THE KINEMATIC ANALYSIS OF AN N-R ROBOT ARM

By

HASSAN ASLROUSTA

Bachelor of Science in Mechanical Engineering

Oklahoma State University

Stillwater, Oklahoma

1980

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, 1982

Name: Hassan Aslrousta Date of Degree: July, 1982

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: THE KINEMATIC ANALYSIS OF AN N-R ROBOT ARM

Pages in Study: 60 Candidate for Degree of Master of Science

Major Field: Mechanical Engineering

Scope and Method of Study: This study reports on the displacement, velocity, and acceleration analysis of an n-r robot arm. The equations of the displacement, velocity, and acceleration of the hand center point were derived as a function of angular displacement, angular velocity, and angular acceleration of each joint in the arm. Different motion programs were used to provide the angular displacement, angular velocity, and angular acceleration of each joint as a function of time. An interactive computer program was developed to plot the displacement, velocity, and acceleration components of the hand as a function of time. Considering that all the joints may not make a complete rotation, a choice of oscillation angle for each link was embedded in the program.

Findings and Conclusions: The method of transformation matrices developed by Denavit and Hartenberg was found to have simplified the analysis of space mechanisms a great deal, especially in the analysis of robot arms. Transformation matrix method was used in this study to simplify the analysis. This analysis can be applied to any robot arm with general skew axis/robot arms with special geometry in order to determine the displacement, velocity, and acceleration characteristics of the hand center point of the robot arm. By knowing the displacement, velocity, and acceleration of the hand, one can simply find the displacement, velocity, and acceleration of any point on an object being held by the hand assuming the object is rigid. This research if applied to a specific robot arm can be used to compare the theoretical and actual results, therefore the difference between the components of actual and theoretical displacement, velocity, and acceleration of the hand can be traced continuously.

ADVISER'S APPROVAL Afmaram H. Soni

THE KINEMATIC ANALYSIS OF AN

N-R ROBOT ARM

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

PREFACE

I would like to express my appreciation to my major adviser, Dr. Atmaram H. Soni, for his guidance and assistance, and particularly for his guidance through useful references. Appreciation is also extended to the other committee members, Dr. Richard L. Lowry and Dr. J. J. Allen, for their assistance in the preparation of the final manuscript.

I would also like to acknowledge the invaluable assistance provided by Mr. Naga Nadhan in all phases of this study. A note of thanks is given to Dr. Y. C. Tsai and Dr. Kwun L. Ting for their helpful suggestions. In addition, appreciation is extended to Miss Kelli L. Winfree for her understanding, encouragement, and assistance in typing this thesis.

Finally, gratitude is expressed to the School of Mechanical and Aerospace Engineering for providing a teaching assistantship which made this study possible.

TABLE OF CONTENTS

LIST OF FIGURES

Figure | Page

LIST OF SYMBOLS

a_i	Length of Common Perpendicular from Z_i to Z_{i+1}
α_i	Angle from Positive Z_i to Positive Z_{i+1}
θ_i	Angle from Positive X_i to Positive X_{i+1}
s_i	Distance Along Joint Axes Z_i from X_i to X_{i+1}
A_i	A 4 X 4 Displacement Matrix
ϕ	Vertical Angle of the Extreme Reach
γ	Horizontal Angle of the Extreme Reach
R	Distance from the base to the center of the hand
Q	The Derivitive Operator Matrix

CHAPTER I

INTRODUCTION

An open loop kinematic chain with one end fixed to ground and a free end where there is a mechanical hand or gripping device with multi-degree of freedom of motion is considered as the simplest structure of robot arms/manipulators [12]. Manipulators are assemblies of links which are assumed to be rigid members and kinematic pairs or joints which permit relative motion between successive links. A major difference between manipulators and other machines, designed to do one specific task, is that manipulators are capable of doing different tasks by simply reprogramming them [1].

Industrial robots have been avilable for 17 years-providing over 4 million hours of robot labor, worldwide, to date [11]. Thus far primary applications have been press loading and unloading, spot welding and spray painting. Through computer-control and the addition of visual and tactile sensors to these robots, the range of applications can be broadened considerably. Advaned design robots will perform arc welding and more complex work handling.

In a closed loop kinematic chain the most difficult problem is that of displacement analysis; given the relative displacements, velocities, and accelerations of the kinematic pairs determine the absolute displacements, velocities, and accelerations of any point of the system. This problem has been studied in depth and many approaches are known for its solution. However, the aim of this research is to extend this analysis to robot arms. There are some fundamental differences between the kinematic analysis of closed loop mechanisms and of manipulators. One major distinction is that a manipulator is usually an open loop system. Since the chain is not closed there are no kinematic dependencies among the joint variables and all are independnet degrees of freedom of the system.

Proposed Research

The overall aim of the research is to provide a proper foundation for the kinematic analysis of robot arms. Such a study will reduce the time and hence the considerable expense involved in the development and control of present and especially future manipulators.

It is intended to develop an algorithm and an efficient computer program for displacement, velocity, and acceleration analysis of all N-R robot arms with general proportions by solving the geometry rather than employing iterative techniques. Such a facility will include as special cases the analysis of robot arms with special proportions, for example those with intersecting and parallel axes. It is considered that the research will provide an analytical foundation for the design and operation of existing and especially future robot arms. The research can be used to compare the performance of a wide variety of robot arms and thus hopefully, produce answers to the difficult problems of selecting the structure and size to produce robot arms with assurance that the resulting machine will meet the required specifications. The central problem in the kinematic analysis of robot arms is the derivation of input-output displacement, velocity and acceleration equations [15], i.e. it is necessary to derive trigonometrical equations relating the input angular displacement, velocity, and acceleration at each joint to the output absolute displacement, velocity, and acceleration components of the robot hand. The input data may be supplied either in an analytical or in numerical form. Analytical data may be supplied [3] by choosing any one of several motion programs for any one of the joints. Numerical data [13] may be supplied in the form of angular displacement vs time, angular velocity vs time, or angular acceleration vs time for any number of joints. The output will be the plots of displacement, velocity, and acceleration components of the robot hand. At the end, an illustrative example will be provided to demonstrate the application of the computer program.

CHAPTER II

METHODOLOGY

Displacement analysis in kinematics has been the objective of a number of studies, such as the vector methods developed by Chace, the transformation matrix method introduced by Denavit and Hartenberg and extended by Uicker et al. [2]. The transformation matrix method developed by Denavit and Hartenberg has simplified the analysis of space mechanisms a great deal, especially in the analysis of robot arms. The method of transformation matrices is used here to simplify this analysis.

Dimensions and Configurations of a Link

The dimensions and configurations of any link in space are determined by sets of parameters a_i, α_i, θ_i, s_i as shown in Figure 1.

Where

i = number of particular joint or pair

z_i = characteristic axis of motion for pair involved

x_i = axis formed by common perpendicular directed from z_{i+1} to z_i

y_i = axis implicitly defined to form a right-handed Cartesian coordinate system $x_i y_i z_i$

a_i = length of common perpendicular from z_i to z_{i+1}; always positive

α_i = angle from positive z_i to positive z_{i+1}, measured counterclockwise about positive x_{i+1}, often called twist angle.

θ_i = angle from positive x_i to positive x_{i+1}, measured counterclockwise about positive z_i

s_i = distance along joint axes z_i from x_i to x_{i+1}. It takes sign from orientation of positive z_i.

Once the four parameters a_i, α_i, θ_i, and s_i have been established for each link of a linkage, the geometry of the linkage is completely specified. The coordinate transformation from the system $x_{i+1}y_{i+1}z_{i+1}$ fixed in joint i+1 to the system $x_iy_iz_i$ fixed in joint i is defined by the 4x4 matrix:

$$A_i = \begin{matrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{matrix} \quad (1)$$

The relative change in oreientation and position from $(x_{i+1}, y_{i+1}, z_{i+1})$ to (x_i, y_i, z_i) can be defined as:

$$\{\bar{x}_i\} = A_i \{\bar{x}_{i+1}\} \quad (2)$$

Where

$$x_i = (x_i, y_i, z_i, 1)^T \quad (3)$$

And

$$\bar{x}_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1}, 1)^T \quad (4)$$

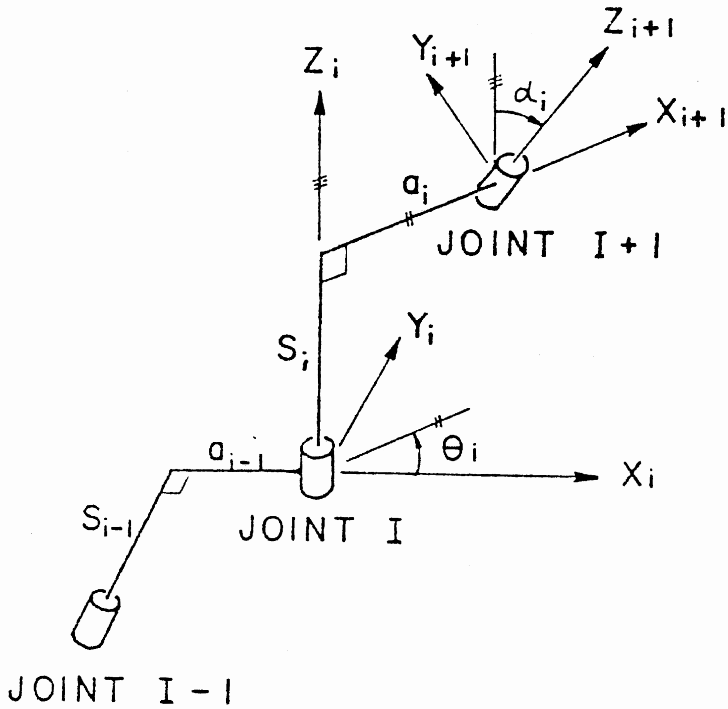

Figure 1. Summary of Symbolic Notation

For an n-r robot arm coordinate transformation may be defined as:

$$\{\bar{X}_1\} = A_1n \quad \{\bar{X}_{n+1}\} \tag{5}$$

Where

$$A_{1n} = A_1A_2A_3 \quad . \; . \; . \; . \; . \; . \; A_n \tag{6}$$

Hand Displacement of the Robot Arm

The coordinate of the hand center point relative to the base coordinate system of n-r robot arm may be defined as:

$$\bar{X} = T_n\bar{X}_h \tag{7}$$

Where $\bar{X}_h = (X_h, Y_h, Z_h, 1)^T$ is the coordinate of the hand center point relative to a coordiante system fixed at the center point of the hand. If the coordinate systems are defined according to the format outlined here, then:

$$\bar{X}_h = (0, 0, 0, 1)^T \tag{8}$$

$\bar{X} = (X, Y, Z, 1)^T$ is the coordinate of the hand center point relative to the base coordinate system XYZ, and T_n is a numerical matrix defined as:

$$T_n = A_1A_2A_3 \; . \; . \; . \; . \; A_n \tag{9}$$

Therefore, the instantaneous position of the hand may be found from these matrix paroducts. They are the displacement matrices for the moving hand. The position of the hand would be completely known if we find γ and ϕ, horizontal and vertical angles of the extreme reach R, respectively. These angles are shown in Figure 2 and they may be found by the following equations:

$$\phi = \sin^{-1}(Z/R) \tag{10}$$

$$\gamma = \tan^{-1}(Y/X) \tag{11}$$

Where

$$R = (X^2 + Y^2 + Z^2)^{1/2} \tag{12}$$

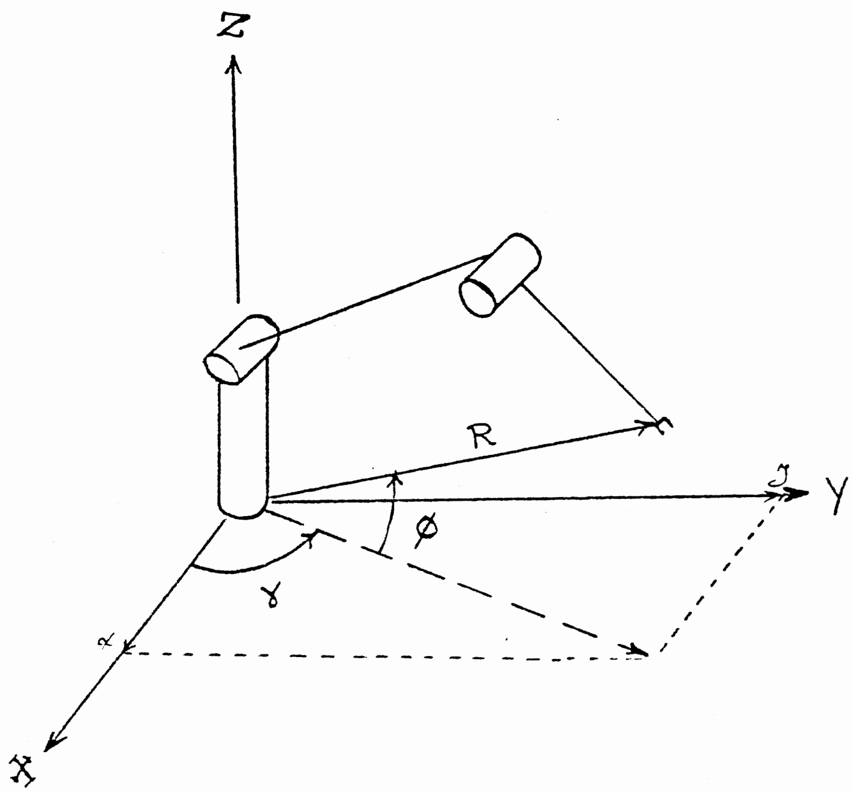

Figure 2. Position of the Hand

To determine the velocity and acceleration of the hand we need to differentiate matrices $A_1, A_2, \ldots, A_n$ and noting that this problem is to be adapted to computer operation a linear operator matrix Q_i will be introduced to perform the differentiation through the following definition:

$$\frac{\partial A_i(\theta_i)}{\partial \theta_i} = Q_i A_i \qquad (13)$$

Under the definition, the Q_i matrix is found to be:

$$Q_i = \begin{matrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \qquad (14)$$

The derivitive of matrix A_i relative to time may be defined as:

$$\frac{\partial A_i(\theta_i)}{\partial t} = Q_i A_i \dot{\theta}_i \qquad (15)$$

where $\dot{\theta}_i = \frac{d\theta_i}{dt}$ is defined as the angular velocity of link i relative to link i - 1.

Hand Velocity of the Robot Arm

The absolute velocity components $\dot{x} = \frac{dx}{dt}$, $\dot{y} = \frac{dy}{dt}$, $\dot{z} = \frac{dz}{dt}$ of the hand relative to the base point may be found by differentiating equation (7). Letting

$$\dot{\bar{X}} = \frac{d\bar{x}}{dt} = \begin{matrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ 0 \end{matrix} \tag{16}$$

We have

$$\dot{\bar{X}} = Q_1A_1A_2 \ . \ . \ . \ A_i \ . \ . \ . \ A_n\bar{X}_h \dot{\theta}_1 + A_1Q_2A_2 \ . \ . \ . \ A_1 \ . \ . \ . \ A_nX_h\dot{\theta}_2 + \ . \ . \ . \ + A_1 \ . \ . \ . \ A_{i-1}Q_iA_i \ . \ . \ . \ A_n\bar{X}_h\dot{\theta}_i + . \ . \ . \ + A_1A_2 . \ . \ . \ A_{n-1}Q_nA_nX_h\dot{\theta}_n \tag{17}$$

or

$$\dot{\bar{X}} = \{B_1\dot{\theta}_1 + B_2\dot{\theta}_2 + . \ . \ . \ + B_i\dot{\theta}_i + \ . \ . \ . \ + B_n\dot{\theta}_n\} \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix} \tag{18}$$

Where

$$B_i = A_1A_2 \ . \ . \ . \ A_{i-1}Q_iA_i \ . \ . \ . \ A_n \tag{19}$$

The instantaneous absolute velocity components $\dot{x}$, $\dot{y}$, and $\dot{z}$ of the hand may be found from equation (18).

