UNIVERSITY OF JORDAN

FACULTY OF GRADUATE STUDIES

COMPUTER AIDED DYNAMIC ANALYSIS OF MECHANICAL SYSTEMS

BY

YOUSEF K. KH. SHASHANI

SUPERVISED BY

DR. MOHAMMAD H. F. DADO

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN MECHANICAL ENGINEERING, FACULTY OF GRADUATE STUDIES, UNIVERSITY OF JORDAN

NOVEMBER, 1993

THIS THESIS WAS DEFENDED SUCCESSFULLY ON 16/11/ 1993

COMMITTEE MEMBERS		SIGNATURE
1.	Dr. Mohammad H. F. Dado	....
2.	Dr. Sa'ad Habali	S. Habali.
3.	Dr. Mazen Al-Qaisi	M. Qaisi.

ACKNOWLEDGMENT

I would like to thank Dr. Mohammad Dado for his encouragement, guidance, and help during my work on this thesis, and for being patient with me.

A special thank for the people in the computer lab for their cooperation.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure		Page

LIST OF APPENDIXES

LIST OF SYMBOLS

Symbol	Description
a, b, c, d, L_C	Constant; length
C	Coefficients in virtual work expression; coefficient of viscous damping
C_C	Constant; coefficient of critical viscous damping
dpsc(i)	The derivative of path constraint *i* with respect to time
dptc(i;j)	The derivative of *ptc(i,j)* with respect to time
dpttc(i)	The derivative of *pttc(i)* with respect to time
f	Loop constraint
fpc(i)	Driving force or torque
F	Constraints' residuals equations
g	Time dependent motion generator; local acceleration of gravity
h	Number of time dependent motion generators
J	Jocobian
K	Global stiffness matrix
K_e	Element stiffness matrix
L	Length
M	Number of varying coordinates
M_o	Number of primary freedoms
ntc	Constant; number of time dependent motion generators
nsc	Constant; number of time path and length constraints
$N_1, N_2,..N_6$	Elastic freedoms numbers
N_k	Number of elements in loop *k*
N_p	Number of active freedoms
psc(i)	Path and length constraint
ptc(i;j)	The partial derivative of *tc(i)* with respect to coordinate *j*
pttc(i)	The partial derivative of *tc(i)* with respect to time
P	Global loading vector; externally applied load
P_e	Element loading vector

Symbol	Description
q	Varying coordinates; number of closed loops
q_d	Independent varying coordinate
r	Virtual displacement
S	Path constraint
t, tm	Time
$tc(i)$	Time dependent motion generator
$usl(i)$	Externally applied load in the direction of freedom number i
w	Number of path and length constraints
W	Virtual work
X	Global deflection vector
X_e	Element deflection vector
α, γ	Angles
α	Angular acceleration
φ	Varying coordinate
θ	Orientation
θ_c	Constant; orientation
ω	Angular velocity

ABSTRACT

COMPUTER AIDED DYNAMIC ANALYSIS OF MECHANICAL SYSTEMS

Prepared by: Yousef K. Kh. Shashani

Supervised by: Dr. Mohammad H. F. Dado

A general purpose computer program that performs dynamic and static analysis of two dimensional mechanical systems is developed and presented in this thesis. The program uses automatic formulations of the kinematic and dynamic equations developed in an earlier report by Dr. Mohammad Dado [1].

The modeling technique used can describe any planar mechanism with arbitrary constraints and motion generators.

This program is capable of performing kinematic analysis, dynamic analysis, static analysis, and static equilibrium position analysis.

Kinematic analysis involves the computations of the positions, velocities and accelerations of the mechanism links and joints.

Two types of dynamic analysis are considered: forward and inverse analysis. In forward dynamic analysis, the driving forces and torques are known, but the motion they generate is unknown. In both types of analysis, the deflections and reactions at the mechanism joints are computed. In inverse dynamic analysis, the configuration of the mechanism can be determined by solving the kinematic equations directly. Static analysis involves the computation of the joints deflections and reactions for stationary structures.

In static equilibrium position analysis, the static configuration of the mechanism under the action of static forces and/or torques is unknown. This kind of problem is solved using the virtual work principle and an iterative procedure.

ABSTRACT

COMPUTER AIDED DYNAMIC ANALYSIS OF MECHANICAL SYSTEMS

Prepared by: Yousef K. Kh. Shashani

Supervised by: Dr. Mohammad H. F. Dado

A general purpose computer program that performs dynamic and static analysis of two dimensional mechanical systems is developed and presented in this thesis. The program uses automatic formulations of the kinematic and dynamic equations developed in an earlier report by Dr. Mohammad Dado [1].

