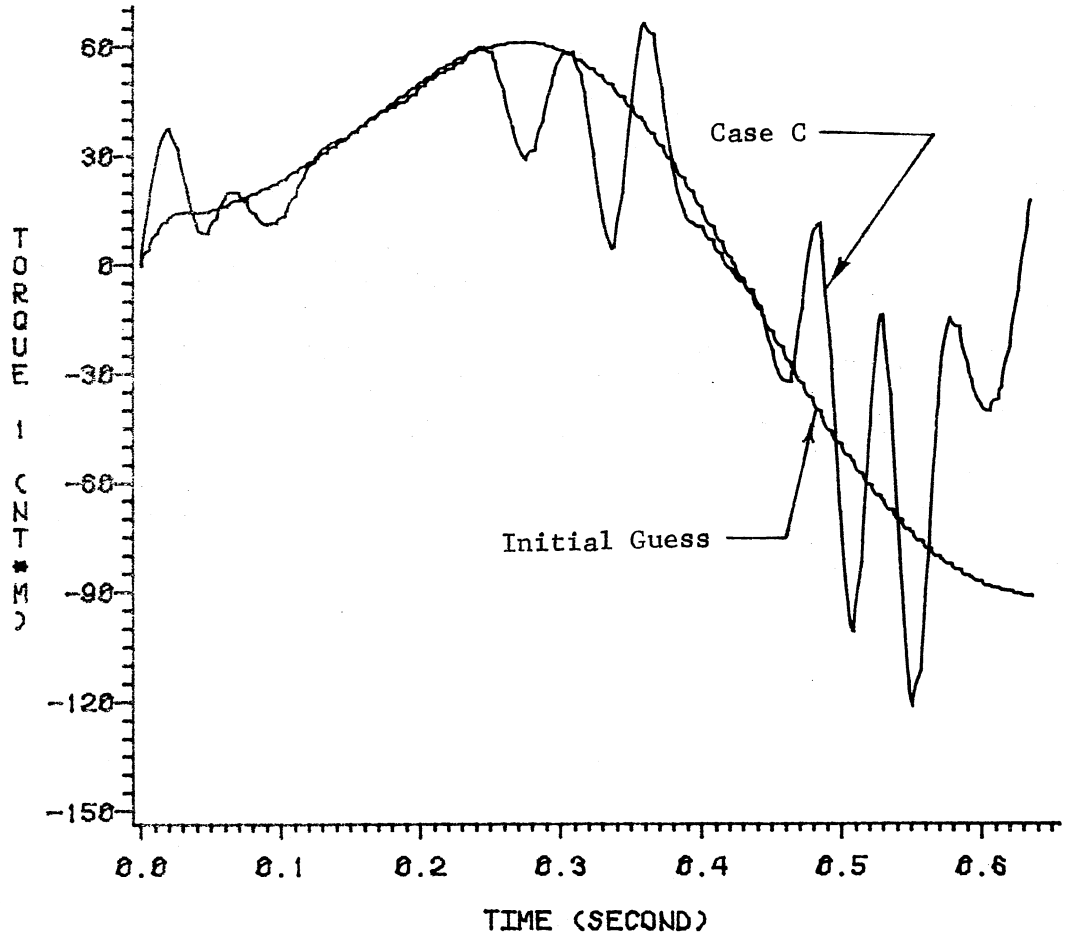

Figure 23. The Profiles of the Torques at the First Joint for the Case C

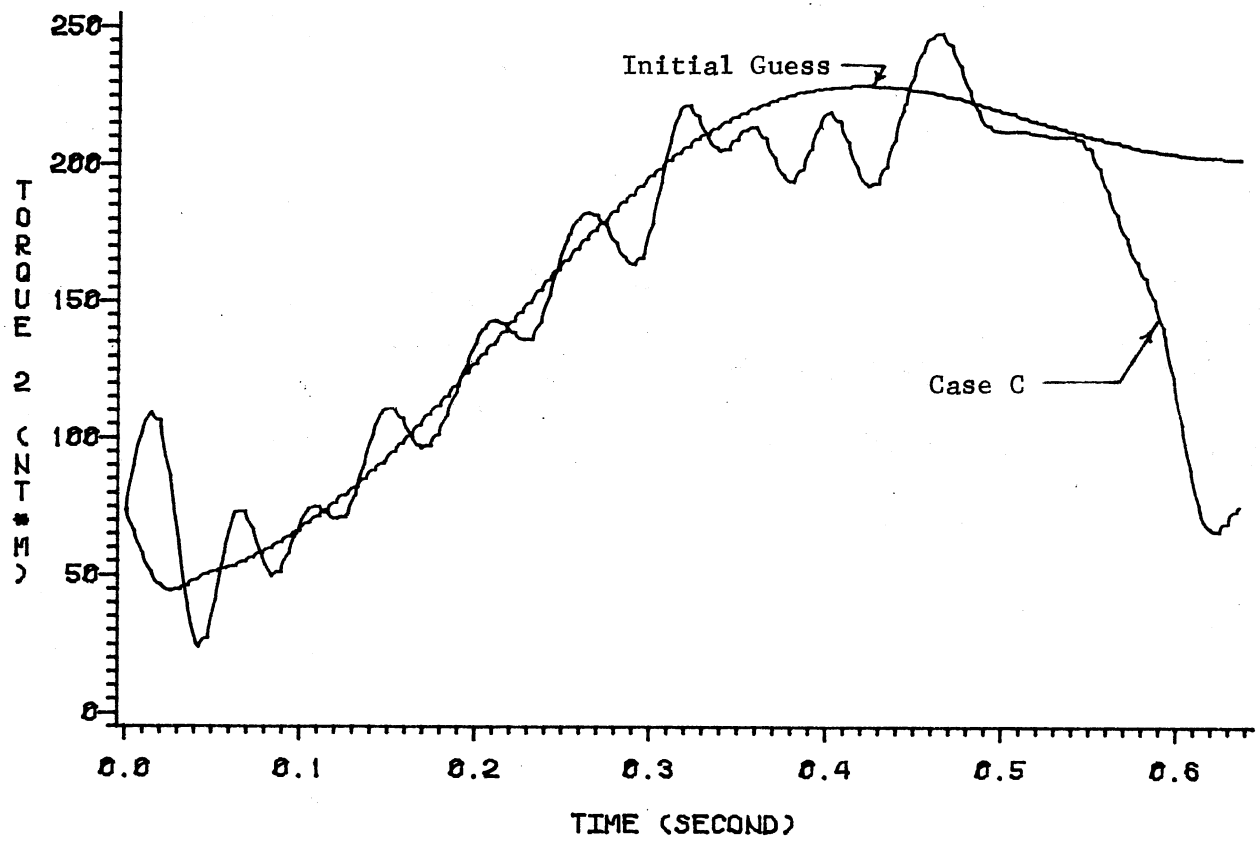

Figure 24. The Profiles of the Torques at the Second Joint for the Case C

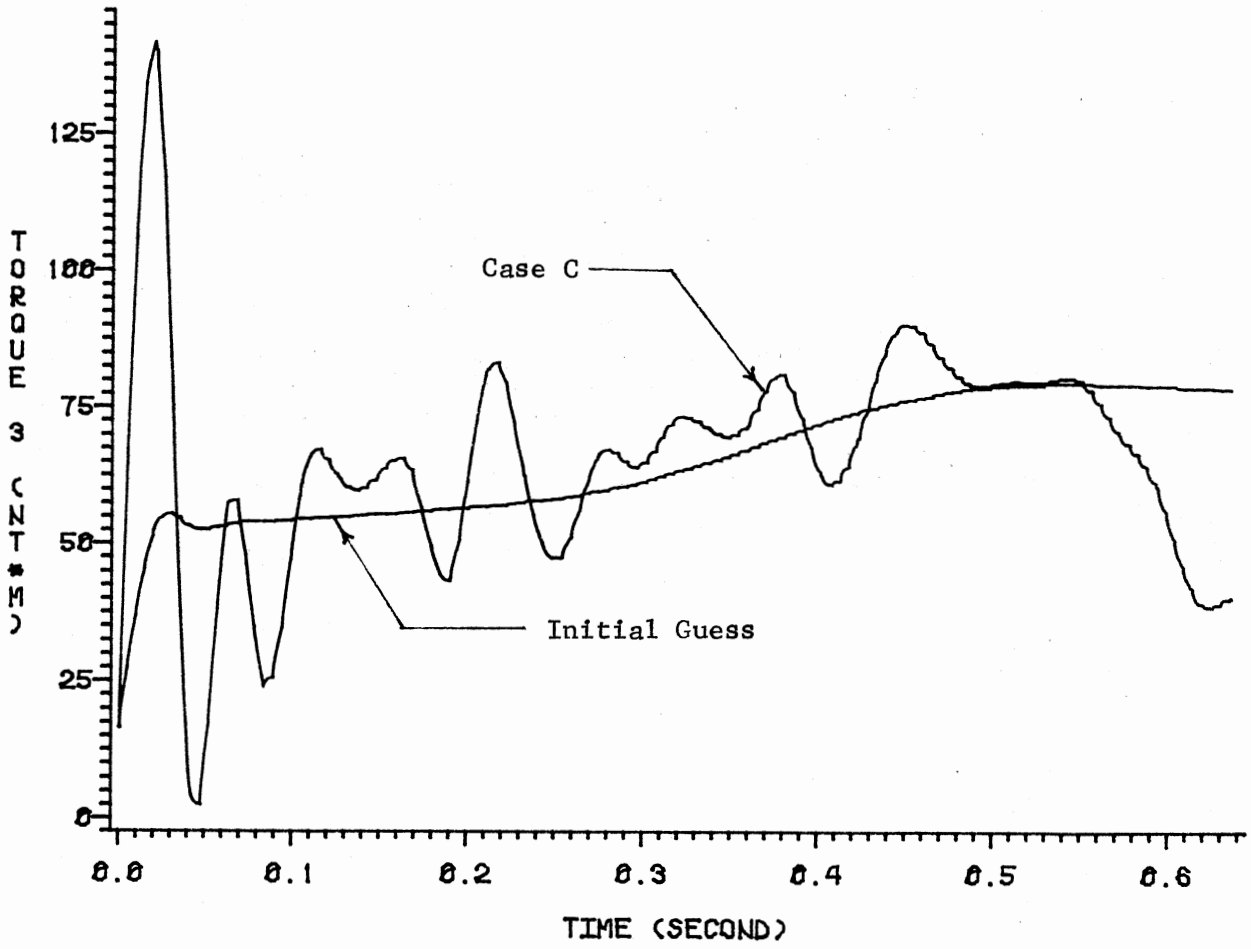

Figure 25. The Profiles of the Torques at the Third Joint for the Case C

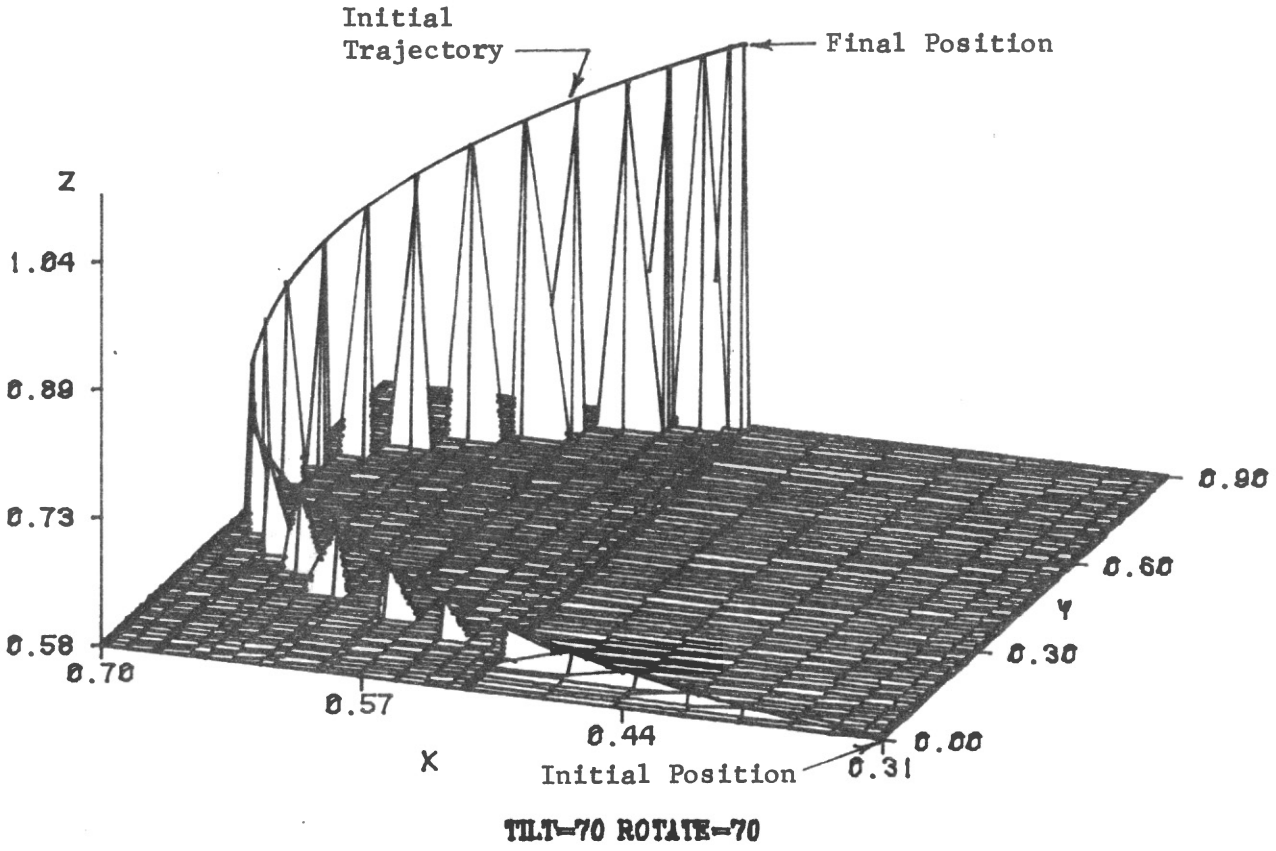