Hand Acceleration of the Robot Arm

The absolute acceleration components $\ddot{x} = \frac{d^2x}{dt^2}$, $\ddot{y} = \frac{d^2y}{dt^2}$, $\ddot{z} = \frac{d^2z}{dt^2}$ of the hand relative to the base point may be found by differentiating equation (18). Letting

$$\ddot{\bar{X}} = \frac{d^2X}{dt^2} \begin{matrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ 0 \end{matrix} \tag{20}$$

We have

$$\ddot{\bar{X}} = \{(B_1\ddot{\theta}_1 + B_2\ddot{\theta}_2 + \ldots + B_i\ddot{\theta}_i + \ldots + B_n\ddot{\theta}_n) + (\dot{B}_1\dot{\theta}_1 + \dot{B}_2\dot{\theta}_2 + \ldots + \dot{B}_i\dot{\theta}_i + \ldots + \dot{B}_n\dot{\theta}_n)\} \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix} \quad (21)$$

Where

$$\dot{B}_i = \frac{dB_i}{dt} = \frac{d}{dt}(A_1A_2 \ldots A_{i-1}Q_iA_i \ldots A_n) \quad (22)$$

$$\dot{B}_i = Q_1A_1 \ldots A_{i-1}Q_iA_i \ldots A_n\dot{\theta}_1 + A_1Q_2A_2 \ldots A_{i-1}Q_iA_i \ldots A_n\dot{\theta}_2 + \ldots + A_1 \ldots A_{i-2}Q_{i-1}A_{i-1}Q_iA_i \ldots A_n\dot{\theta}_{i-1} + A_1 \ldots A_{i-1}Q_iQ_iA_i \ldots A_n\dot{\theta}_i + \ldots + A_1 \ldots A_{i-1}Q_iA_i \ldots A_{n-1}Q_nA_n\dot{\theta}_n \quad (23)$$

Computer Program

It is intended to develop an efficient interactive computer program for displacement, velocity, and acceleration analysis of all n-R robot arms with general proportions by solving the geometry rather than employing iterative techniques. Inputs of the program are the link parameters a_i, α_i, s_i, θ_i, $\dot{\theta}_i$, $\ddot{\theta}_i$, a_i, α_i, and s_i are constants, but θ_i, $\dot{\theta}_i$, and $\ddot{\theta}_i$ are variables. These variables may be supplied either in an analytical or in numerical form. Analytical data may be supplied by choosing any one of several motion programs for each joint in the arm. These moition programs are:

1. Constant Velocity
2. Simple Harmonic
3. Modified Harmonic
4. Cycloidal
5. 3-4-5 Polynomial
6. Cubic Curve Type 2

The above motion programs have been selected and modified from [3]. Numerical data may be supplied in the form of angular displacement, angular velocity, or angular acceleration. The following summary includes the characteristics of the above motion programs. The notations used throughout these motion programs include:

θ = joint angular displacement, rad.

θ_0 = initital joint angular displacement, rad.

θ_f = final joint angular displacement, rad.

$\Delta\theta$ = $\theta_f - \theta_0$, maximum joint angular displacement, rad.

t = time for joint angular displacement of , sec.

T = time for maximum joint angular displacement $\Delta\theta$, sec.

$\dot{\theta}$ = relative angular velocity, rad./sec.

$\ddot{\theta}$ = relative angular acceleration, rad./sec^2.

Constant Velocity Motion Program

The joint angular motion of a constant velocity motion program is plotted in Figure 3. Angular displacement, velocity, and acceleration of a link or a joint may be found from the following equations:

$$\theta = \theta_0 + \Delta\theta\left(\frac{t}{T}\right)$$

$$\dot{\theta} = \frac{\Delta\theta}{T}$$

$$\ddot{\theta} = 0.0$$

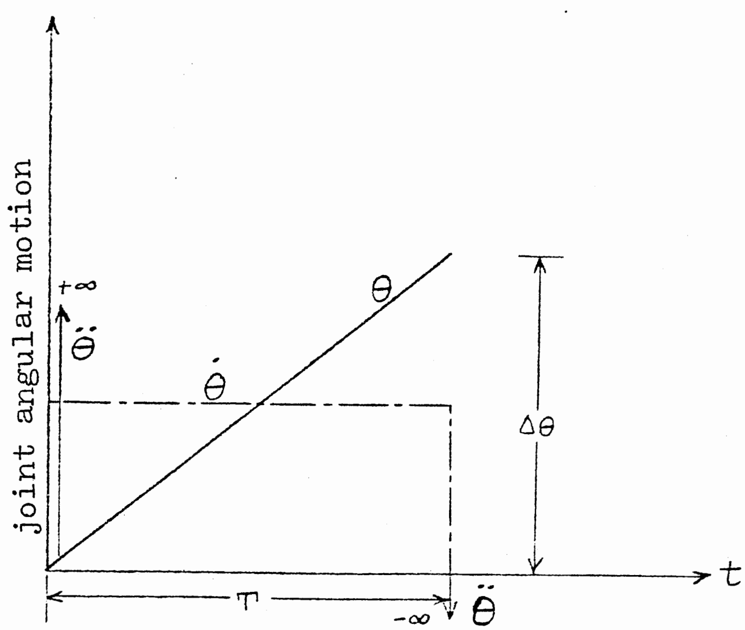

Figure 3. Constant Velocity Motion Program

Simple Harmonic Motion Program

The joint angular motion of a simple harmonic motion program is plotted in Figure 4. The equations for joint angular displacement, velocity, and acceleration may be defined as

$$\theta = \theta_0 + \frac{\Delta\theta}{2} \left(1 - \cos\frac{\pi t}{T}\right) \quad (27)$$

$$\dot{\theta} = \frac{\Delta\theta\pi}{2T} \sin\frac{\pi t}{T} \quad (28)$$

$$\ddot{\theta} = \frac{\Delta\theta}{2} \left(\frac{\pi}{T}\right)^2 \cos\frac{\pi t}{T} \quad (29)$$

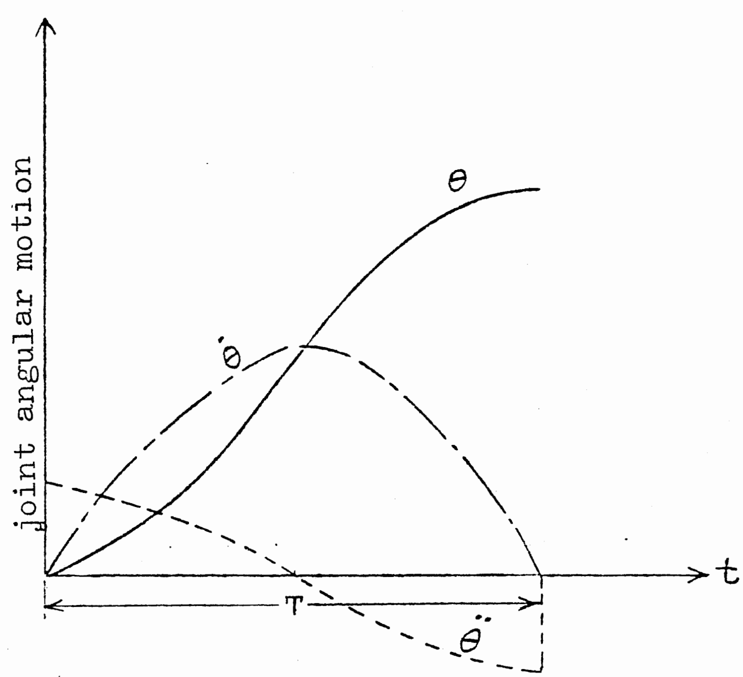

Figure 4. Simple Harmonic Motion Program

Modified Harmonic Motion Program

The joint angular motion of a modified harmonic motion program is plotted in Figure 5. The equations for joint angular displacement, velocity, and acceleration may be defined as

$$\theta = \theta_0 + \frac{\Delta\theta}{2}\left(1-\cos\frac{\pi t}{T}\right) - 1/4\left(1 - \cos\frac{2\pi t}{T}\right) \tag{30}$$

$$\dot{\theta} = \frac{\Delta\theta\pi}{2T}\left(\sin\frac{\pi t}{T} - 1/2\sin\frac{2\pi t}{T}\right) \tag{31}$$

$$\ddot{\theta} = \frac{\Delta\theta}{2}\left(\frac{\pi}{T}\right)^2\left(\cos\frac{\pi t}{T} - \cos\frac{2\pi t}{T}\right) \tag{32}$$

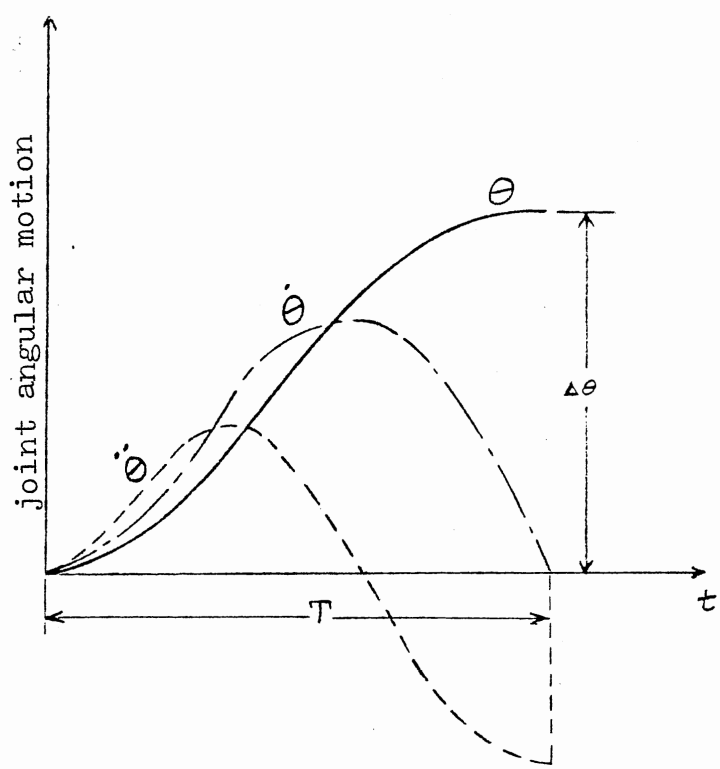

Figure 5. Modified Harmonic Motion Program

Cycloidal Motion Program

The joint angular motion of a cycloidal motion program is plotted in Figure 6. The equations of joint angular displacement, velocity, and acceleration may be defined as

$$\theta = \theta_0 + \frac{\Delta\theta}{\pi} \left(\frac{\pi t}{T} - 1/2 \sin \frac{2\pi t}{T} \right) \tag{33}$$

$$\dot{\theta} = \frac{\Delta\theta}{T} \left(1 - \cos \frac{2\pi t}{T} \right) \tag{34}$$

$$\ddot{\theta} = \frac{2\Delta\theta\pi}{T^2} \sin \frac{2\pi t}{T} \tag{35}$$

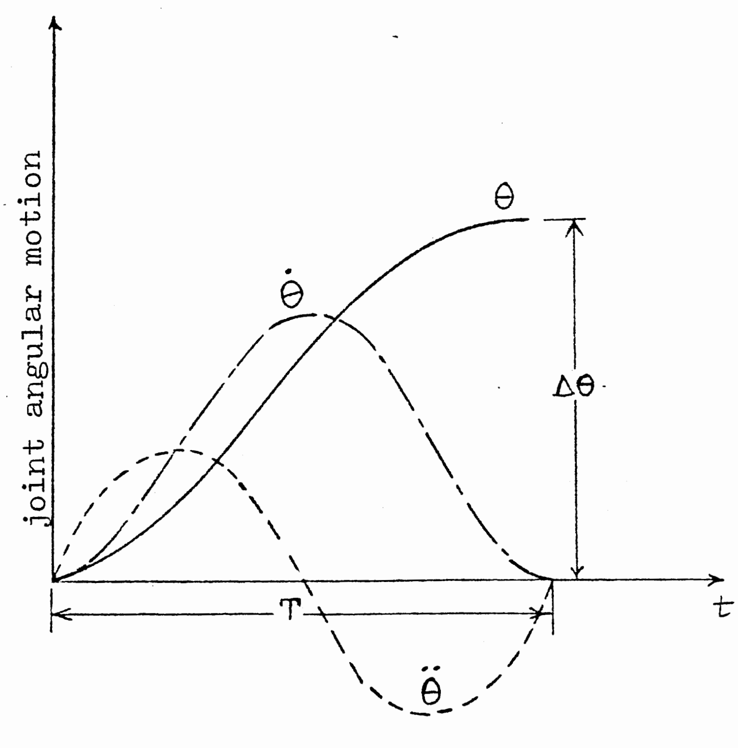

Figure 6. Cycloidal Motion Program

3-4-5 Polynomial Motion Program

The joint angular motion of 3-4-5 polynomial motion program is plotted in Figure 7. The equations of joint angular displacement, velocity, and acceleration may be defined as

$$\theta = \theta_0 + \frac{10\Delta\theta}{T^3} t^3 - \frac{15\Delta\theta}{T^4} t^4 + \frac{6\Delta\theta}{T^5} t^5 \quad (36)$$

$$\dot{\theta} = \frac{30\Delta\theta}{T^3} t^2 - \frac{60\Delta\theta}{T^4} t^3 + \frac{30\Delta\theta}{T^5} t^4 \quad (37)$$

$$\ddot{\theta} = \frac{60\Delta\theta}{T^3} t - \frac{180\Delta\theta}{T^4} t^2 + \frac{120\Delta\theta}{T^5} t^3 \quad (38)$$

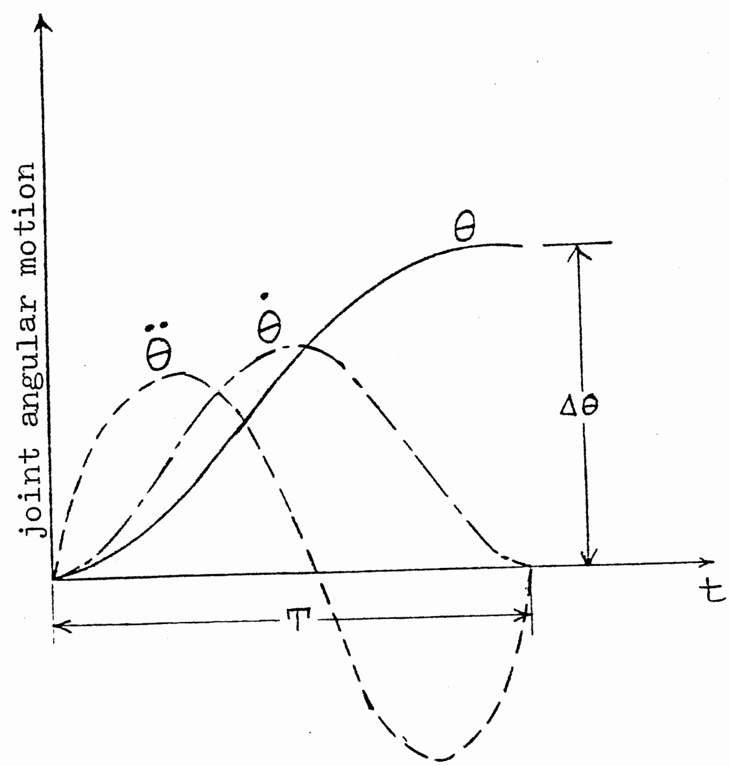

Figure 7. 3-4-5 Polynomial Motion Program

Cubic Curve Motion Program (Type 2)

The joint angular motion of a cubic motion program type 2 is plotted in Figure 8. The equations of joint angular displacement, velocity, and acceleration may be defined as

$$\theta = \theta_0 + \Delta\theta \frac{t^2}{T^2} \left(3 - \frac{2t}{T}\right) \tag{39}$$

$$\dot{\theta} = \frac{6\Delta\theta t}{T^2} \left(1 - \frac{t}{T}\right) \tag{40}$$

$$\ddot{\theta} = \frac{6\Delta\theta}{T^2} \left(1 - \frac{2t}{T}\right) \tag{41}$$

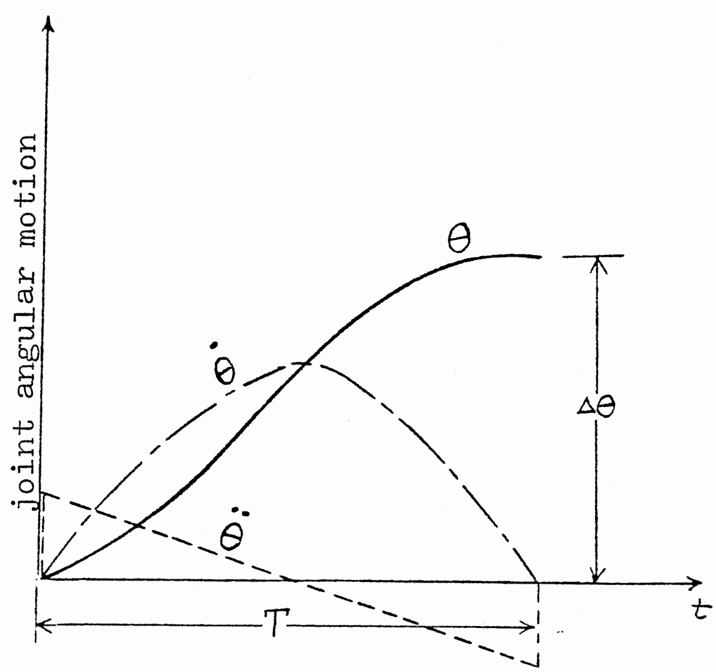

Figure 8. Cubic Curve (Type 2) Motion Program

The following is a summary of the numerical methods used in this study to differentiate/integrate the input data if supplied in a numerical form in order to provde the angular velocity and angular acceleration/angular displacemetn and angular velocity of a particular joint. These numerical methods are Simpson's integration method and Newton's differentiation method.