The modeling technique used can describe any planar mechanism with arbitrary constraints and motion generators.

This program is capable of performing kinematic analysis, dynamic analysis, static analysis, and static equilibrium position analysis.

Kinematic analysis involves the computations of the positions, velocities and accelerations of the mechanism links and joints.

Two types of dynamic analysis are considered: forward and inverse analysis. In forward dynamic analysis, the driving forces and torques are known, but the motion they generate is unknown. In both types of analysis, the deflections and reactions at the mechanism joints are computed. In inverse dynamic analysis, the configuration of the mechanism can be determined by solving the kinematic equations directly. Static analysis involves the computation of the joints deflections and reactions for stationary structures.

In static equilibrium position analysis, the static configuration of the mechanism under the action of static forces and/or torques is unknown. This kind of problem is solved using the virtual work principle and an iterative procedure.

CHAPTER I

INTRODUCTION

Using computers and computer software in the area of machine design, the design engineer can test his design efficiently and accurately, before building a prototype, and thus reducing both time and cost needed to complete the design.

In this thesis, a general-purpose dynamic analysis program is developed. Part of the program was developed in an earlier report by Dr. Mohammad H. F. Dado [1]. The program is capable of performing kineto-elasto-static analysis of any planar mechanism. The modeling technique can describe any two-dimensional mechanism with arbitrary constraints and time-dependent motion generators. The great advantage of this program is in its continuous elastic model of the links. The configuration of the mechanism can be open-loop, closed loop, or mixed loop. This program utilizes a Graphic User Interface Environment in its input and output operations and requires no preprogramming, and an equation parser has been added to it, so that there will be no need for re-compilation. Thus the program is designed to be a stand-alone package.

The program is capable of performing kinematic analysis, dynamic analysis, static analysis and static equilibrium position analysis. Two types of dynamic analysis are considered, forward and inverse analysis. The program is implemented on a personal computer. Also the program is going to be used for the solution of some problems, where several case-studies are going to be considered, and the results will be discussed.

Type of work and Equipment needed:

All the work is going to be theoretical, and the equipment needed is an IBM PC/AT Personal Computer or compatible.

Survey:

Several general-purpose dynamic analysis programs were developed by various universities. In the late 1960's the University of Michigan started the development of two such programs, DRAM (Dynamic Response of Articulated Machinery) and ADAMS (Automatic Dynamic Analysis of Mechanical Systems) [2]. DRAM was developed for two-dimensional mechanical systems, while ADAMS was developed for three-dimensional mechanical systems. Langrangian equations of motion with constraints are the bases of the dynamic analysis of DRAM. DRAM is an interactive program which requires no equation solving and no preprogramming. The ADAM program follows the same procedure in the dynamic analysis.

Integrated Mechanisms Program (IMP) [3] is another advanced program in the area of dynamic analysis. IMP performs the dynamic analysis for three-dimensional closed-loop mechanisms. It utilizes matrix coordinate transformation through the joints of each loop to perform kinematic analysis. The concept of virtual work is used to complete the dynamic analysis and determine internal joint forces. IMP also performs static analysis in which it determines the equilibrium configuration of the mechanism under a given loading condition.

Burton Paul and A. Amin, at the University of Pennsylvania, developed a dynamic analysis program called Dynamic of Machine (DYMAC) [4]. It was developed for planar mechanisms and it can accommodate arbitrary constraints due to higher pairs and motion generators. The program uses Langrange's form of d'Alemberts principle to formulate the differential equations of motion.

In 1984, Dr. M. Dado developed a general-purpose dynamic analysis program [1]. It analysis two dimensional mechanisms with elastic links. The program accepts arbitrary constraints and time-dependent motion generators. It utilizes matrix displacement method in structural analysis to solve for the internal joints reactions and link deflections. Also, it uses the gross motion analysis procedure developed by C. Bagci [5] to solve the forward dynamic problem, where the driving forces are known and the mechanism motion parameters are unknowns. The program is capable of performing kineto-elasto-static analysis of any planar mechanism. This program is the basis of this thesis, with additions and modifications added to it , such as the ability to perform static equilibrium position analysis. Also, the program was re-written in the 'C' programming language and a powerful user interface has been developed.

CHAPTER II

THE MODELING TECHNIQUE

In this chapter, the modeling technique is discussed, in which the mechanism properties are described. The modeling technique was developed in an earlier report by Dr. Mohammad Dado [1]. There are two major parts in the modeling technique. The description of the mechanism topological characteristics and the description of its constraints, external loads, and driving forces and torques. Before we move to the modeling steps, some basic definitions must be discussed.