Figure 26. The Initial Guessed Trajectory

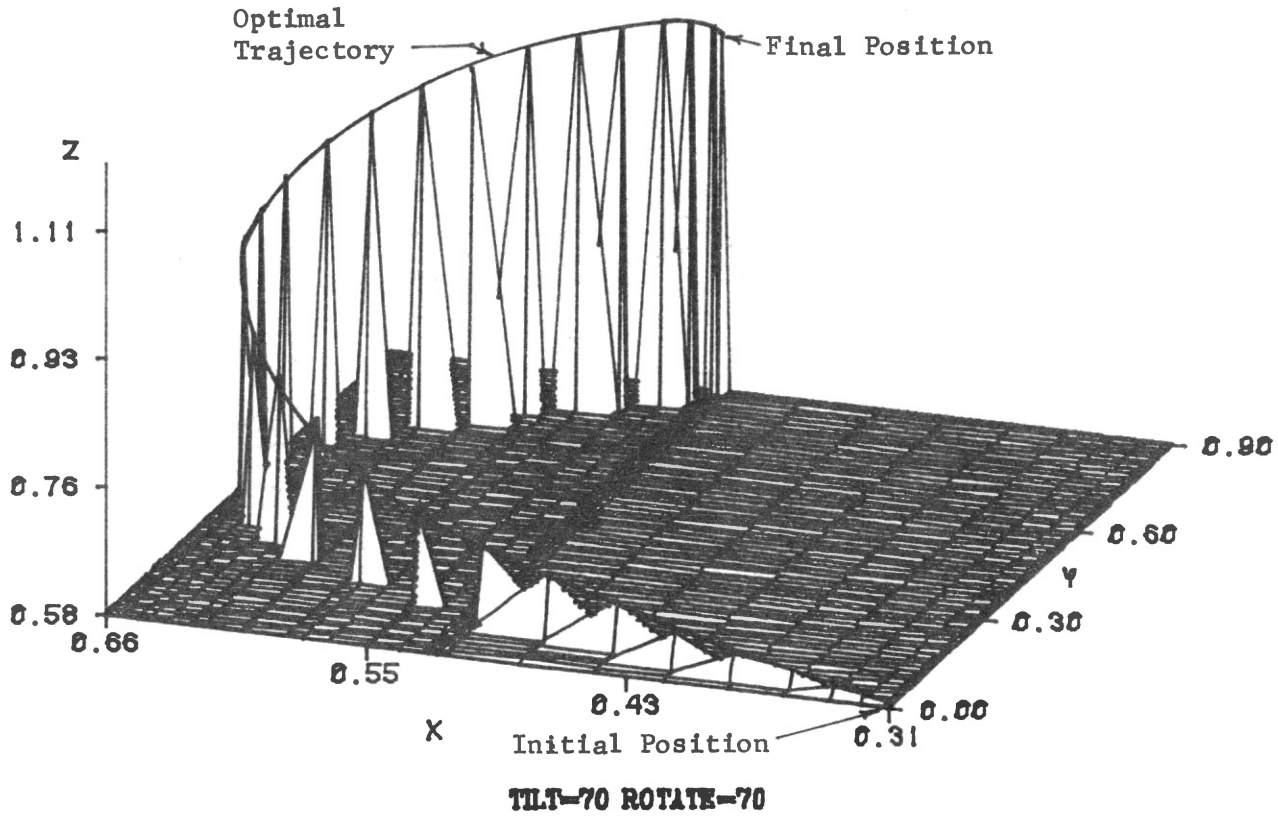

Figure 27. The Optimal Trajectory for the Case A

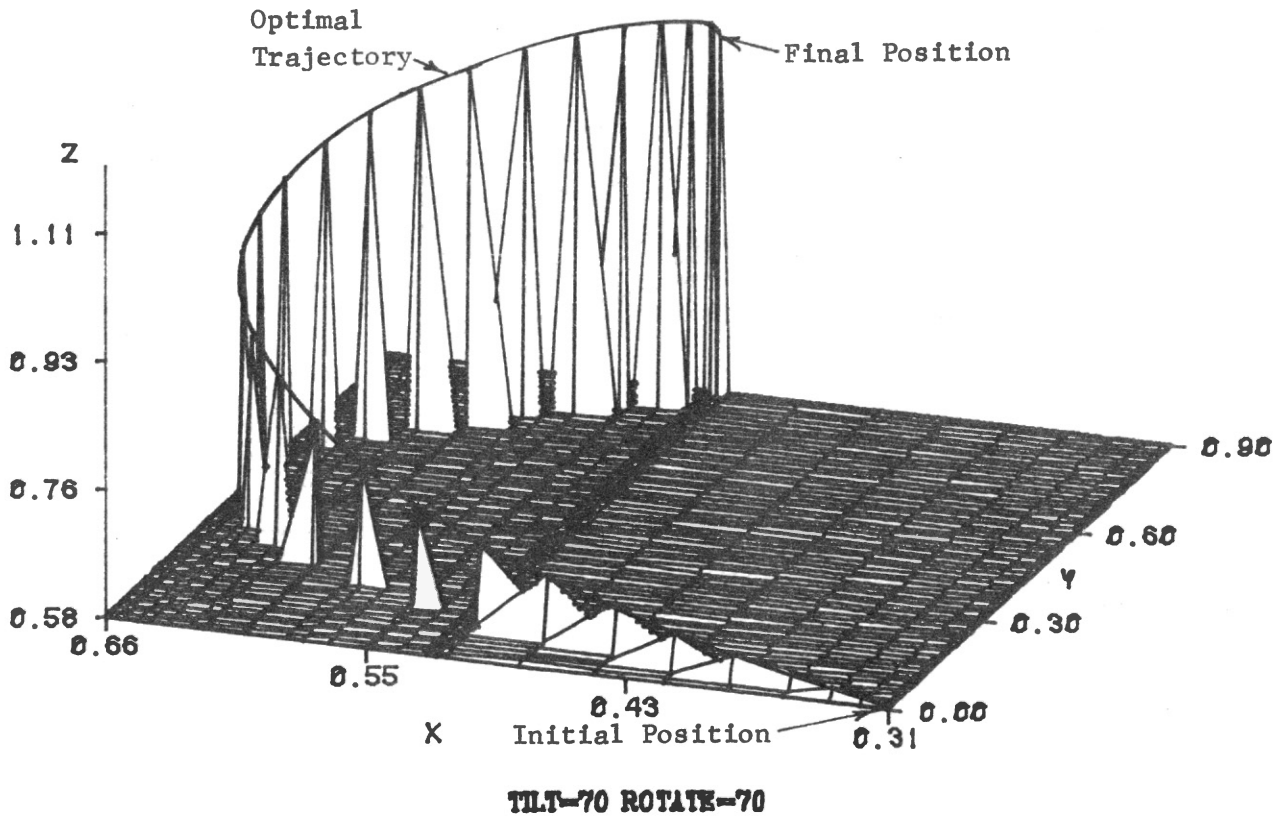

Figure 28. The Optimal Trajectory for the Case B

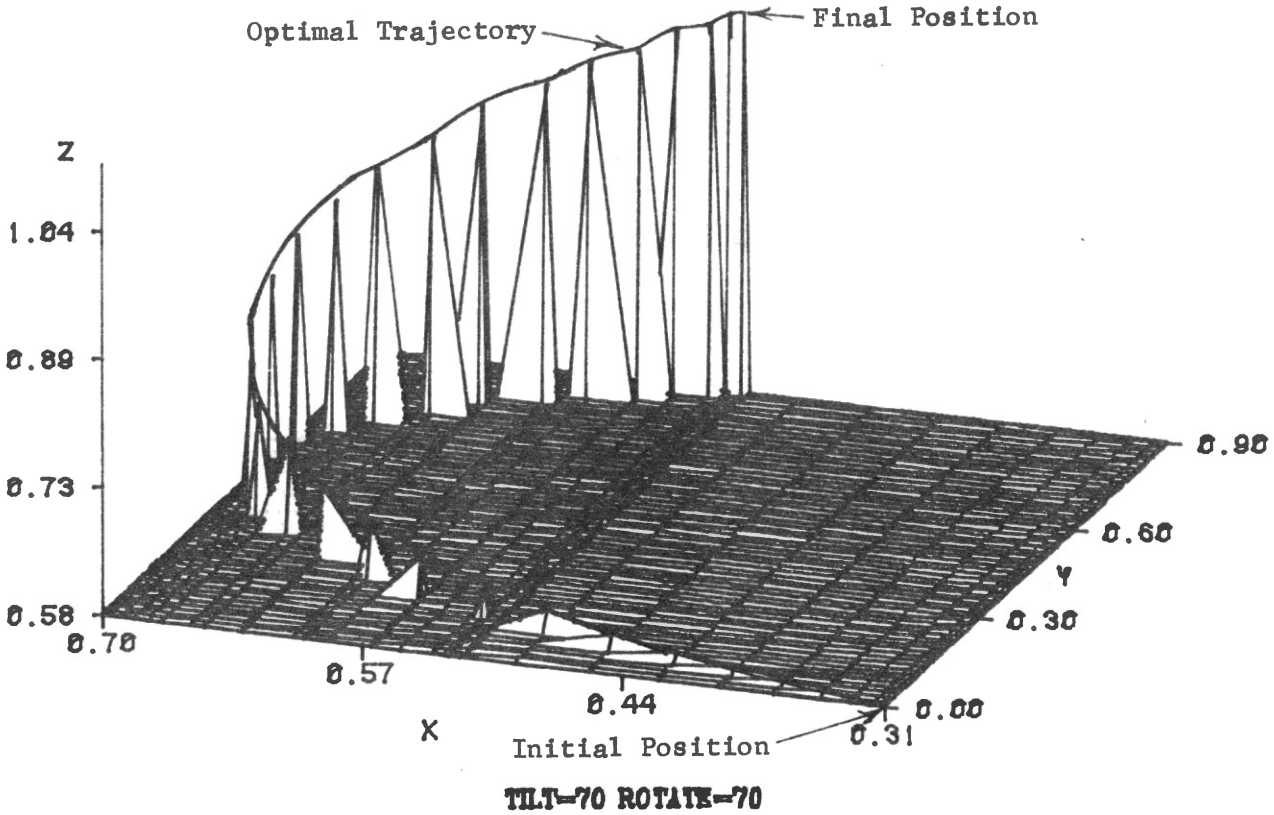

Figure 29. The Optimal Trajectory for the Case C

CHAPTER V

SUMMARY AND DISCUSSION

This research provides a general methodology for the synthesis problem of the optimal trajectory of n DOF robots. The problem was: Given the initial and the final joint angles of the robot, determine the optimal trajectory with certain criterion, such as the minimum energy consumption in this study, and under certain constraints, such as reachability and driving ability of actuators in this study.

A general algorithm based on the theories of variational calculus, Pontryagin's maximum principle and Bellman's principle of optimality was developed for solving the problem stated as above. From the basis of affine transformation, the method of Denavit and Hartenberg and Newton-Euler method, a systematic approach was developed to derive the explicit form of driving torques or forces which may include all of the dynamics of the robot system in addition to the viscous effect. The same approach can also be applied to the kinematic analysis for any n DOF robot, and implemented on a computer in a very efficient manner.

Essentially, the algorithm developed here is a kind of direct approach, or "brute force" approach. We broke the time spectrum into small time intervals, and assumed that the joints moves with constant accelerations in each small interval of time. From the principle of optimality and the maximum principle, the pattern search technique was used to optimize the trajectory, interval by interval, in a recursive

manner until the criteria of convergence was satisfied. By applying the principle of optimality and the maximum principle, we found that the realistic equations of motion and the realistic constrains can be easily incorporated into the procedures of solution.