Simpson's Integration Method

Simpson's rule is a numerical integration technique which is based on the use of parabolic arcs to approximate a function. Using the Taylor series expansions [14] the area of the two panels between x_{j-1} and x_{j+1} may be found from

$$I(X_{j+1}) - (X_{j-1}) = \frac{\Delta X}{2} [f(X_{j-1}) + 4f(X_j) + f(X_{j+1})] - \frac{(\Delta X)^5}{90} f^{iv}(X_j) + \text{Higher Order Terms} \quad (42)$$

Equation (42) is used to find angular displacement and angular velocity from a supplied numerical data in the form of angular acceleration vs time.

Newton's Differentiation Method

Newton's forward, backward, and central difference relationships are used to differentiate a supplied numerical data in the form of angular displacement vs time in order to provide the angular velocity and angular acceleration of a particular joint. Newton's forward and backward difference formulas approximating the first and second derivatives with up to second order terms of the Taylor series used are, respectively

$$f'(X_i) = \frac{-f(X_i + 2\Delta X) + 4f(X_i + \Delta X) - 3f(X_i)}{2\Delta X}$$

$$f''(X_i) = \frac{2f(X_i) - 5f(X_i + \Delta X) + 4f(X_i + 2\Delta X) - f(X_i+3\Delta X)}{\Delta X^2}$$

$$f'(X_i) = \frac{3f(X_i) - 4f(X_i - \Delta X) - f(X_i - 2\Delta X)}{2\Delta X}$$

$$f''(X_i) = \frac{2f(X_i) - 5f(X_i - \Delta X) + 4f(X_i - 2\Delta X) - F(X_i-3\Delta X)}{\Delta X^2}$$

Newton's central difference formulas up to fourth order term of the Taylor series were used to differentiate the data associated with the midpoint of a range. The first and second derivatives of the central data may be found from the following equations, respectively

$$f'(X_i) = \frac{f(X_i - 2\Delta X) - 8f(X_i - \Delta X)+8f(X_i + \Delta X) - f(X_i +2\Delta X)}{12\Delta X}$$

$$f''(X_i) =$$

$$\frac{-f(X_i - 2\Delta X) + 16f(X_i - \Delta X) - 30f(X_i) +16f(X_i +\Delta X)-f(X_i+2\Delta X)}{12\Delta X^2}$$

Since there is a lot of matrix multiplications in the algorithm we are limited with computer memory space and this limits the data points to about two hundred and fifty (this result is obtained by trying to supply more than 250 data points) when numerical data are supplied. To overcome this problem, one may run the program in a computer with bigger memory.

CHAPTER III

RESULTS

The following example illustrates the hand displacement, velocity, acceleration of a 6-R robot arm. The link parameters arbitrarily chosen are as follows:

$a_1 = 0.0$ in.	$s_1 = 0.0$	$\alpha_1 = 90.0$ degrees
$a_2 = 30.0$ in.	$s_2 = 0.0$	$\alpha_2 = 0.0$ degrees
$a_3 = 30.0$ in.	$s_3 = 0.0$	$\alpha_3 = 0.0$ degrees
$a_4 = 4.0$ in.	$s_4 = 0.0$	$\alpha_4 = 90.0$ degrees
$a_5 = 2.0$ in.	$s_5 = 0.0$	$\alpha_5 = 90.0$ degrees
$a_6 = 1.0$ in.	$s_6 = 0.0$	$\alpha_6 = 0.0$ degrees

Figure 10 shows a robot arm with the above dimensions. The range of oscillation of the joint angles arbitrarily chosen are:

$\theta_1 = 0$ to 360 degrees

$\theta_2 = 0$ to 90 degrees

$\theta_3 = -120$ to 0 degrees

$\theta_4 = -90$ to 90 degrees

$\theta_5 = -120$ to +120 degrees

$\theta_6 = 0$ to 360 degrees

The motion programs used for joints angular motion are:

Joint 1 : Cycloidal

Joint 2 : Cubic Curve Type 2

Joint 3 : Modified Harmonic

Joint 4 : 3-4-5 Polynomial

Joint 5 : Simple Harmonic

Joint 6 : Arbitrarily Chosen Data

The results are ploted in Figure 10 through 24.