2.1 Definitions

Varying Coordinates: The configuration of the mechanism at any time is descried by its varying coordinates. These coordinates represent either angular or linear variables. They are labeled as φ_1, φ_2, ..., φ_M, where M is the number of varying coordinates.

Kinematic Closed-Loop: Any mechanism can be described by a closed loop or a combination of closed loops. Each loop consists of a series of connected links and it must be independent from other loops. Loop independence means that every loop must contain at least one moving link which is different from the links of other loops and this loop cannot be represented by a linear combination of other loops.

Primary and Ordinary Coordinates: the primary coordinate is the varying coordinate which is associated with a driving force or torque and its position, velocity, and acceleration are only determined by solving the dynamic equations of the system. And the ordinary coordinate is the varying coordinate which its position, velocity, and acceleration are determined using the constraints equations.

Dependent and Independent Coordinates: These terms are used in static equilibrium position analysis. Dependent coordinates can be determined by solving the loop constraints equations directly given the values for the independent coordinates. Independent coordinates are the necessary coordinates to determine the system configuration, their number is equal to the number of degrees of freedom of the system.

Links and Elements: Mechanisms consists from combination of links and kinematic pairs. A link may have two or more ends on which kinematic pairs can be located. A link also has a mass and a mass moment of inertia. An element is a weightless, straight-line which is used to construct a link in combination with other elements.

Elastic Elements: Elastic elements are those elements which experience deflections at their ends when loaded. The elastic properties of these elements are: length, orientation, cross-section area and its area moment of inertia, and modulus of elasticity.

Element Freedoms: The element freedoms describe all possible elastic motions which can be experienced by both ends of an elastic element. At each end there are two linear freedoms and one rotational freedom. The coordinate system describing these freedoms can have any relative orientation with respect to a fixed frame. The element freedoms are numbered locally from 1 to 6 and they are given global numbers also. The inertial and external loads are applied in the direction of these freedoms.

Time-dependent Constraints: Time-dependent constraints relate one varying coordinate or more to time. These relations can be established through the velocity and acceleration with which certain coordinates are driven. These constraints are provided in equation form and they are specified by the user of the program.

Spatial Constraints: Spatial constraints relate the varying coordinates with each other. These constraints arise when, for example, some point in the mechanism has to trace a defined path or the length of some elements are related by certain equalities.

Applied Loads: The applied loads are the set of forces and torques that the mechanism is subjected to. They are applied in the direction of elemental freedoms. They can be functions of time or the positions and velocities of certain coordinates. They allow the introduction of spring forces and dampers.

Known Driving Forces and Torques: The known driving forces and torques are associated with the primary coordinates. They are applied in the direction of elemental freedoms, where the motion described by the primary coordinates is in the same direction of the elastic motion represented by the elemental freedoms.

Kinematic analysis: In Kinematic analysis, the values of the varying coordinates are determined at a given instant of time along with their first and second time derivatives. The positions, velocities, and accelerations of specified points on the linkage are also determined.

Inverse Dynamic Analysis: In inverse dynamic analysis the linkage is driven with known motion generators. The kinematic parameters can be solved for at any specified time along with the inertial loads. The purpose of this analysis is to determine the joint relations, element internal loads, and element deflections.

Forward Dynamic Analysis: Forward dynamic analysis is similar to the inverse dynamic analysis in its final goal. However, the motion generators which are driving the linkage are unknowns. Instead, the driving forces or torques are known functions of time or linkage parameters.

Static Analysis: Static analysis is similar to structural analysis where the joint and ground reactions are determined for a stationary frame subjected to a system of external and internal loads.

Static equilibrium position analysis: In static equilibrium position analysis, the static configuration of a system of rigid bodies, subjected to a set of constraints and acted upon by static loads, is determined.

2.2 Modeling Steps

The modeling technique is divided into twelve steps. These steps are illustrated through several examples. The following is the discussion of these steps.

Step 1: Identify the Mechanism Coordinates

The coordinates are those parameters that describe the configuration of the mechanism at any time. Consider the mechanism shown in Figure (1), there are five coordinates. Coordinates φ_1-φ_4 describe the angular orientations of links a, b, c and d with respect to the fixed frame x-y. Coordinate φ_5 represents the position of the slider with respect to a fixed point along its path. The order in which the coordinates