The algorithm developed in this study is very simple, not only in the sense of algorithm itself but also in the sense of implementation on a computer. In addition, the problem of convergence is also avoided. Besides, it can also be implemented on-line, in real-time mode instead of off-line, since a learning of a new situation can be incorporated in arriving at a new trajectory. In the real cases, the initial feasible solution can be assigned by guiding the robot in the teaching mode, which is available for most of industrial robots. This feature is very helpful whenever the initial feasible solution such as obstacle avoidance is difficult to determine mathematically.

The most popular 3R robot was taken as an illustrated example to demonstrate the methodology developed. The results for this particular robot were presented and interpreted. The user could see the results in a manner of a 3-dimensional plot either on IBM 3279 terminal or Tektronix 4662 digital plotter. It is quite helpful to the user to realize the shape of the trajectory and then compare it with the physical case.

In this tudy, we consider only the characteristics of the robot itself. The characteristics of the actuators is beyond our scope, however, it is believed that even though the characteristics of actuators are incorporated, the same algorithm is still applicable. However, further investigation in this area is highly desirable.

BIBLIOGRAPHY

1. Angeles, J. Spatial Kinematic Chains: Analysis-Synthesis-Optimization. New York: Spring-Verlag, 1982.

2. Baumeister, T. (Chief Editor), Mark's Standard Handbook for Mechanical Engineers. New York: McGraw Hill, 1978.

3. Bellman, R. E. Dynamic Programming. New Jersey: Princeton University Press, 1957.

4. Bellman, R. E., and S. E. Dreyfus. Applied Dynamic Programming. New Jersey: Princeton University Press, 1962.

5. Bevridge, G. S., and R. S. Schechter. Optimization: Theory and Practice. New York: McGraw Hill, 1970.

6. Chow, C. K., and D. H. Jacobson. "Studies of Human Locomotion via Optimal Programming." Mathematical Bioscience, Vol. 28, 1976, pp. 99-135.

7. Hatze, H. "The Complete Optimization of a Human Motion." Mathematical Bioscience, Vol. 28, 1976, pp. 99-135.

8. Hollerbach, J. M. "A Recursive Formulation of Manipulator Dynamics and a Comparative Study of Dynamic Formulation Complexity." IEEE Transactions on System, Man, and Cybernetics, Vol. SMC-10, No. 11, Nov. 1980, 730-736.

9. Humang, T. C. Engineering Mechanics, Vol. 2. Massachusets: Addison-Wesley, Inc., 1973.

10. Kahn, M. E. and B. Roth. "The Near-Minimum-Time Control of Open-Loop Articulated Kinematics Chain." J. of Dynamic Systems, Measurements and Control, Transactions of ASME, Series G, Vol. 93, Sept. 1971, pp. 164-172.

11. Luh, J. Y. S., M. W. Walker and R. P. Paul. "Resolved Acceleration Control of Mechanical Manipulators." IEEE Transactions on Automatic Control, Vol. AC-25, No.3, Jan. 1980, pp. 468-474.

12. Luh, J. Y. S., M. W. Walker and R. P. Paul. "On-Line Computational Scheme for Mechanical Manipulators." J. of Dynamic Systems, Measurement, and Control, Transactions of ASME, Vol. 102, June, 1980, pp. 69-76.

13. Luh, J. Y. S., and C. S. Lin. "Optimum Path Planning for Mechanical Manipulators." J. of Dynamic Systems, Measurement, and Control, Transactions of ASME, Vol. 102, June, 1981, pp. 143-151.

14. Milenkovic, V. "Computer Synthesis of Continuous Path Robot Motion." Proceedings of the 5th World Congress on Theory of Machines and Mechanisms, 1979, pp. 1332-1335.

15. Naganathan, G. "Human Joint Motion Synthesis in Cyclic Weight-lifting." (Unpub. Research Study, Oklahoma State University, 1981).

16. Paul, R. P. "Manipulator Cartesian Path Control." IEEE Transactions on System, Man, and Cybernetics, Vol. SMC-9, No. 11, Nov. 1979, pp. 702-711.

17. Paul, R. P. Robot Manipulators: Mathematics, Programming, and Control. Massachusetts: MIT Press, 1981.

18. Pennock, G. R. and A. T. Yang "Dynamic Analysis of a Multi-Rigid-Body Open-Chain System." J. of Mechanical Design, Transactions of ASME, 82-DET-7, Dec. 1982.