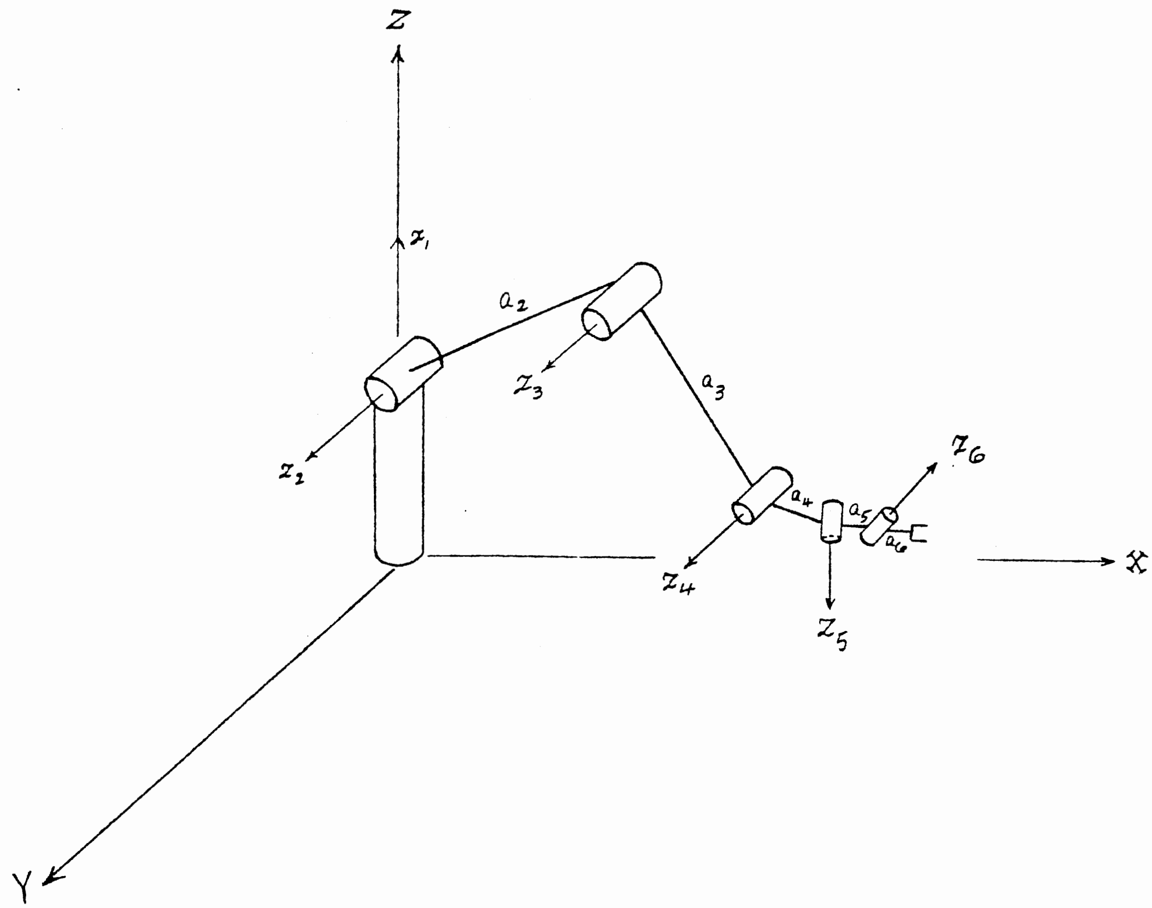

Figure 9. Outlook Example of the Robot Arm

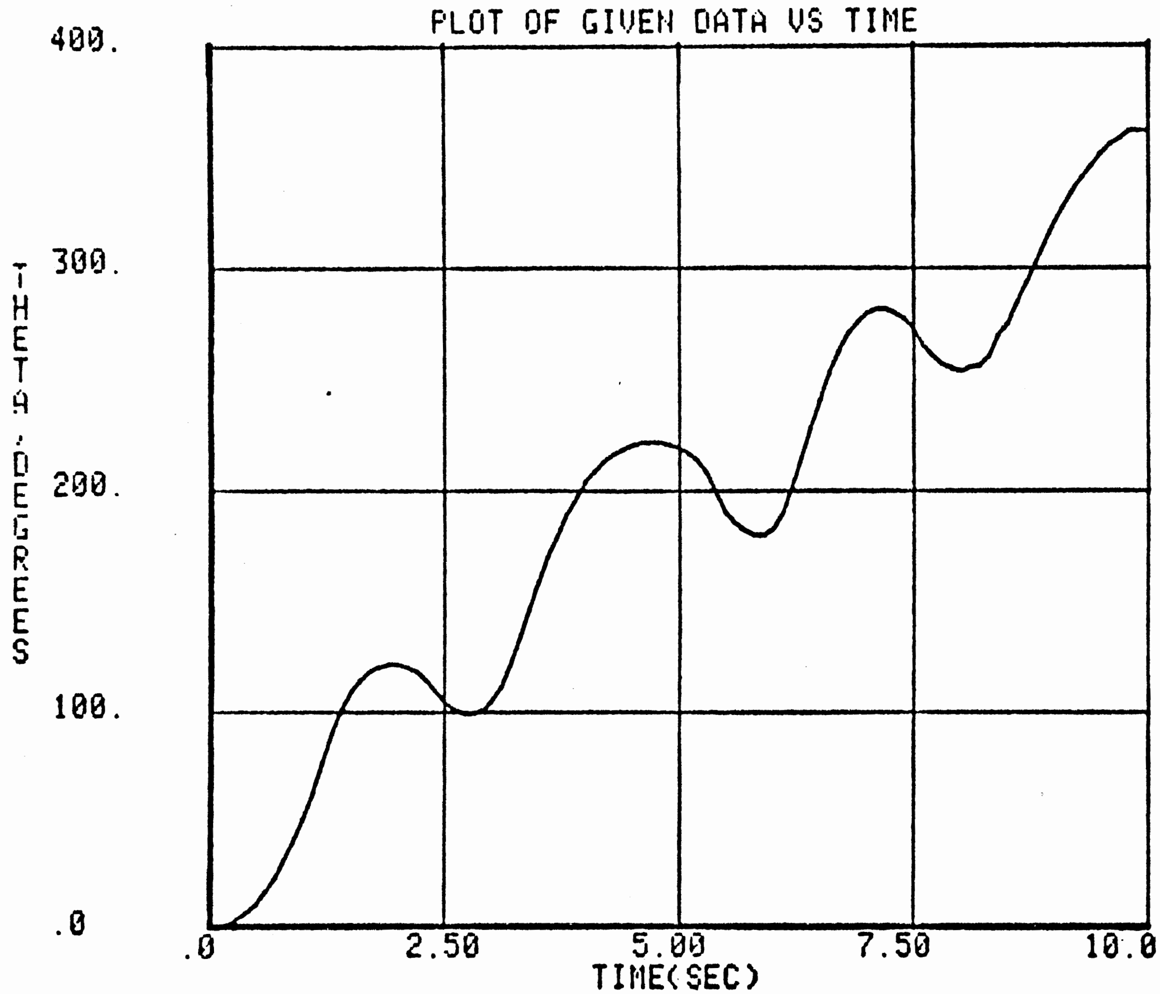

Figure 10. Given Numerical Data vs Time

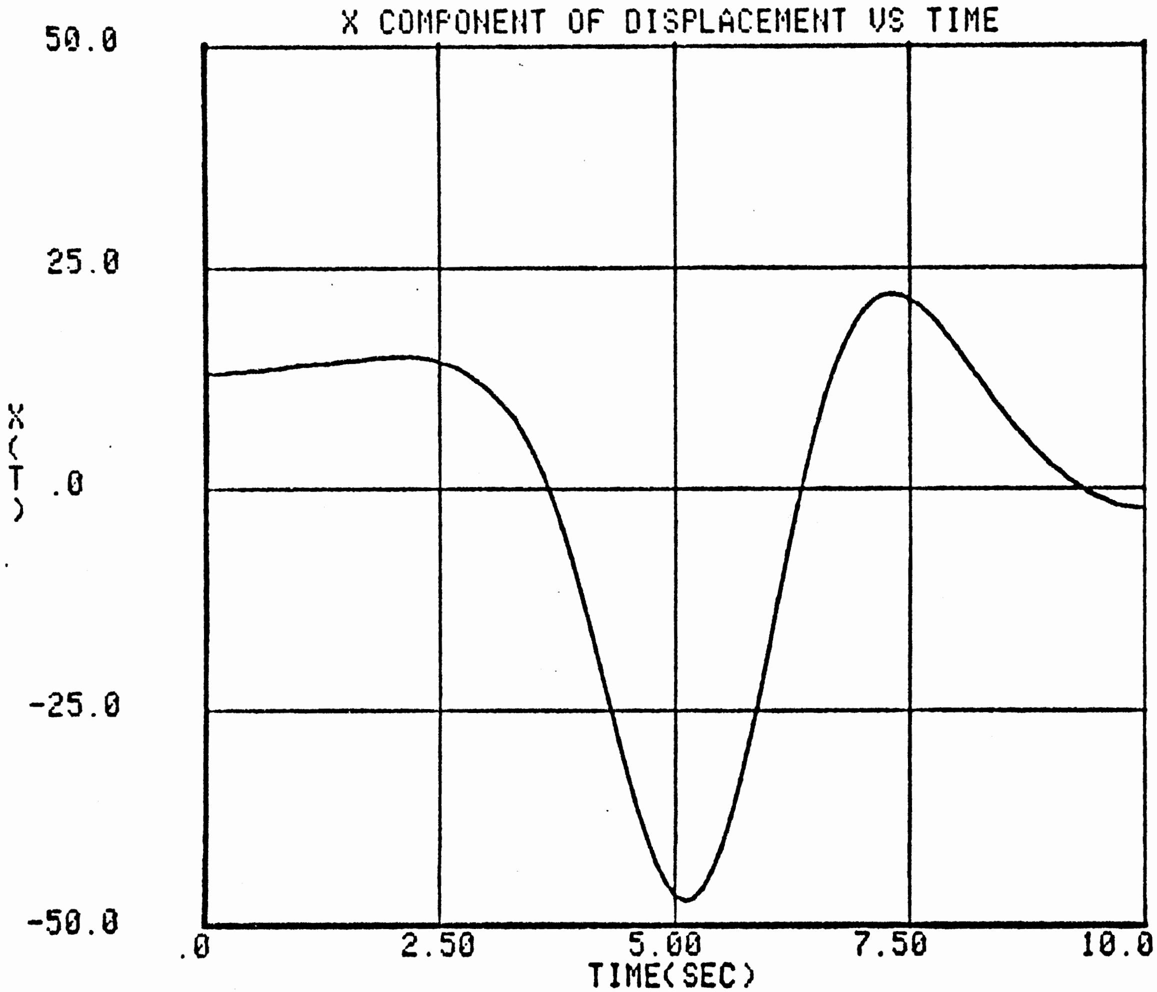

Figure 11. X Component of Displacement vs Time

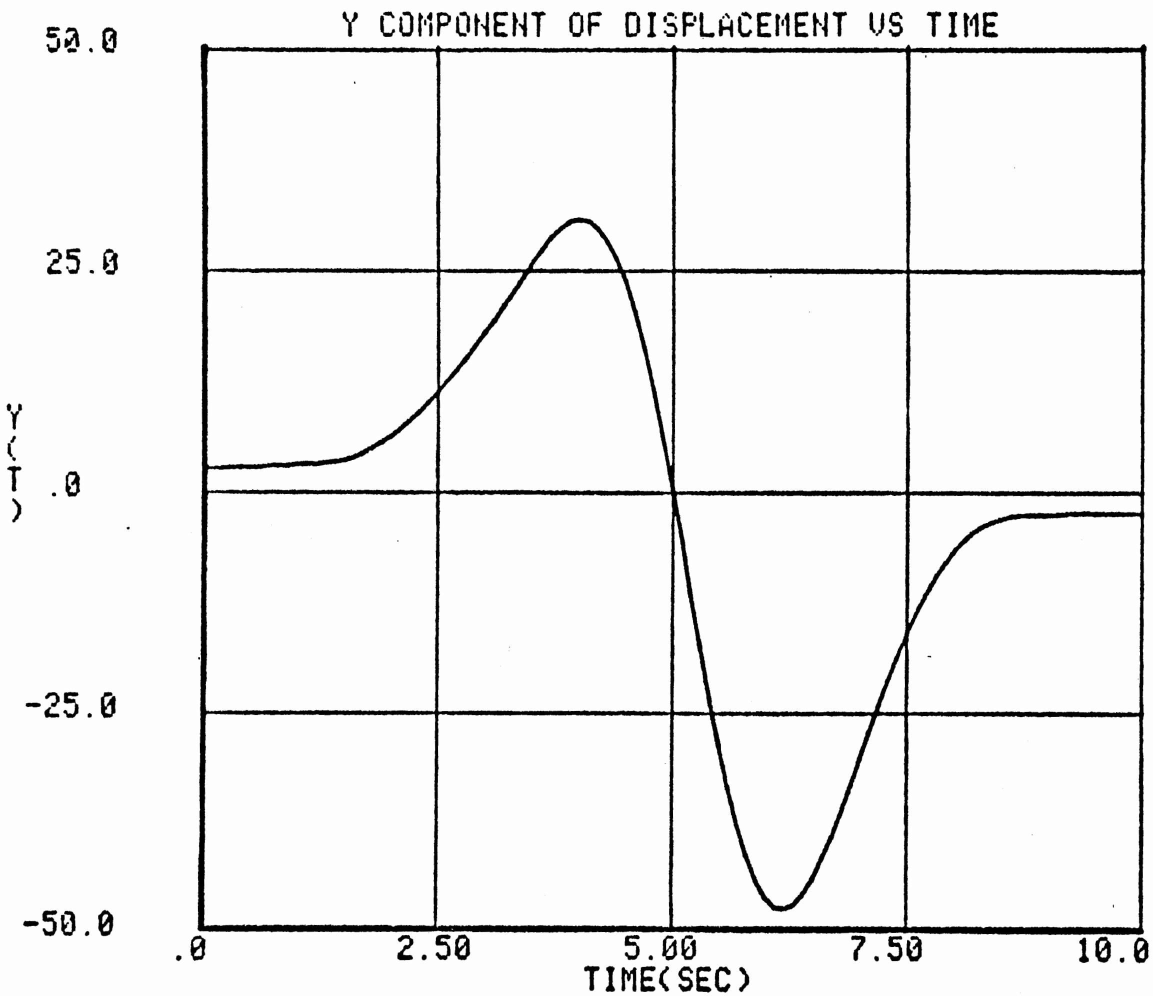

Figure 12. Y Component of Displacement vs Time

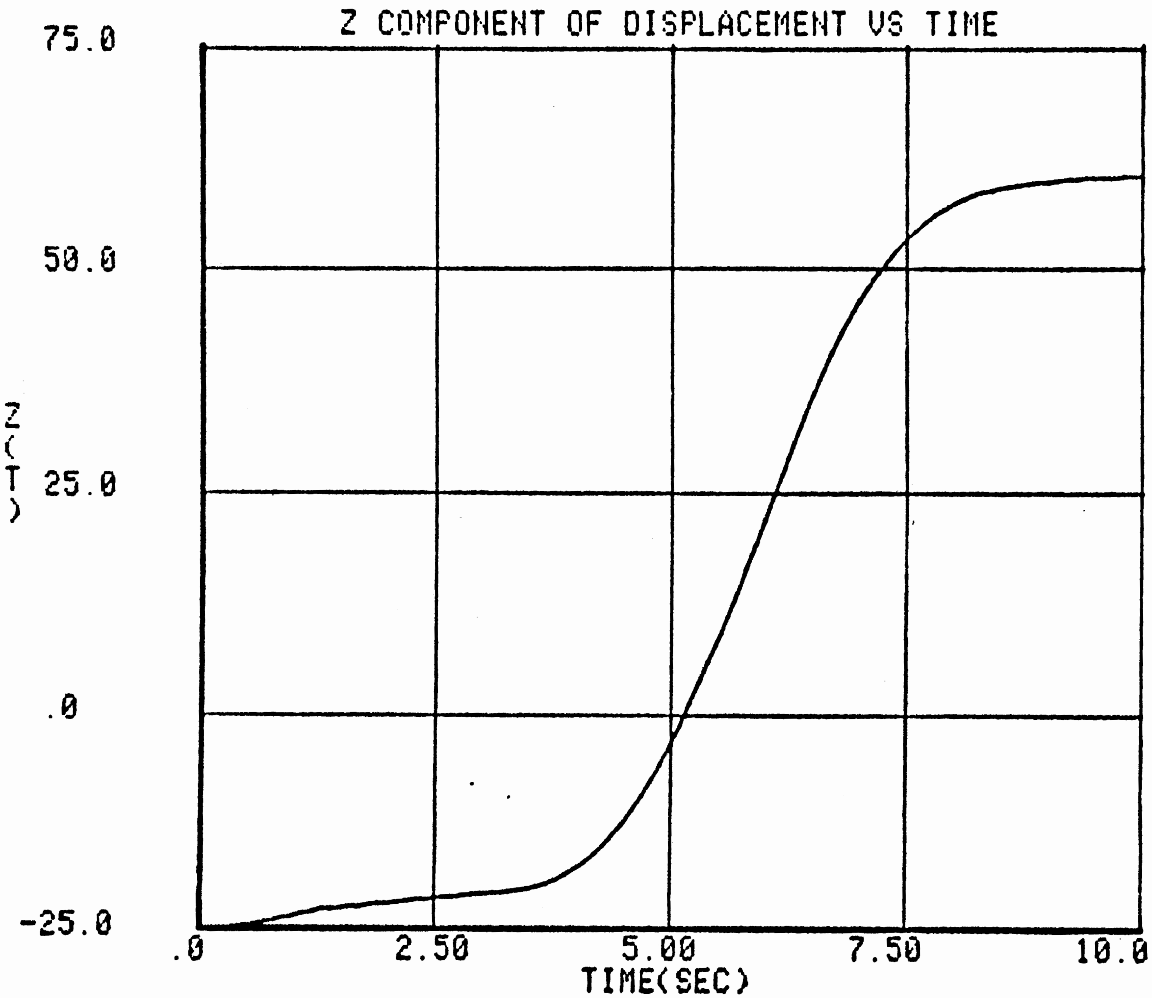

Figure 13. Z Component of Displacement vs Time

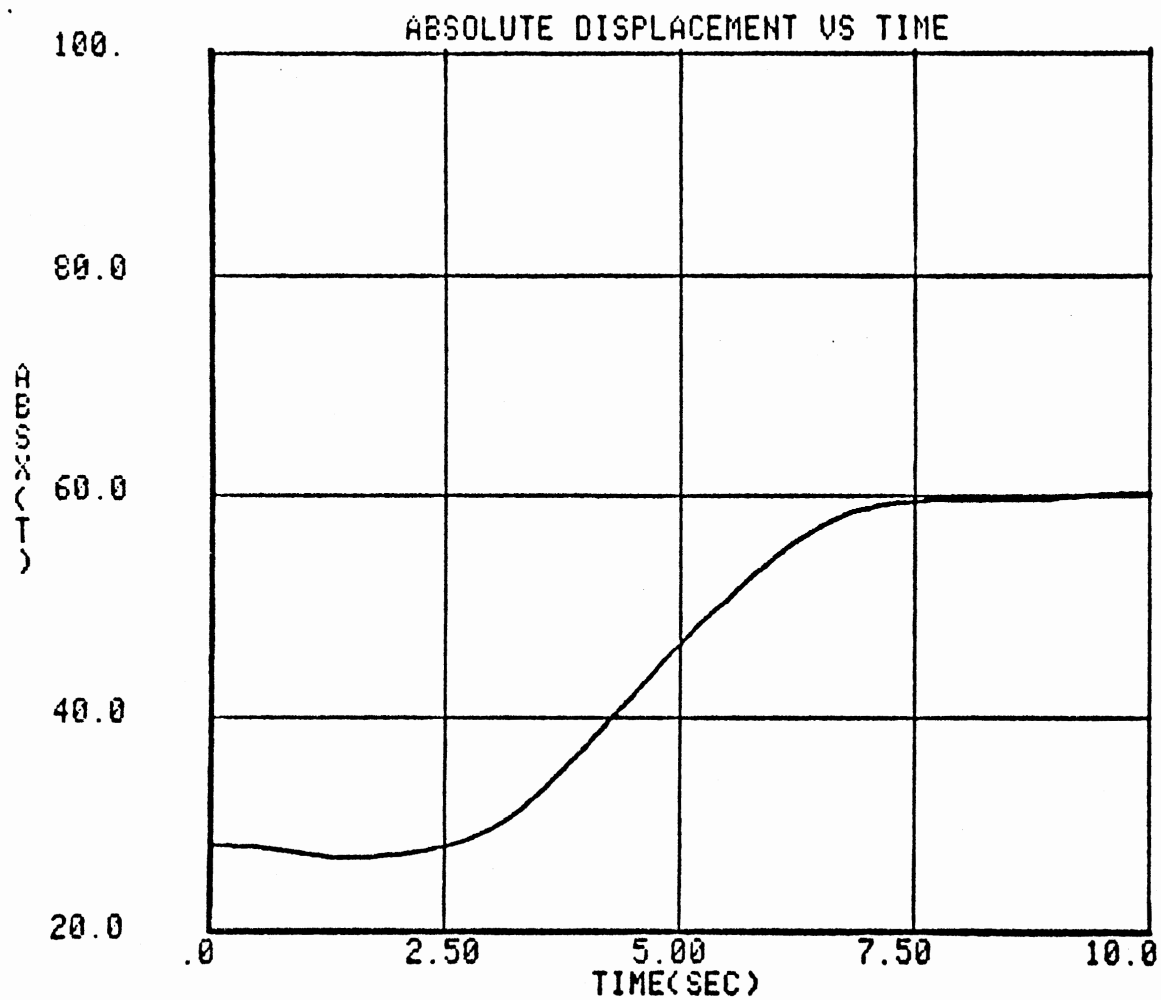

Figure 14. Absolute Displacement vs Time

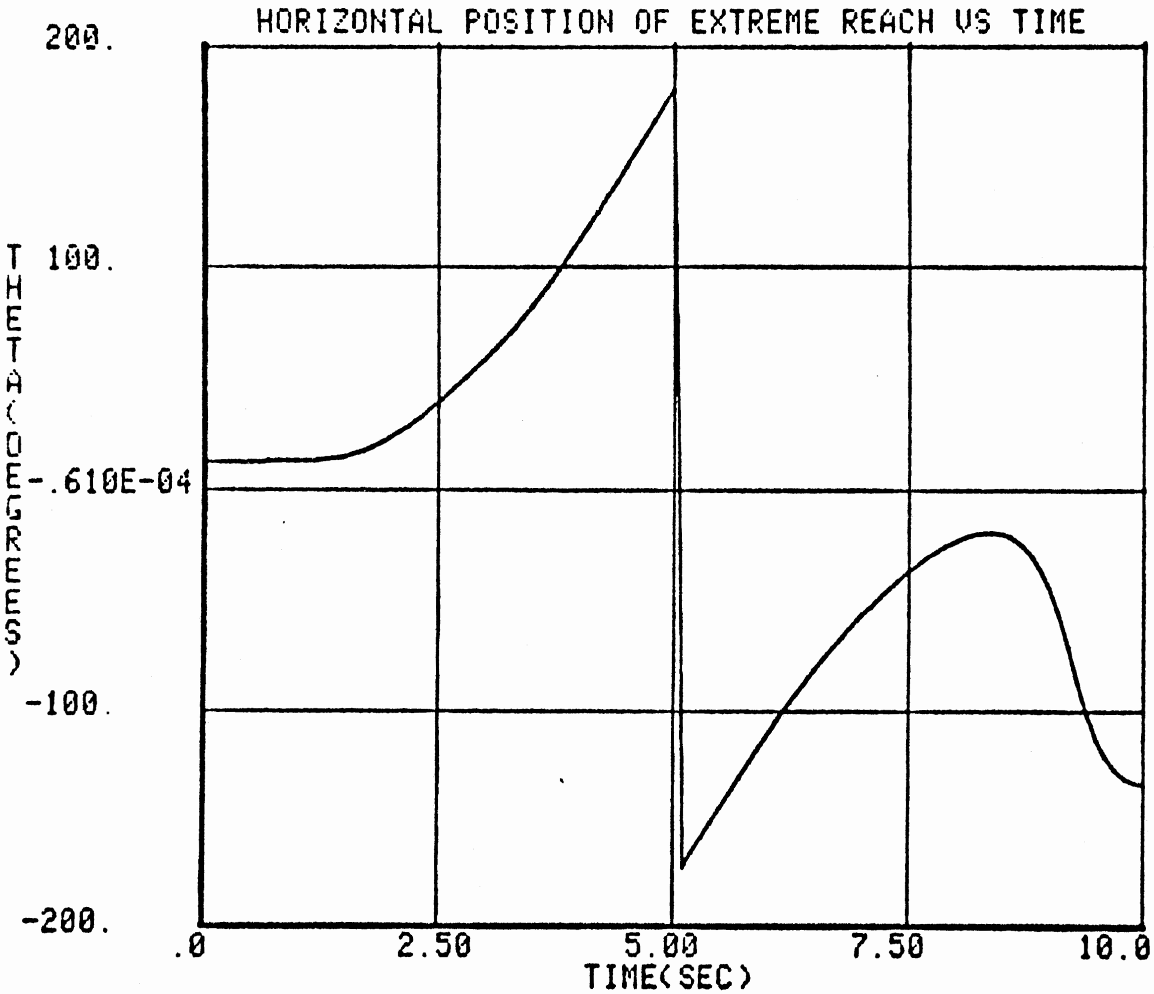

Figure 15. Horizontal Position of Extreme Reach vs Time