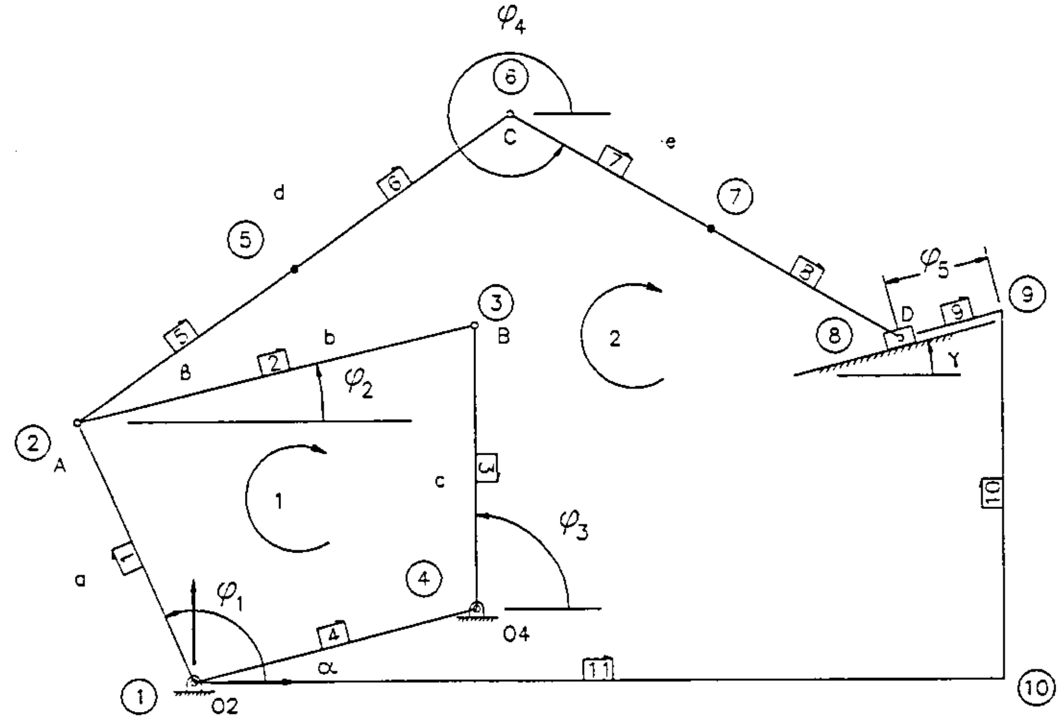

Figure (1) A Six-bar Mechanism and Description of the Elements [1].

are numbered is arbitrary if kinematic and inverse dynamic analysis are performed. In forward dynamic analysis the primary coordinates are numbered last. For example, if the mechanism in Figure (1) is driven with known torque at O_2, the coordinate representing the angular position of link a will be φ_5.

Step 2: Identify the elements

An element of a mechanism is identified by a number and direction. It does not have to be a physical element. It can be introduced in a manner that provides convenience in description. In Figure (1), elements 9, 10, and 11 are introduced to conveniently describe the mechanisms closed-loops.

Step 3: Identify the Closed Loops

Any mechanism can be represented by one or more closed loops. For open loop mechanisms or robots, the loops are closed by virtual elements which may vary in length and/or orientation. The virtual elements' lengths and orientations are described by coordinates which satisfy the constraints equations of the mechanism or robot. For convenience the loops are always traced in the clockwise direction. Consider the mechanism shown in Figure (1), there are two closed loops. Loop 1 consists of elements 1, 2, 3, and 4. Loop 2 consists of links 1, 5, 6, 7, 8, 9, 10, and 11.

Step 4: Identify the Joints of the Mechanism

The elements of the mechanism are connected by joints. These joints are identified using numbers which may be assigned in any convenient order. Joints may be introduced within any link as a point of interest, where masses and mass moments of inertia of the links can be lumped or information are needed. These information are like position, velocity, acceleration, deflection, and internal forces at the joint location. For the mechanism shown in Figure (1), there are 9 joints. Joints 5 and 7 represent points of interest at links d and e.

Step 5: Identify the Elastic Freedoms

Elastic freedoms describe the elastic motions at the ends of each elastic element. The assignment of freedoms depends on the kinematic pair and type of constraints present at the ends of the element. Figure (2) shows different types of kinematic and