19. Pieper, D. L. "The Kinematics of Manipulators Under Computer Control." Thesis, Oklahoma State University, 1981.) Stanford Artifical Intelligence Laboratory, Stanford University, AIM72, 1968.

20. Pierre, D. A. Optimization Theory with Application. New York: John Wiley and Sons, 1969.

21. Pontryagin, L. S. The Mathematical Theory of Optimal Processes. New York: The Macmillan Company, 1964.

22. Roth, B., J. Rastergar and V. Scheinman. "On the Design of Computer Controlled Manipulators." On Theory and Practice of Robots and Manipulators, New York: Springer-Verlag, 1974, pp. 39-62.

23. Roth, B. "Performance Evaluation of Manipulators from a Kinematics Viewpoint." Performance Evaluation of Programmable Robots and Manipulators, National Bureau of Standards, 1976, pp. 39-62.

24. Roth, B. "Robots." Applied Mechanics Reviews, Vol. 31, 1978, pp. 1511-1519.

25. Silver, W. M. "On the Equivalence of Lagarangian and Newton-Euler Dynamics for Manipulators." The International J. of Robotics Research, Vol. 1, No. 2., Summer 1982, pp. 60-70.

26. Soni, A. H., Mechanism Synthesis and Analysis. Florida: Robert E. Krieger Publishing Company, 1981.

27. Shimano, B. "The Kinematic Design and Force Control of Computer-Controlled Manipulators" (Ph.D. Dissertation, Stanford University, 1978.)

28. Tanner, W. R. Industrial Robots. Second Edition. Robotics International of SME, 1981.

29. Tsai, Y. C. "Synthesis of Robots/Manipulators for a Prescribed Working Space." (Ph.D. Dissertation, Oklahoma State University, 1982.)

30. Vukobratovic, M., V. Potkonjak, and D. Hristic. "Dynamic Method for the Evaluation and Choice of Industrial Manipulator." Proceedings of 9th International Symposium on Industrial Robots, Washington, D. C., 1979., pp. 549-561.

31. Vukobratovic, M., and M. Kircanski. "A Method for Optimal Synthesis of Manipulation Robot Trajectories." J. of Dynamic Systems, Measurement, and Control, Transactions of ASME, Vol. 104, June 1982, pp. 188-193.

32. Whitney, D. E. "The Mathematics of Coordinated Control of Proshtetic Arms and Manipulators." J. of Dynamic Systems, Measurement, and Control, Transactions of ASME, Dec. 1972. pp. 303-309.

33. Wilde, D. Optimum Seeking Methods. New Jersey: Prentice Hall, 1964.

34. "Japan, Corporate Strategies for the 80's." Businessweek, July 19, 1982, pp. 20-42.

APPENDIX

RELEVANT INFORMATION FOR THE ILLUSTRATED EXAMPLE

Followings are the inputs to the computer program written for the illustrative example in Chapter IV.

Initial Joint Angles (deg): (0, 70, 210)

Final Joint Angles (deg): (60, 45, 315)

Length of Each Link (m): (.610, .610, .610)

Weight of Each Link (kg): (12.02, 12.02, 12.02)

Radius of Each Joint (m): (.039, .039, .039)

Viscous Coefficient of Each Joint: (.2, .2, .2)

Center of Mass of Each Link (m): (.305, .305, .305)

Load (kg); 10

Number of Intervals: 30

Increment of Search Angle (deg): 1

No. of Reduction of Search Angle: 6

Convergence Criteria: .0001

VITA

JAMES YUNN-JIA WANG

Candidate for the Degree of

Master of Science

Thesis: A METHOD FOR THE SYNTHESIS OF OPTIMAL TRAJECTORY OF ROBOTS

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Jia-Yi, Taiwan, Republic of China, March 3, 1956, the son of Mr. and Mrs. Chieun Wang.

Education: Graduated from Taipei Municipal Chien-Kuo High School, Taipei, Taiwan, Republic of China, in June, 1974; received Bachelor of Science degree in Power Mechanical Engineering from National Tsing Hua University, Hsin Chu, Taiwan, Republic of China, in June, 1978; completed requirements for the Master of Science degree at Oklahoma State University in July, 1983.

Professional Experience: Project Engineer, Texas Instruments Taiwan Limited, Taipei, Taiwan, Republic of China, June, 1980 - July, 1981; Graduate Research Assistant, Agriculture Experiment Station, Oklahoma State University, Stillwater, Oklahoma, August, 1982 - present.