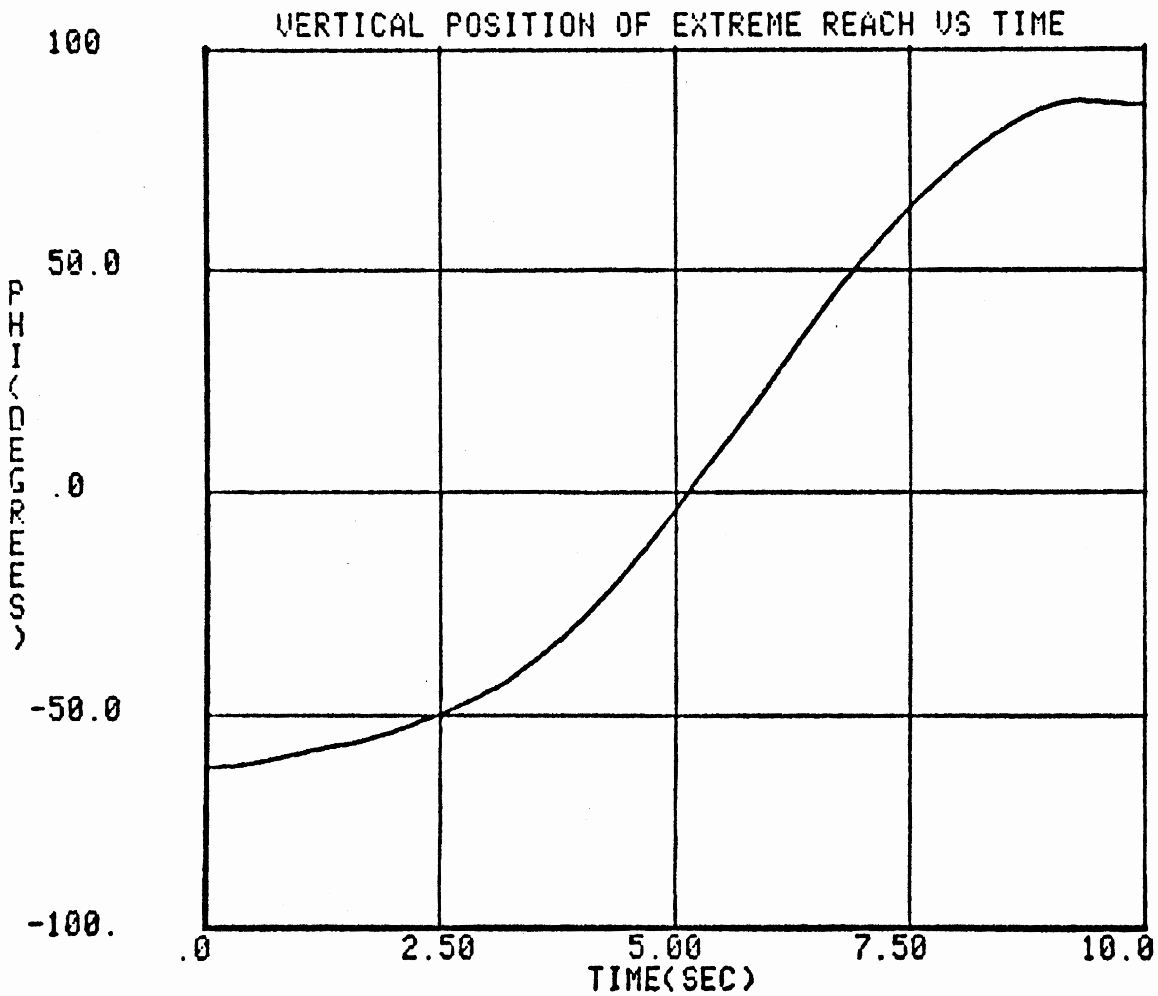

Figure 16. Vertical Position of Extreme Reach vs Time

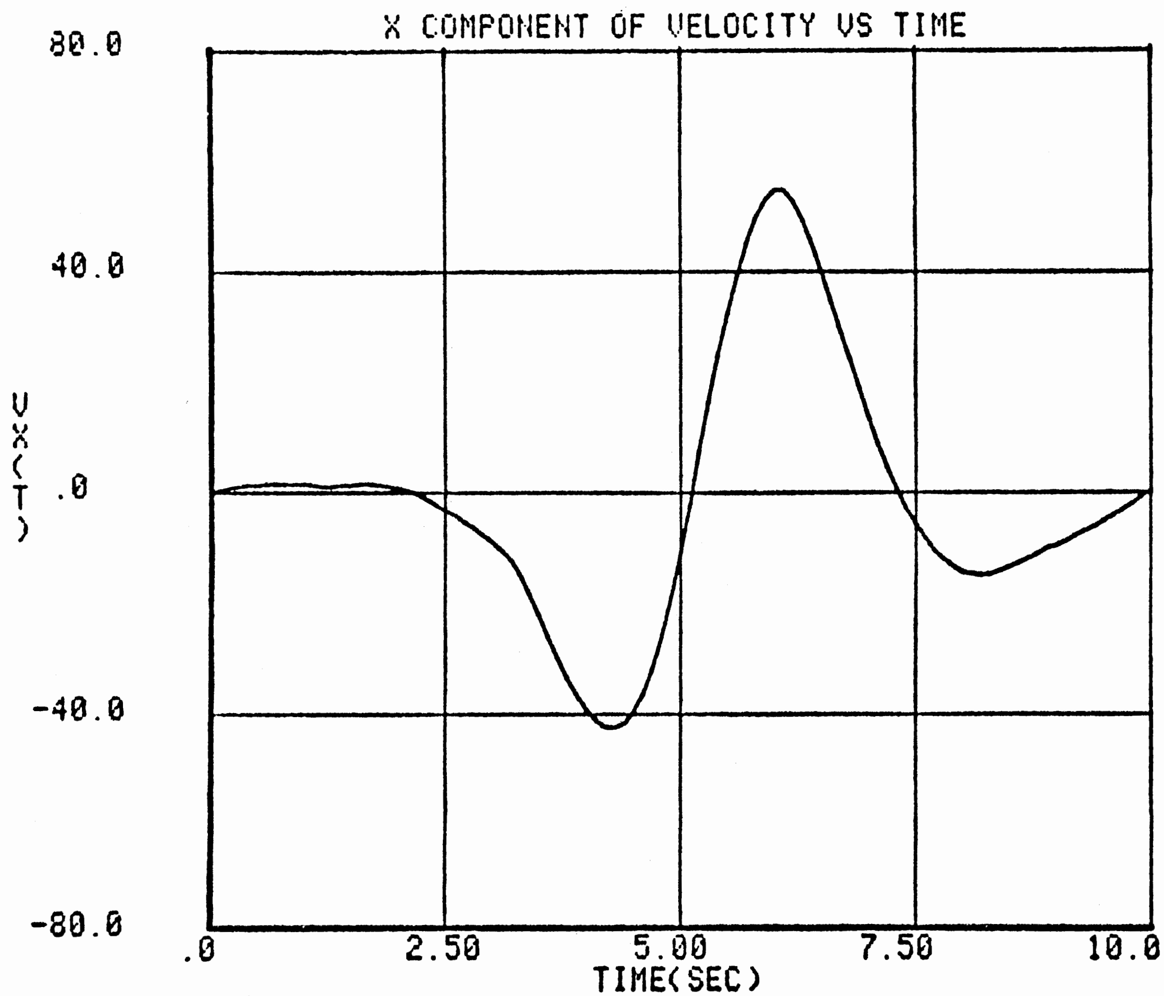

Figure 17. X Component of Velocity vs Time

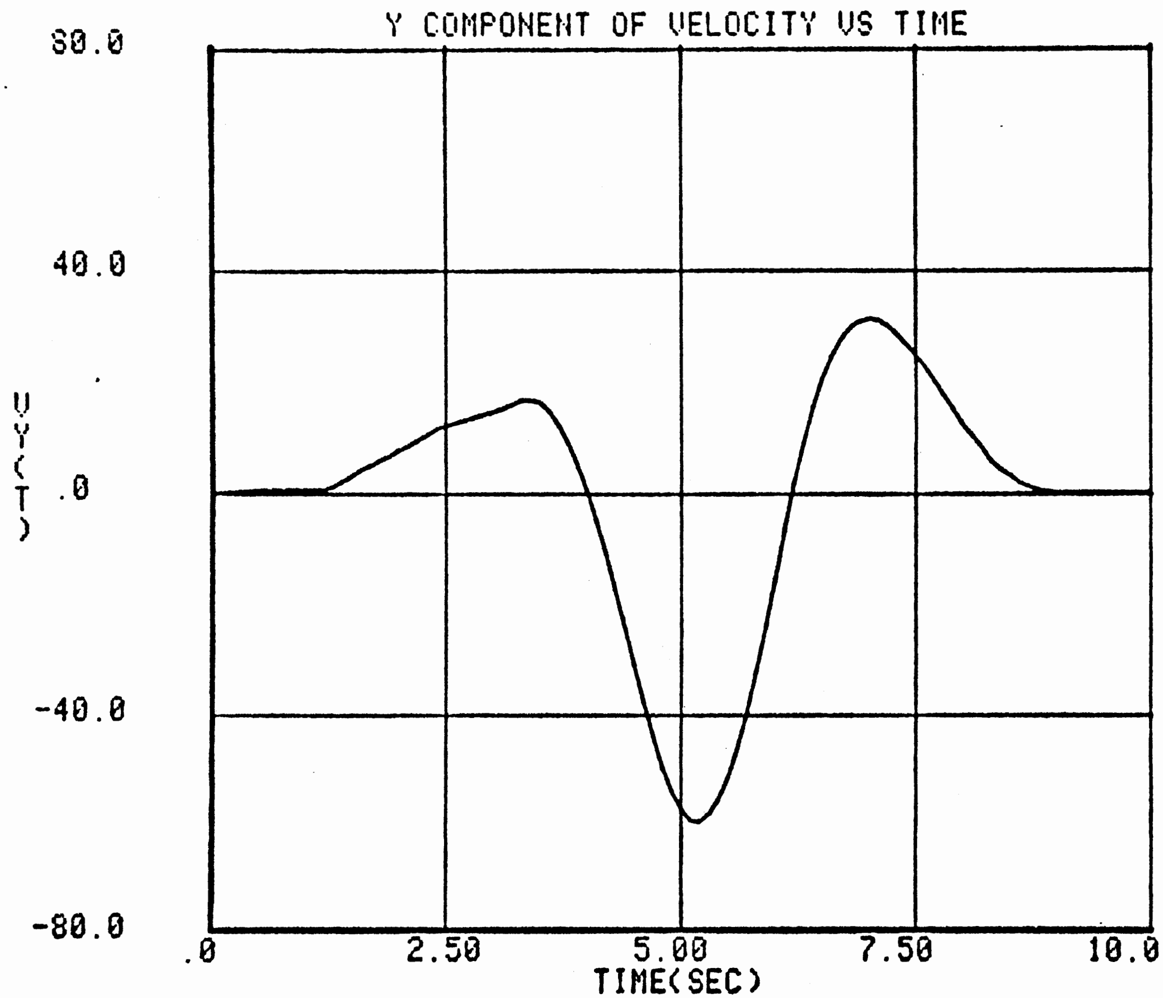

Figure 18. Y Component of Velocity vs Time

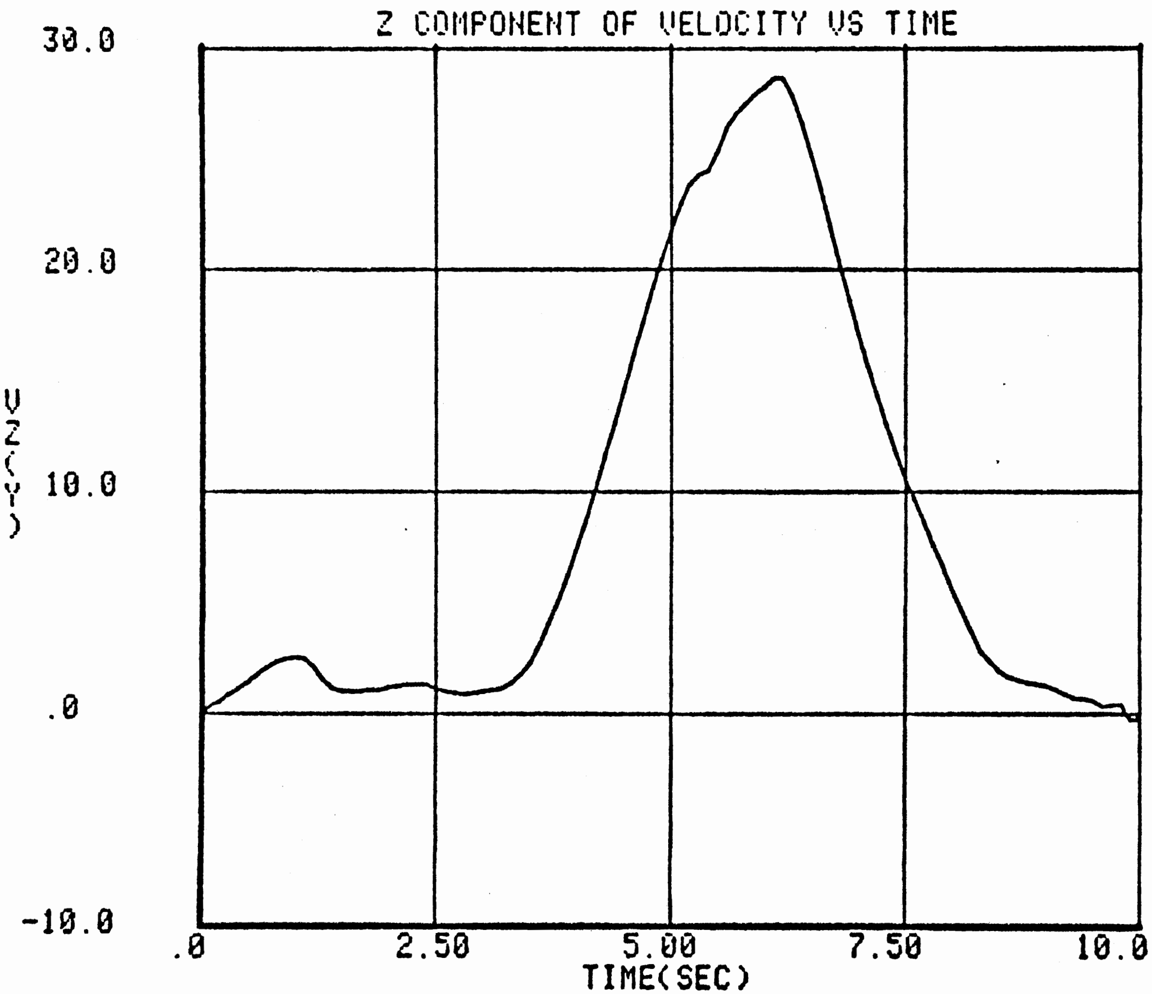

Figure 19. Z Component of Velocity vs Time

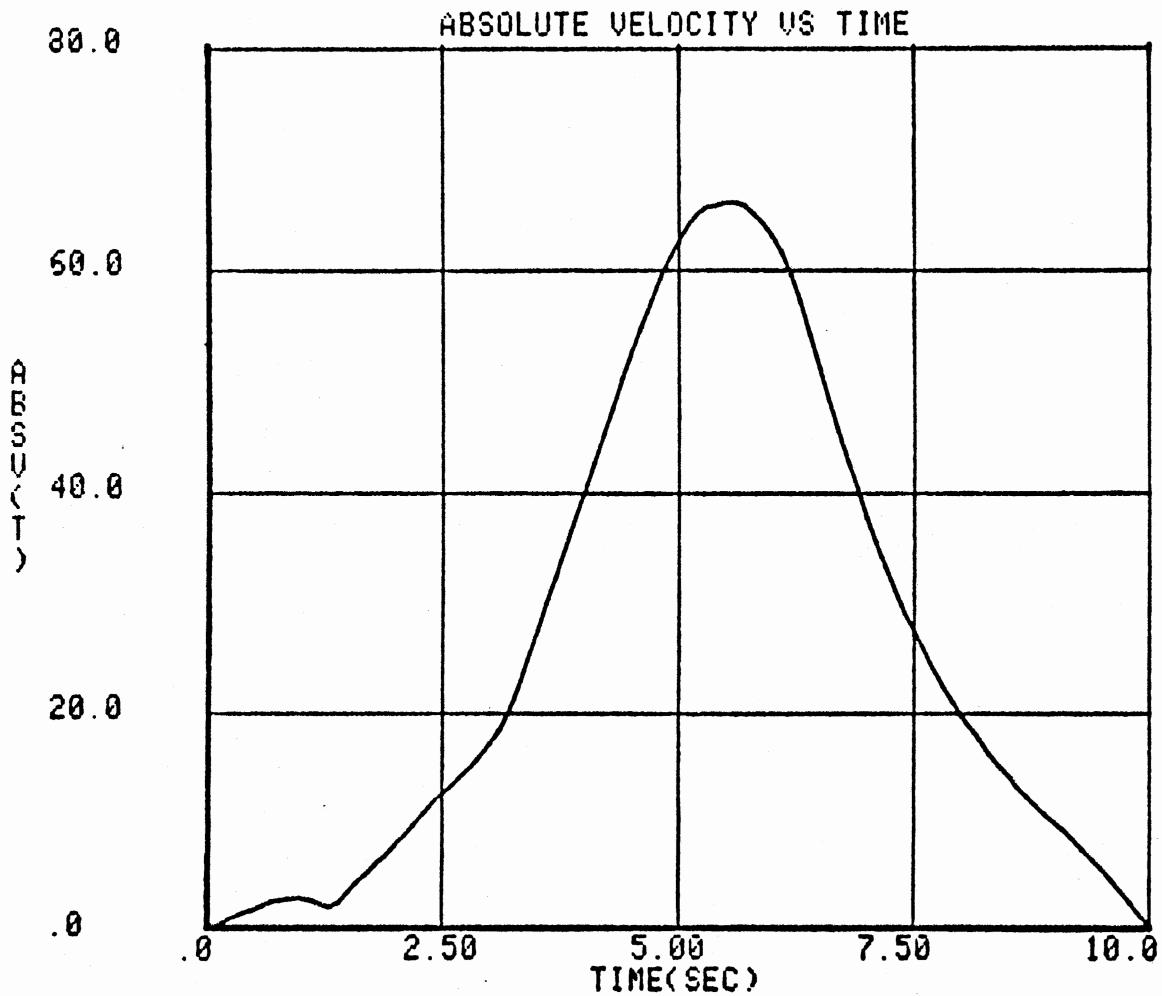

Figure 20. Absolute Velocity vs Time

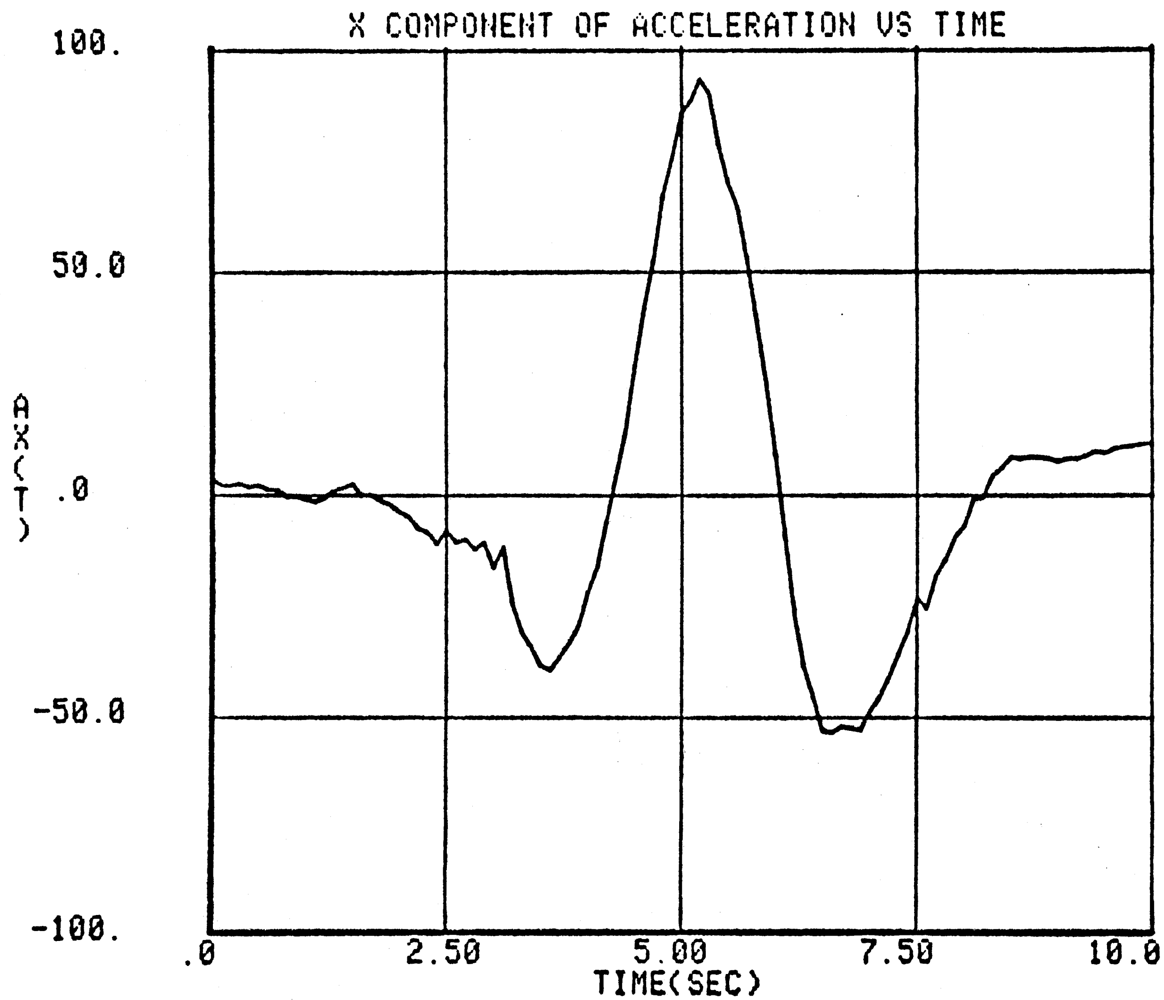

Figure 21. X Component of Acceleration vs Time

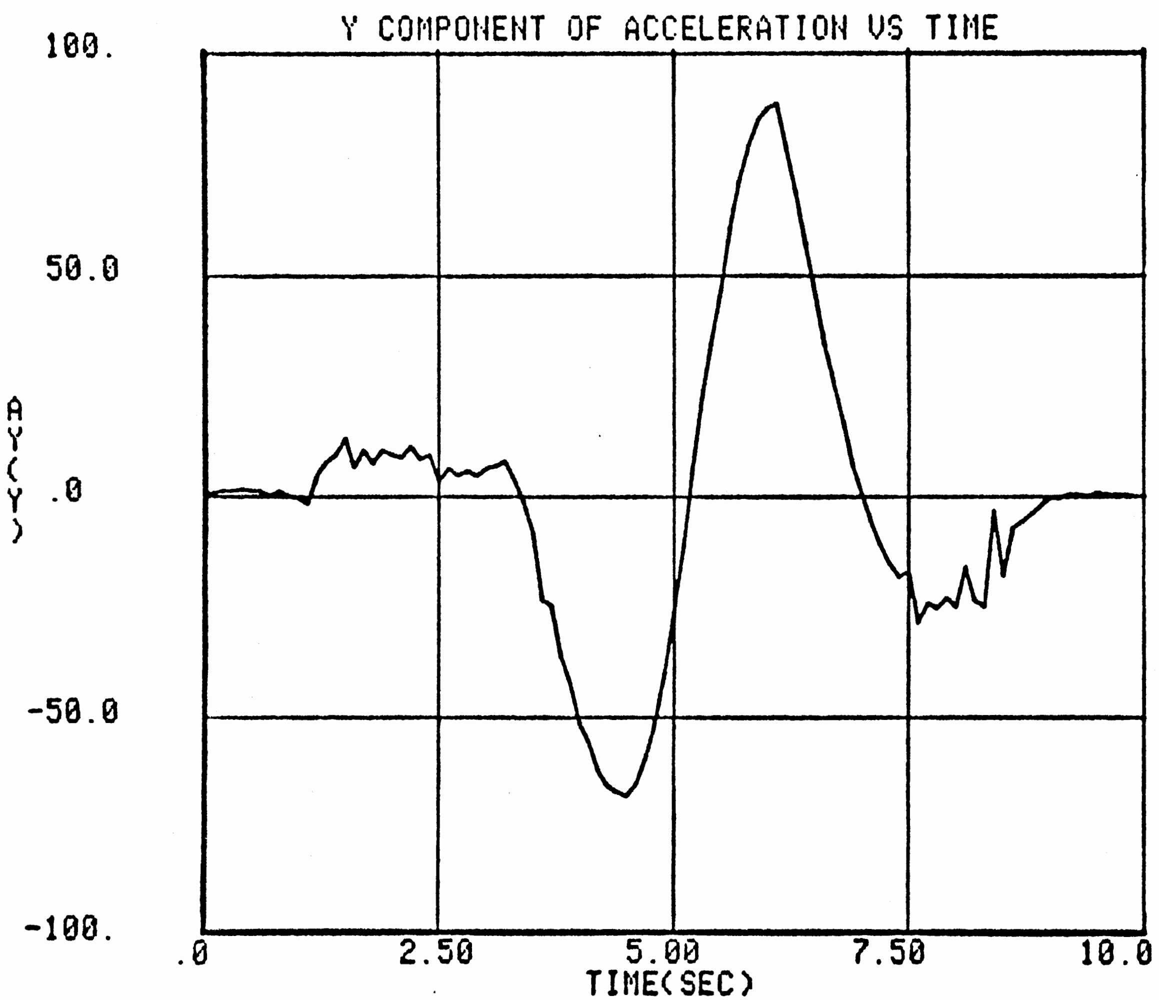