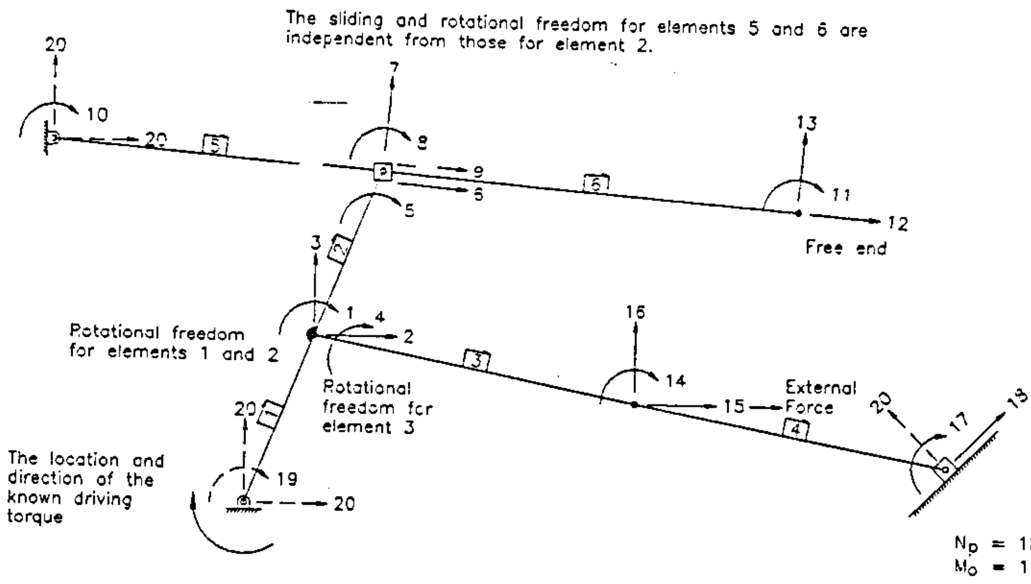

Figure (2) Types of Freedoms for Deferent Types of Joints. Broken-Line Arrows are restrained Freedoms [1].

rigid pairs and how the elastic freedoms are assigned. The numbering of these freedoms plays an important role. The unrestrained freedoms which are also called active freedoms are numbered first from 1 to N_p , where N_p is the number of active freedoms. The primary freedoms in which direction the driving forces and torques are applied are numbered next from $N_p + 1$ to $N_p + M_o$ where M_o is the number of primary freedoms. Finally, the completely restrained freedoms are numbered with the same number which is $N_p + M_o + 1$. If virtual elements are present, their elastic freedoms need not be considered and their numbers are taken as 0.

Step 6: Identify the Elastic Properties of the Elements

The elastic properties of an element consist from its length, cross-section area and area moment of inertia, and modulus of elasticity. These properties must be provided for each element.

Step 7: Identify the Inertial Properties of the system

The mass and mass moment of inertia for each link are introduced by lumping them at the proper location within the link. The link with the lumped mass must be dynamically equivalent to the actual link. This dynamic equivalence is achieved if the following conditions are satisfied.

1. The mass of the link is equal to the lumped mass.
2. The center of mass of the link coincides with the location of the lumped mass.
3. The mass-moment of inertia about the center of mass is assumed to be the lumped mass-moment of inertia.

Constructing the link from two or more elements allows these conditions to be satisfied.

Step 8: Model the Moving Slider Path

In some mechanisms with prismatic joints, the path of the slider may be a moving link. Modeling such mechanisms for dynamic analysis using the line element technique is done by the following procedure. Treat the straight portions of the moving link between the slider location and any other joint on the path as separate links. These links are variable in length and orientation. Lump the masses and mass moment of inertia at the proper locations for these variable length links. Consider the length of each element in these links as a varying coordinate. The lumped inertia is also varying and it is determined by these varying coordinates. The variable lengths of the elements are related by length constraints equations which must be introduced in the user constraints. To illustrate this procedure, consider the inverted slider crank mechanism shown in Figure (3). The moving slider path represented by a varying coordinate. These coordinates are related by three equations. These equations are:

$$\varphi_3 + \varphi_4 + \varphi_5 + \varphi_6 = L_2 \ , \qquad \text{(a)}$$

$$\varphi_3 = \varphi_4 \ , \qquad \text{(b)}$$

and

$$\varphi_5 = \varphi_6 \qquad \text{(c)}$$

Equations b and c are obtained by considering that the inertia is lumped at the midpoints of the variable length links.

Step 9: Model The Time-Dependent Motion Generators

Mechanisms can be driven with time-dependent motion generators where the motion of the input parameters are known functions of time. The most general form of these functions is

$$g_e(\varphi_1, \varphi_2, ..., \varphi_M, t) = 0 \ , \ e = 1,2,...,h.$$

where h is the number of time-dependent motion generators constraints equations. The functions g and their partial derivatives with respect to φ's and t and the total time derivatives must be supplied by the user. This way of modeling allows the use of general functions and do not restrict the user to any specified form.

Step 10: Model Path and Length Constraints

Path and length constraints relates the varying coordinates with each other. These relations are given in equation form. The most general form is

$$S_r = (\varphi_1, \varphi_2, ..., \varphi_M) = 0 \ , \ r = 1,2,...,w.$$

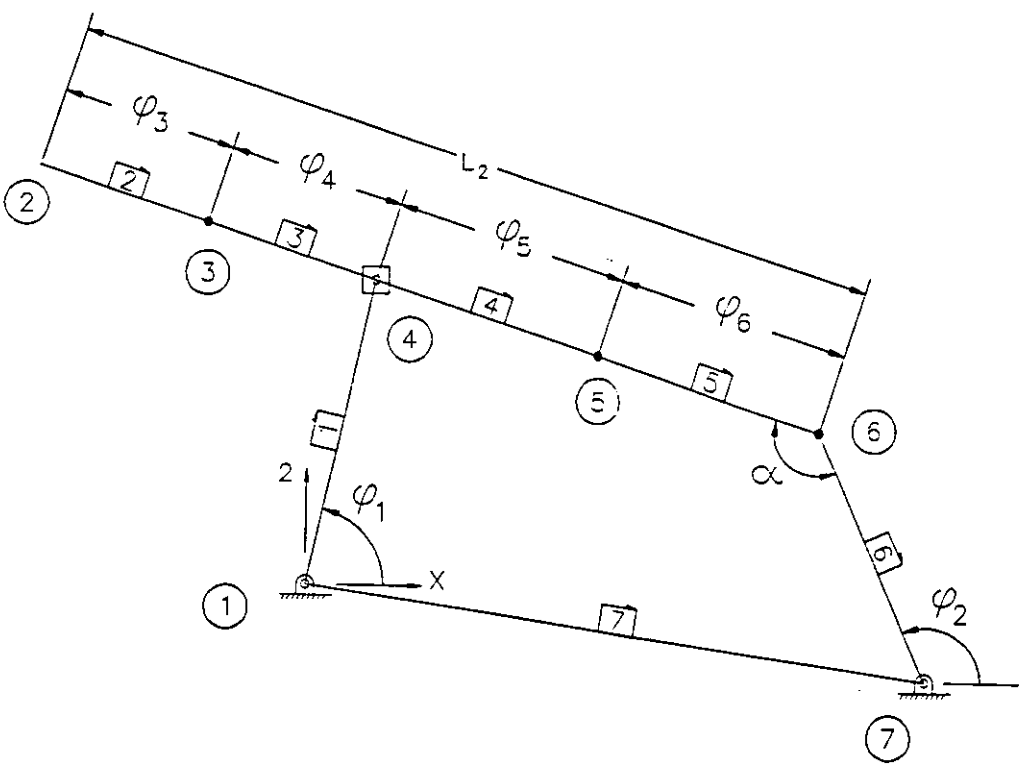

Figure (3) An Inverted Slider-Crank Mechanism That Illustrates the Occurrence of Variable Length Elements [1].

where w is the number of such constraints equations. The functions S_r and their partial derivatives with respect to φ's and the total time derivatives of those partial derivatives must be provided by the user.

Step 11: Identify the Externally Applied Loads

Mechanisms may be subjected to external loads. The magnitudes and directions of the external loads may depend on time, coordinates positions and velocities. Spring and damper forces are modeled by the use of external loads. The external loads are applied at joints and their directions are defined by freedoms present at these joints. The functions representing the applied loads are identified by numbers of these freedoms. For the mechanism shown in Figure (2), let the force $F = 5 - 2t$ lb be applied at the midpoint of the connecting rod and in the horizontal direction. Then at freedom #15 the external load is given as

$$P = 5 - 2t$$

Step 12: Model the Driving Forces and Torques

The location and direction of the driving forces and torques are defined by the mechanism primary coordinates and primary elastic freedoms. The function representing the driving force and torque can be in terms of time, position, and velocity of the varying coordinates. This allows the introduction of preloaded springs as driving forces or torques.

These are the modeling steps for a general problem which includes all the features of the program. Actual problems need a certain combination of these steps, for instance, the kinematic analysis requires only steps 1,2,3,4,9 and 10. In static analysis only steps 1,2,5,6 and 11 should be considered.

CHAPTER III

THE SOLUTION TECHNIQUE

In this chapter, four types of analysis are discussed: Kinematic analysis, dynamic analysis, static analysis and static equilibrium position analysis. Two types of dynamic analysis problems are considered; forward and inverse analysis . Static analysis is a part of the inverse dynamic analysis. The solution techniques to these problems are discussed in the following sections.