Figure 22. Y Component of Acceleration vs Time

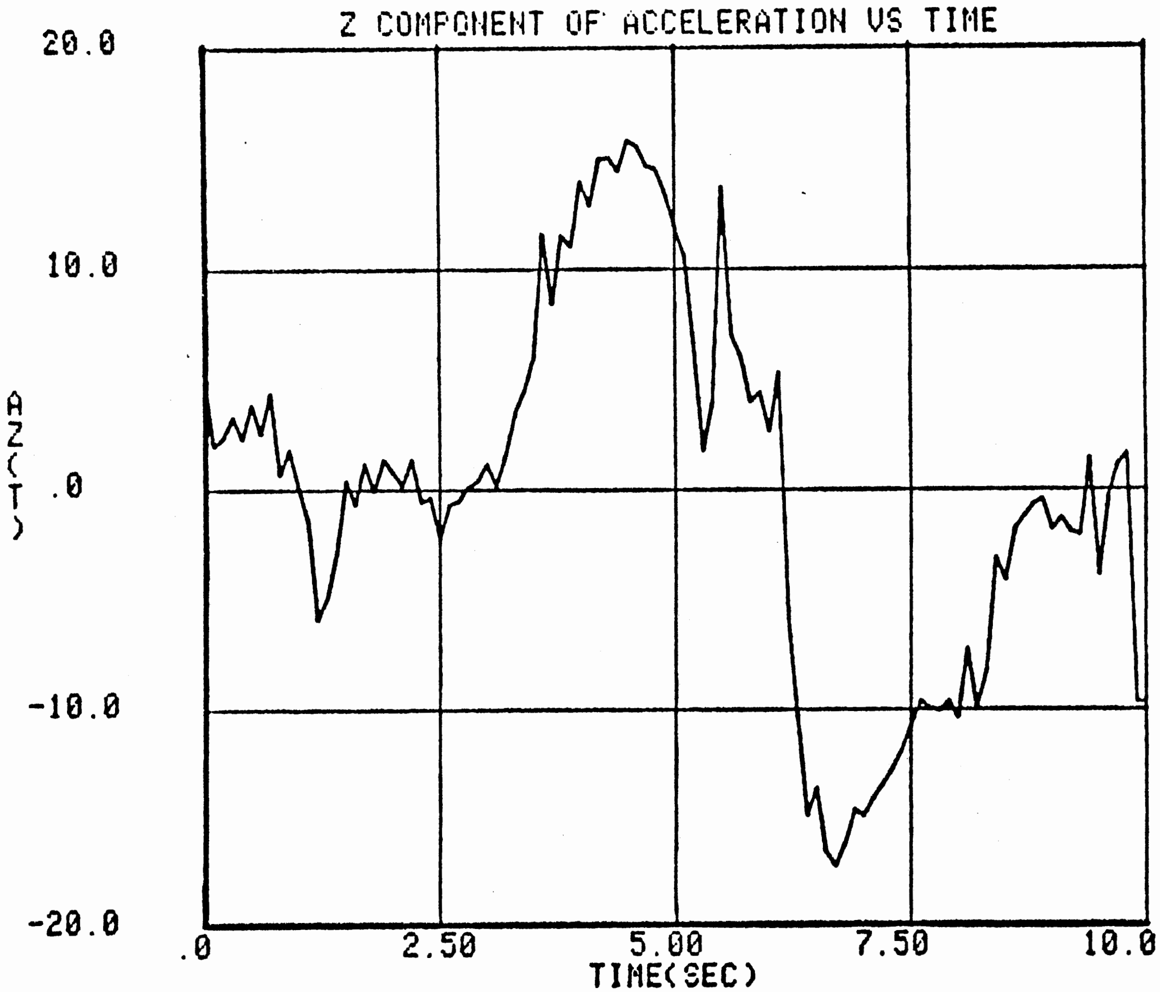

Figure 23. Z Component of Acceleration vs Time

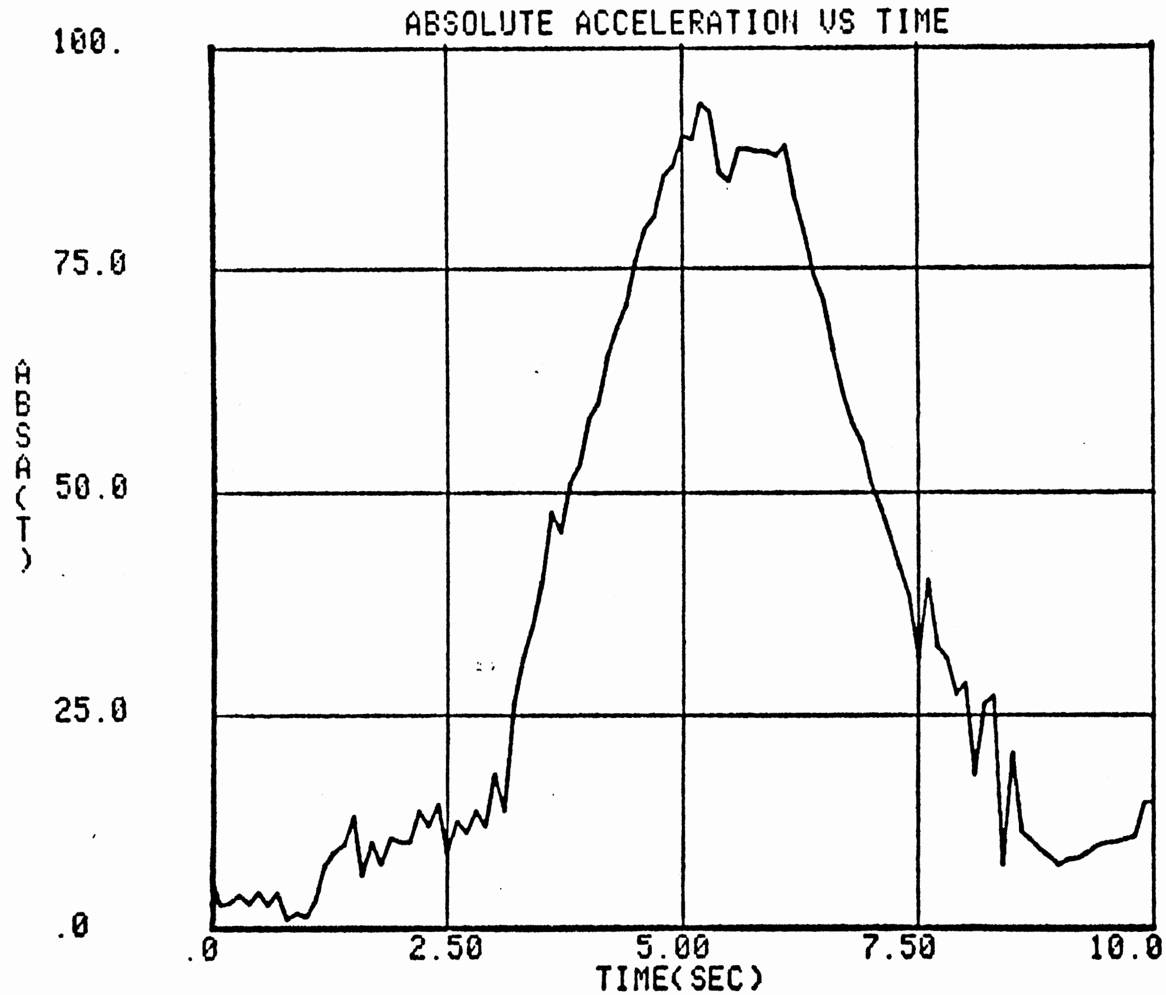

Figure 24. Absolute Acceleration vs Time

CHAPTER IV

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

This research provides a foundation for the kinematic analysis of robot arms/manipulators. The problem was: Given the relative angular displacements, angular velocities, and angular accelerations of each one of the joints in the arm determine the absolute displacement, velocity, and acceleration components of the robot hand.

To find the answers to the above problem equations of displacement, velocity, and acceleration of the robot hand were derived and an interactive computer program was developed, verified, and applied to plot the results. Input data of this program may be supplied either in an analytical form or in numerical form. Analytical data may be supplied by choosing any one of the provided motion programs for any one of the joints angular motion, and numerical data may be supplied in the form of angular displacement vs time, angular velocity vs time, or angular accelerations vs time. Numerical methods are used to differentiate/integrate the supplied numerical data in order to provide angular velocity and angular accelration/angular displacement and angular velocity of the corresponding joint.

The general kinematic analysis problem as stated above has been titled the "direct task" [2]. Another completely different problem in robots and manipulators is called the "reverse task". Given the positions, and orientations, absolute velocities and absolute accelerations of the output link (the end effector), determine the relative displacements, velocities, and accelerations required in the various joints as functions of time. This problem is related to the general kinematic synthesis or optimization problem and is yet to be solved. This study may be used to compare the kinematic performance of the designed robot arms and hence reduce the time and risk involved in selecting the structure and size in order to produce robot arms with assurance that the resulting machine will meet the required specifications.

In analyzing a specific robot arm using this research, the results should be compared for verification purposes with the results obtained experimentally from that particular robot arm. After the above comparison, the difference between the components of actual and desired displacement, velocity, and acceleration of the hand thus can be monitored continuously.

BIBLIOGRAPHY

1. Loeff, L. A., and Soni, A. H. "An Algorithm for Computer Guidance of a Manipulator in Between Obstacles." ASME, Journal of Engineering for Industry, 22 (1975), 836-842.

2. Uicker, J. J. "Some Unanswered Questions on the Kineamatic Analysis of Robots and Manipulators", Proceedings of the National Science Foundation, Workshop on the Impact on the Academic Community of Required Research Activity for Generalized Robotic Manipulators. University of Florida. (February 1978) 192-211.

3. Soni, A. H. Mechanism Synthesis and Analysis. McGraw-Hill Book Company, New York, (1976).

4. Pieper, D. L. "The Kinematics of Manipulators Under Computer Control." Ph.D. Thesis. Stanford University (1968).

5. Hartenberg, R. S. and Denavit, J. "Kinematic Synthesis of Linkages." McGraw-Hill, New York, (1961).

6. Uicker, J. J., J. Denavit, and R. S. Hartenberg. "An Iterative Method for the Displacement Analysis of Special Mechanisms." Journal of Applied Mechanics, 71 (June 1964), 309.

7. J. Denavit and R. S. Hartenberg. "A Kinematic Notation for Lower Pair Mechanisms Based on Matrices." Journal of Applied Mechanics, 22, Trans. ASME, 77, (1955), 215.

8. Roth, B. "Performance Evaluation of Manipulators from a Kinematic Viewpoint," Performance Evaluation of Programmable Robots and Manipulators, NSB Special Publication 459 (1979), 39-62.

9. Tsai, Y. C. and A. H. Soni. "Accessible Region and Synthesis or Robot Arms," ASME Paper No. 80-DET-101, Presented at ASME 16th Mechanisms Conference, California, September 28 - October 1, 1980.

10. Tsai, Y. C. "Synthesis of Robots/Manipulators for a Prescribed Working Space." Ph.D. Dissertation,

Oklahoma State Univesity (1981).

11. Duffy, J. "A Kinematic Analysis and Classification of Robot Arms" Workshop on the Impact on the Academic Community of Required Research Activity for Generalized Robotic Manipulators. University of Florida, (1978), 43-64.

12. Sugiomoto, D. "An Approach to Structural Synthesis of Robots," Proceedings of the 9th International Symposium on Industrial Robots. Washington, D. C. (1979).

13. Hornbeck. R. W. "Numerical Methods", QPI Series, Quantum Publishers, Inc., New York, (1975).

14. Vicker, J. J., J. Denavit, and R. S. Hartenberg. "Velocity, Acceleration, and Static-Force Analysis of Spatial Linkages." Journal of Applied Mechanics, 21. (December 1975), 903.