3.1 Kinematic Analysis [1]

The number of varying coordinates that describe the motion of a mechanism and the number of constraints equations must be equal. There are three types of constraints in the kinematic analysis:- (1) closed-loop constraints, (2) time-dependent motion generators constraints, (3) spatial constraints. The equations representing these constraints are expressed in terms of the varying coordinates. The solution of these equations and the solution of the equations representing their partial derivatives with respect to the varying coordinates and time, provide the positions, velocities and accelerations of the moving links. The following subsections discuss the formulation of the constraint equations for each type of constraint.

3.1.1 Loop Constraints Equations and Their Derivatives

Consider any loop k in a mechanism, let this loop consist of N_k elements. Each element has its own length and orientation with respect to a fixed frame of reference, see Figure (4). L_i and θ_i are the length and orientation of the i -th element. For the k-th loop there are two constraints equations. These equations are:

$$f_{1k} = \sum_{i=1}^{N_k} L_i \cos\theta_i = 0 \qquad (1)$$

and

$$f_{2k} = \sum_{i=1}^{N_k} L_i \sin\theta_i = 0 \qquad (2)$$

For a general element i the length and orientation are given as

$$L_i = L_{c_i} \pm \varphi_n \qquad (3)$$

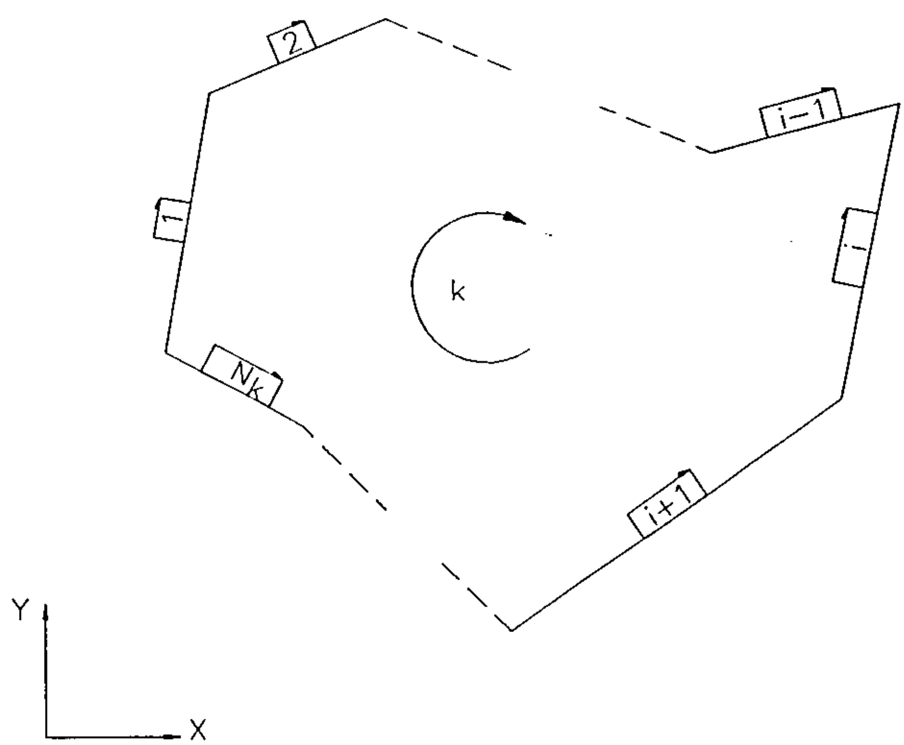

Figure (4) A General Form of a Closed-Loop[1].

and

$$\theta_i = \theta_{c_i} \pm \varphi_m \tag{4}$$

where L_{c_i} and θ_{c_i} are the constant components of the length and orientation respectively, φ_n and φ_m are the varying components of the length and orientation. φ_n and φ_m refer to the varying coordinates describing the configuration of the mechanism. To illustrate the loop constraints equations, consider the mechanism shown in Figure (5). Loop 1 consists from elements 1,5,6,7, and 8. The varying coordinates of the mechanism are $\varphi_1, \varphi_2, ..., \varphi_5$. The length of element 1 is

$$L_1 = a + 0$$

and its orientation is

$$\theta_1 = 0 + \varphi_1$$

For element 3

$$L_3 = b + 0$$

and

$$\theta_3 = 180^o + \varphi_2$$

For element 5

$$L_5 = e + \varphi_3$$

and

$$\theta_5 = (180^o - \alpha) + \varphi_2$$

The partial derivatives of the closed-loop constraints equations with respect to the varying coordinate assist in solving for the positions, velocities, and accelerations of the links of the mechanism. For the k-th closed-loop these partial derivatives are obtained as

$$\frac{\partial f_{1k}}{\partial \varphi_n} = \sum_{i=1}^{N_k} \delta_{in} \cos \theta_i \tag{5}$$

$$\frac{\partial f_{2k}}{\partial \varphi_n} = \sum_{i=1}^{N_k} \delta_{in} \sin \theta_i \tag{6}$$

$$\frac{\partial f_{1k}}{\partial \varphi_m} = \sum_{i=1}^{N_k} -\delta_{im} L_i \sin \theta_i \tag{7}$$

$$\frac{\partial f_{2k}}{\partial \varphi_m} = \sum_{i=1}^{N_k} \delta_{im} L_i \cos \theta_i \tag{8}$$