APPENDIX

COMPUTER PROGRAM

The following 16 pages are the listings of the interactive computerprogram written to analyze the displacement, velocity, and acceleration of the robot hand. This program can be used to analyze 2 to 9-R robot arms. If it is desired to analyze robot arms with more than 9 revolute pairs only the dimension statements must be changed both in the main program and the subroutines. Since the program is interactive and uses TSO plotting routines, the user must run it in a Tektronix terminal. The instructions of how to use the program are embedded with the program.

```
C
C
C *** THE KINEMATIC ANALYSIS OF AN N-R ROBOT ARM ***
C
C
      DIMENSION AW(9,4,4),Q(9,4,4),B(9,4,4),WK(9,2,4,4),SUM(4,4),
     1C(9,4,4),SUM1(4,4),TT(9,4,4),A(9),THETA(9),ALFA(9),S(9),OMEGA(9),
     2ANACC(9),V(4,4),W(4,4),CE(4,4),WP(4,4),P(4,4),B1(9,4,4),WK1(4,4)
     3,WK2(4,4),WK3(4,4),WK4(4,4),CD(9,4,4),SUM2(4,4),DELTH(9),
     4THZ(9),THF(9),TIME(201),ABSX(201),ABSV(201),ABSA(201),
5GAMA(201),PHI(201),X(201),Y(201),Z(201),XD(201),YD(201),ZD(201),XD
     6D(201),YDD(201),ZDD(201)
      INTEGER PP(9)
C     USE INTERACTIVE PROGRAMING TO ENTER THE INPUT DATA
      WRITE(6,1)
  1   FORMAT(1H1//////////16X,'THE KINEMATIC ANALYSIS OF AN N-R ROBOT
AR
     1M'///35X,'BY'//29X,'HASSAN ASLROUSTA')
      PAUSE
  15  WRITE(6,10)
 10   FORMAT(20X,'* INSTRUCTIONS *'////' ENTER THE DATA REQUESTED AFTER
     1? IS APPEARED'/' IF YOU ARE AT THE BOTTOM OF THE SCREEN HIT PAGE
K
     2EY')
      WRITE(6,2)
 2    FORMAT(///' ENTER THE NUMBER OF JOINTS IN THE ARM N>=2,AND HIT
RET
     1URN KEY')
      READ(5,*) N
      WRITE(6,3)
  3   FORMAT(' ENTER THE LINK LENGTHS A1,A2,A3,....,AN (IN ANY UNIT)
     1'/' HIT RETURN KEY AFTER EACH VALUE IS ENTERED')
      DO 4 I=1,N
  4   READ(5,*) A(I)
      WRITE(6,5)
```

```
  5   FORMAT(' ENTER THE TWIST ANGLES ALFA1,ALFA2,...,ALFAN (IN DEGREES)
     1'/' HIT RETURN KEY AFTER EACH VALUE IS ENTERED')
      DO 6 I=1,N
  6   READ(5,*) ALFA(I)
      WRITE(6,7)
  7   FORMAT(' ENTER THE OFFSETS S1,S2,S3,...,SN (UNIT MUST BE THE SAME
     1'/' AS THE LINK LENGTHS),AND HIT RETURN KEY AFTER EACH VALUE')
      DO 8 I=1,N
  8   READ(5,*) S(I)
      PY=3.14159265/180.0
      PY1=3.14159265
      WRITE(6,9)
  9   FORMAT(' ENTER THE RANGE OF OSCILLATION OF JOINTS 1,2,...,N(IN DEG
     1REES)'/' FOR EACH ? ENTER 2 VALUES WITH A COMMA IN BETWEEN')
      DO 11 I=1,N
      READ(5,*) THZ(I),THF(I)
      DELTH(I)=(THF(I)-THZ(I))*PY
      THZ(I)=THZ(I)*PY
 11   CONTINUE
      TAW=10.0
      WRITE(6,31)
 31   FORMAT(' WILL YOU USE NUMERICAL DATA FOR AT LEAST ONE JOINT ANGULA
     1R MOTION ?'/' ENTER 1 FOR YES AND 2 FOR NO AND HIT RETURN KEY')
      READ(5,*) NP
      IF(NP.EQ.1) GO TO 32
      NK=100
      DELT=0.1
      GO TO 92
 32   WRITE(6,91)
 91   FORMAT(' ENTER THE NUMBER OF DATA POINTS AND HIT RETURN KEY')
      READ(5,*) NK
```

```
      DN=NK
      DELT=TAW/DN
 92   WRITE(6,12)
 12   FORMAT(' YOU HAVE THE OPTION OF CHOOSING ANALYTICAL OR NUMERICAL I
     1NPUT FOR EACH JOINT ANGULAR MOTION.'/' ENTER THE CORRESPONDING NUM
     2BER FOR JOINTS 1,2,...,N AFTER EACH ? AND HIT RETURN KEY'///16X,'1
     3.  CONSTANT VELOCITY'/16X,'2.  SIMPLE HARMONIC'/16X,'3.  MODIFIED
     4HARMONIC'/16X,'4.  CYCLOIDAL'/16X,'5.  3-4-5 POLYNOMIAL'/16X,'6.
     5CUBIC CURVE TYPE 2'/16X,'7.  DATA FOR JOINT ANGULAR DISPLACEMENT'/
     616X,'8.  DATA FOR JOINT ANGULAR VELOCITY'/16X,'9.  DATA FOR JOINT
     7ANGULAR ACCELERATION')
      DO 14 I=1,N
 14   READ(5,*) PP(I)
      T=0.0
      NT=NK+1
      DO 500 III=1,NT
      TIME(III)=T
      DO 13 I=1,N
      II=PP(I)
      CALL SUPPLY(THETA,OMEGA,ANACC,N,I,II,TAW,THZ,DELTH,PY1,T,NK,DELT,I
     1II,PY,NT)
 13   CONTINUE
C     INITIAL SETUP
C
      DO 30 I=1,4
      DO 20 J=1,4
      WK(1,1,I,J)=0.0
  20  WK(N,2,I,J)=0.0
      WK(1,1,I,I)=1.0
      WK(N,2,I,I)=1.0
  30  CONTINUE
C
```

```
C      COMPUTE THE TRANSFORMATION MATRICES A1,A2,A3,....,AN,AND THE
C      DERIVITIVE OPERATOR MATRICES Q1,Q2,Q3,...,QN
C
       DO 40 I=1,N
       DO 35 J=1,4
       DO 35 K=1,4
       AW(I,J,K)=0.0
       Q(I,J,K)=0.0
   35  CONTINUE
   40  CONTINUE
       DO 50 I=1,N
       SITH=SIN(THETA(I))
       COTH=COS(THETA(I))
       SIAL=SIN(PY*ALFA(I))
       COAL=COS(PY*ALFA(I))
       AW(I,1,1)=COTH
       AW(I,2,1)=SITH
       AW(I,1,2)=-SITH*COAL
       AW(I,2,2)=COTH*COAL
       AW(I,3,2)=SIAL
       AW(I,1,3)=SITH*SIAL
       AW(I,2,3)=-COTH*SIAL
       AW(I,3,3)=COAL
       AW(I,1,4)=A(I)*COTH
       AW(I,2,4)=A(I)*SITH
       AW(I,3,4)=S(I)
       AW(I,4,4)=1.0
       Q(I,1,2)=-1.0
       Q(I,2,1)=1.0
   50  CONTINUE
C
```

```
C     COMPUTE THE MATRICES Q1A1,Q2A2,....,QNAN,AND Q1Q1A1,Q2Q2A2,
C     ....,QNQNAN
C
      DO 60 I=1,N
      DO 60 J=1,4
      DO 60 K=1,4
      B(I,J,K)=0.0
      DO 60 L=1,4
      B(I,J,K)=B(I,J,K)+Q(I,J,L)*AW(I,L,K)
   60 CONTINUE
      DO 65 I=1,N
      DO 65 J=1,4
      DO 65 K=1,4
      B1(I,J,K)=0.0
      DO 65 L=1,4
      B1(I,J,K)=B1(I,J,K)+Q(I,J,L)*B(I,L,K)
   65 CONTINUE
C
C     COMPUTE THE TRANFORMATION MATRICES A1,A1A2,A1A2A3,A1A2....AN-1,
C     AND A1,A1A2....AN,AND ALSO COMPUTE A2A3....AN,A3A4....AN,AN
C
      DO 80 I=2,N
      M=N+1-I
      DO 70 J=1,4
      DO 70 K=1,4
      WK(I,1,J,K)=0.0
      WK(M,2,J,K)=0.0
      DO 70 L=1,4
      WK(I,1,J,K)=WK(I,1,J,K)+WK(I-1,1,J,L)*AW(I-1,L,K)
      WK(M,2,J,K)=WK(M,2,J,K)+AW(M+1,J,L)*WK(M+1,2,L,K)
   70 CONTINUE
   80 CONTINUE
```

```
      DO 86 I=2,N
      DO 85 J=1,4
      DO 85 K=1,4
      TT(I,J,K)=0.0
      DO 85 L=1,4
      TT(I,J,K)=TT(I,J,K)+WK(I,1,J,L)*AW(I,L,K)
  85  CONTINUE
  86  CONTINUE
C
C     COMPUTE THE MATRICES Q1A1,Q1A1A2...AN,A1Q2A2...AN,A1A2...QNAN
C
      DO 90 I=1,4
      DO 90 J=1,4
      V(I,J)=0.0
      W(I,J)=0.0
  90  CONTINUE
      DO 130 I=1,N
      DO 100 J=1,4
      DO 100 K=1,4
      SUM(J,K)=0.0
      DO 100 L=1,4
      SUM(J,K)=SUM(J,K)+WK(I,1,J,L)*B(I,L,K)
 100  CONTINUE
      DO 110 J=1,4
      DO 110 K=1,4
      C(I,J,K)=0.0
      DO 110 L=1,4
      C(I,J,K)=C(I,J,K)+SUM(J,L)*WK(I,2,L,K)
 110  CONTINUE
      DO 115 J=1,4
      DO 115 K=1,4
```

```
      SUM1(J,K)=C(I,J,K)*OMEGA(I)
      SUM2(J,K)=C(I,J,K)*ANACC(I)
 115  CONTINUE
      DO 120 J=1,4
      DO 120 K=1,4
      V(J,K)=V(J,K)+SUM1(J,K)
      W(J,K)=W(J,K)+SUM2(J,K)
 120  CONTINUE
 130  CONTINUE
C
C     FIND ABSOLUTE DISPLACEMENT AND ABSOLUTE VELOCITY OF THE ROBOT HAND
C
      X(III)=TT(N,1,4)
      Y(III)=TT(N,2,4)
      Z(III)=TT(N,3,4)
      ABSX(III)=(X(III)**2+Y(III)**2+Z(III)**2)**0.5
      GAMA(III)=ATAN2(Y(III),X(III))
      PHI(III)=ARSIN(Z(III)/ABSX(III))
      GAMA(III)=GAMA(III)*1./PY
      PHI(III)=PHI(III)*1./PY
      XD(III)=V(1,4)
      YD(III)=V(2,4)
      ZD(III)=V(3,4)
      ABSV(III)=(XD(III)**2+YD(III)**2+ZD(III)**2)**0.5
      DO 170 I=1,4
      DO 170 J=1,4
 170  P(I,J)=0.0
      DO 260 K=1,N
      DO 180 I=1,4
      DO 180 J=1,4
 180  CD(K,I,J)=0.0
      DO 250 KK=1,N
```

```
      DO 200 I=1,4
      DO 190 J=1,4
190   WK3(I,J)=0.0
200   WK3(I,I)=1.0
      DO 230 KKK=1,N
      CALL SELECT(K,KK,KKK,WK2,N,B1,AW,B)
      DO 210 I=1,4
      DO 210 J=1,4
      WK1(I,J)=0.0
      DO 210 J1=1,4
210   WK1(I,J)=WK1(I,J)+WK3(I,J1)*WK2(J1,J)
      DO 220 I=1,4
      DO 220 J=1,4
      WK3(I,J)=WK1(I,J)
220   CONTINUE
230   CONTINUE
      DO 225 I=1,4
      DO 225 J=1,4
225   WK4(I,J)=WK3(I,J)*OMEGA(KK)
      DO 240 I=1,4
      DO 240 J=1,4
240   CD(K,I,J)=CD(K,I,J)+WK4(I,J)
250   CONTINUE
      DO 255 I=1,4
      DO 255 J=1,4
      CE(I,J)=CD(K,I,J)*OMEGA(K)
      P(I,J)=P(I,J)+CE(I,J)
255   CONTINUE
260   CONTINUE
      DO 270 I=1,4
      DO 270 J=1,4
```

```
 270  WP(I,J)=W(I,J)+P(I,J)
C
C     FIND ABSOLUTE ACCELERATION OF THE ROBOT HAND
C
      XDD(III)=WP(1,4)
      YDD(III)=WP(2,4)
      ZDD(III)=WP(3,4)
      ABSA(III)=(XDD(III)**2+YDD(III)**2+ZDD(III)**2)**0.5
 500  T=T+DELT
      WRITE(6,501)
 501  FORMAT('  HIT RETURN KEY AFTER EACH PLOT TO SEE THE NEXT PLOT')
      PAUSE
      CALL QCKPLT(TIME,X,NT,'TIME(SEC)$','X(T)$','X COMPONENT OF
DISPLAC
     1EMENT VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,Y,NT,'TIME(SEC)$','Y(T)$','Y COMPONENT OF
DISPLAC
     1EMENT VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,Z,NT,'TIME(SEC)$','Z(T)$','Z COMPONENT OF
DISPLAC
     1EMENT VS TIME$',0,5,0)
      PAUSE
      CALL
QCKPLT(TIME,GAMA,NT,'TIME(SEC)$','THETA(DEGREES)$','HORIZONTA
     1L POSITION OF EXTREME REACH VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,PHI,NT,'TIME(SEC)$','PHI(DEGREES)$','VERTICAL
POS
     1ITION OF EXTREME REACH VS TIME$',0,5,0)
      PAUSE
C
C     PLOT ABSOLUTE DISPLACEMENT,ABSOLUTE VELOCITY,AND ABSOLUTE
C     ACCELERATION VS TIME
C
      CALL QCKPLT(TIME,ABSX,NT,'TIME(SEC)$','ABSX(T)$','ABSOLUTE
```

```
DISPLAC
     1EMENT VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,XD,NT,'TIME(SEC)$','VX(T)$','X COMPONENT OF VELOC
     1ITY VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,YD,NT,'TIME(SEC)$','VY(T)$','Y COMPONENT OF VELOC
     1ITY VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,ZD,NT,'TIME(SEC)$','VZ(Y)$','Z COMPONENT OF VELOC
     1ITY VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,ABSV,NT,'TIME(SEC)$','ABSV(T)$','ABSOLUTE VELOCIT
     1Y VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,XDD,NT,'TIME(SEC)$','AX(T)$','X COMPONENT OF ACCE
     1LERATION VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,YDD,NT,'TIME(SEC)$','AY(Y)$','Y COMPONENT OF ACCE
     1LERATION VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,ZDD,NT,'TIME(SEC)$','AZ(T)$','Z COMPONENT OF ACCE
     1LERATION VS TIME$',0,5,0)
      PAUSE
      CALL QCKPLT(TIME,ABSA,NT,'TIME(SEC)$','ABSA(T)$','ABSOLUTE ACCELER
     1ATION VS TIME$',0,5,0)
      PAUSE
      WRITE(6,16)
 16   FORMAT(' WOULD YOU LIKE TO ANALYSE ANOTHER ROBOT ARM ?'/' ENTER 1
     1FOR YES AND 2 FOR NO,AND HIT RETURN KEY')
      READ(5,*) KI
      IF(KI.EQ.1) GO TO 15
```

```
      STOP
      END
      SUBROUTINE SELECT(K,KK,KKK,AS,N,B1,AW,B)
C
C     THIS SUBROUTINE PROVIDES THE PROPER MATRIX TO BE USED IN
C     ACCELERATION EQUATION
C
      DIMENSION AS(4,4),B1(9,4,4),AW(9,4,4),B(9,4,4)
      IF(KKK.NE.KK) GO TO 300
      IF(KK.NE.K) GO TO 320
      DO 290 I=1,4
      DO 290 J=1,4
  290 AS(I,J)=B1(KKK,I,J)
      RETURN
  300 IF(KKK.EQ.K) GO TO 320
      DO 310 I=1,4
      DO 310 J=1,4
  310 AS(I,J)=AW(KKK,I,J)
      RETURN
  320 DO 330 I=1,4
      DO 330 J=1,4
  330 AS(I,J)=B(KKK,I,J)
      RETURN
      END
      SUBROUTINE
SUPPLY(THETA,OMEGA,ANACC,N,I,II,TAW,THZ,DELTH,PY1,T,NK,
     1DELT,III,PY,NT)
C
C     THIS SUBROUTINE PROVIDES ANGULAR DISPLACEMENT,ANGULAR
C     VELOCITY,AND ANGULAR ACCELERATION CORRESPONDING TO
C     THE MOTION PROGRAM CHOSEN FOR EACH OF THE N JOINTS
C     OF THE ROBOT ARM
C
```

```
      DIMENSION THETA(9),OMEGA(9),ANACC(9),THZ(9),DELTH(9)
     1,AK(9,201),BK(9,201),CK(9,201)
      IF(II.EQ.1) GO TO 101
      IF(II.EQ.2) GO TO 102
      IF(II.EQ.3) GO TO 103
      IF(II.EQ.4) GO TO 104
      IF(II.EQ.5) GO TO 105
      IF(II.EQ.6) GO TO 106
      IF(II.EQ.7) GO TO 107
      IF(II.EQ.8) GO TO 108
      IF(III.EQ.1) GO TO 52
      DO 53 LJ=2,NT
      IF(III.EQ.LJ) GO TO 54
   53 CONTINUE
   52 WRITE(6,55) I
   55 FORMAT(' YOU HAVE CHOSEN TO GIVE DATA FOR JOINT(',I2,')
ANGULAR'/'
     1 ACCELERATION. ENTER THE DATA ONE AFTER EACH ? (DATA IN
RAD./SEC.S
     2.)'/' HIT RETURN KEY AFTER EACH VALUE IS ENTERED')
      DO 56 MN=1,NK
      READ(5,*) CK(I,MN)
   56 CONTINUE
      BK(I,1)=0.0
      BK(I,2)=DELT*(CK(I,2)+CK(I,1))/2.0
      DO 57 MN=3,NK
      BK(I,MN)=BK(I,MN-2)+DELT*(CK(I,MN-2)+4.*CK(I,MN-1)+CK(I,MN))/3.
   57 CONTINUE
      AK(I,1)=0.0
      AK(I,2)=DELT*(BK(I,2)+BK(I,1))/2.0
      DO 58 MN=3,NK
      AK(I,MN)=AK(I,MN-2)+DELT*(BK(I,MN-2)+4.*BK(I,MN-1)+BK(I,MN))/3.
   58 CONTINUE
```

```
      AK(I,NK+1)=AK(I,NK)
      BK(I,NK+1)=BK(I,NK)
      CK(I,NK+1)=CK(I,NK)
  54  THETA(I)=AK(I,III)
      OMEGA(I)=BK(I,III)
      ANACC(I)=CK(I,III)
      RETURN
  107 IF(III.EQ.1) GO TO 202
      DO 201 LJ=2,NT
      IF(III.EQ.LJ) GO TO 301
 201  CONTINUE
 202  WRITE(6,37) I
  37  FORMAT(' YOU HAVE CHOSEN TO GIVE DATA FOR JOINT(',I2,') ANGULAR'/'
     1 DISPLACEMENT. ENTER THE DATA ONE AFTER EACH ? (DATA IN DEGREES)'/
     2' HIT RETURN KEY AFTER EACH VALUE IS ENTERED')
      DO 38 MN=1,NK
      READ(5,*) AK(I,MN)
      AK(I,MN)=AK(I,MN)*PY
 38   CONTINUE
      NM=NK-2
      DO 204 JJ=1,2
      BK(I,JJ)=(-3.0*AK(I,JJ)+4.0*AK(I,JJ+1)-AK(I,JJ+2))/(2.0*DELT)
      CK(I,JJ)=(2.0*AK(I,JJ)-5.0*AK(I,JJ+1)+4.0*AK(I,JJ+2)-AK(I,JJ+3))/D
     1ELT**2
 204  CONTINUE
      LM=NK-1
      DO 41 MN=3,NM
      BK(I,MN)=(AK(I,MN-2)-8.*AK(I,MN-1)+8.*AK(I,MN+1)-AK(I,MN+2))/(12.*
     1DELT)
      CK(I,MN)=(-AK(I,MN-2)+16.*AK(I,MN-1)-30.*AK(I,MN)+16.*AK(I,MN+1)-A
     1K(I,MN+2))/(12.*DELT**2)
 41   CONTINUE
```

```
      DO 205 JJ=LM,NK
      BK(I,JJ)=(AK(I,JJ-2)-4.0*AK(I,JJ-1)+3.0*AK(I,JJ))/(2.0*DELT)
      CK(I,JJ)=(-AK(I,JJ-3)+4.0*AK(I,JJ-2)-5.*AK(I,JJ-1)+2.*AK(I,JJ))/
     1DELT**2
 205  CONTINUE
      AK(I,NK+1)=AK(I,NK)
      BK(I,NK+1)=BK(I,NK)
      CK(I,NK+1)=CK(I,NK)
 301  THETA(I)=AK(I,III)
      OMEGA(I)=BK(I,III)
      ANACC(I)=CK(I,III)
      RETURN
 101  THETA(I)=THZ(I)+DELTH(I)*T/TAW
      OMEGA(I)=DELTH(I)/TAW
      ANACC(I)=0.0
      RETURN
 102  THETA(I)=THZ(I)+DELTH(I)*(1.0-COS(PY1*T/TAW))/2.0
      OMEGA(I)=DELTH(I)*PY1*SIN(PY1*T/TAW)/(2.0*TAW)
      ANACC(I)=DELTH(I)*(PY1/TAW)**2*COS(PY1*T/TAW)/2.0
      RETURN
 103  THETA(I)=THZ(I)+DELTH(I)*((1.0-COS(PY1*T/TAW))-0.25*(1.0-COS(
     12.0*PY1*T/TAW)))/2.0
      OMEGA(I)=DELTH(I)*PY1*(SIN(PY1*T/TAW)-0.5*SIN(2.0*PY1*T/TAW))/
     1(2.0*TAW)
      ANACC(I)=DELTH(I)*(PY1/TAW)**2*(COS(PY1*T/TAW)-
COS(2.0*PY1*T/TAW))
     1/2.0
      RETURN
 104  THETA(I)=THZ(I)+DELTH(I)*(PY1*T/TAW-0.5*SIN(2.0*PY1*T/TAW))/PY1
      OMEGA(I)=DELTH(I)*(1.0-COS(2.0*PY1*T/TAW))/TAW
      ANACC(I)=2.0*DELTH(I)*PY1*SIN(2.0*PY1*T/TAW)/TAW**2
      RETURN
```

```
 105  THETA(I)=THZ(I)+10.0*DELTH(I)*(T/TAW)**3-15.0*DELTH(I)*(T/TAW)
     1**4+6.0*DELTH(I)*(T/TAW)**5
      OMEGA(I)=30.0*DELTH(I)*T**2/TAW**3-60.0*DELTH(I)*T**3/
     1TAW**4+30.0*DELTH(I)*T**4/TAW**5
      ANACC(I)=60.0*DELTH(I)*T/TAW**3-180.0*DELTH(I)*T**2/TAW**4
     1+120.0*DELTH(I)*T**3/TAW**5
      RETURN
 106  THETA(I)=THZ(I)+DELTH(I)*T**2*(3.0-2.0*T/TAW)/TAW**2
      OMEGA(I)=6.0*DELTH(I)*T*(1.0-T/TAW)/TAW**2
      ANACC(I)=6.0*DELTH(I)*(1.0-2.0*T/TAW)/TAW**2
      RETURN
 108  IF(III.EQ.1) GO TO 109
      DO 110 LJ=2,NT
      IF(III.EQ.LJ) GO TO 111
 110  CONTINUE
 109  WRITE(6,112) I
 112  FORMAT(' YOU HAVE CHOSEN TO GIVE DATA FOR JOINT(',I2,')
ANGULAR'/'
     1 VELOCITY. ENTER THE DATA ONE AFTER EACH ? (DATA IN RAD./SEC.)'/'
     2HIT RETURN KEY AFTER EACH VALUE IS ENTERED')
      DO 113 MN=1,NK
      READ(5,*) BK(I,MN)
 113  CONTINUE
      AK(I,1)=0.0
      AK(I,2)=DELT*(BK(I,2)+BK(I,1))/2.0
      DO 114 MN=3,NK
      AK(I,MN)=AK(I,MN-2)+DELT*(BK(I,MN-2)+4.*BK(I,MN-1)+BK(I,MN))/3.
 114  CONTINUE
      NM=NK-2
      DO 115 JJ=1,2
      CK(I,JJ)=(-3.*BK(I,JJ)+4.*BK(I,JJ+1)-BK(I,JJ+2))/(2.*DELT)
 115  CONTINUE
      LM=NK-1
```

```
      DO 116 MN=3,NM
      CK(I,MN)=(BK(I,MN-2)-8.*BK(I,MN-1)+8.*BK(I,MN+1)-BK(I,MN+2))/
     1(12.*DELT)
 116  CONTINUE
      DO 117 JJ=LM,NK
      CK(I,JJ)=(BK(I,JJ-2)-4.*BK(I,JJ-1)+3.*BK(I,JJ))/(2.*DELT)
 117  CONTINUE
      AK(I,NK+1)=AK(I,NK)
      BK(I,NK+1)=BK(I,NK)
      CK(I,NK+1)=CK(I,NK)
 111  THETA(I)=AK(I,III)
      OMEGA(I)=BK(I,III)
      ANACC(I)=CK(I,III)
      RETURN
      END
```

VITA

Hassan Aslrousta

Candidate for the Degree of

Master of Science

Thesis: THE KINEMATIC ANALYSIS OF AN N-R ROBOT ARM

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Karage, Iran, April 9, 1957, the son of Mr. and Mrs. Aslrousta.

Education: Graduated from Karage City High School in May, 1975; received Bachelor of Science in Mechanical Engineering degree in 1980 from Oklahoma State University; completed the requirements for the Master of Science Degree at Oklahoma State University in July 1982.

Professional Experience: Graduate teaching assistant, Oklahoma State University, 1980-1982.