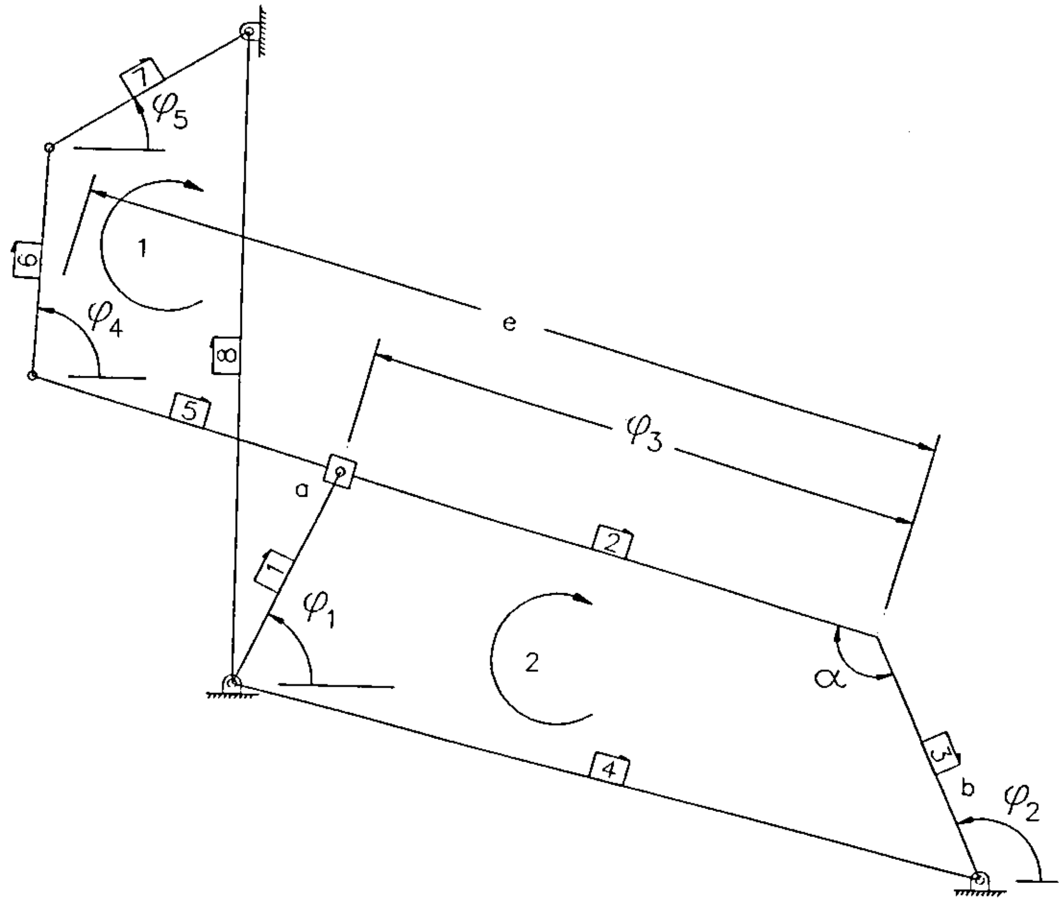

Figure (5) An Eight-bar Mechanism That Illustrates the Model Parameters [1].

where δ_{in} is 1 if the i-th element has φ_n as its varying component for L_i and similarly δ_{im} is 1 if the i-th element has φ_m as its varying component for θ_i . δ_{in} and δ_{im} are 0 otherwise.

The total time derivatives of the partial derivatives of the closed-loop constraints equations is needed to obtain the accelerations of the varying coordinates. Those total time derivatives are obtained by differentiating equation 5 through 8 with respect to time. They are obtained as

$$\frac{d}{dt}\frac{\partial f_{1k}}{\partial \varphi_n} = \sum_{i=1}^{N_k} -\delta_{in}\dot{\varphi}_n \sin\theta_i \tag{9}$$

$$\frac{d}{dt}\frac{\partial f_{2k}}{\partial \varphi_n} = \sum_{i=1}^{N_k} \delta_{in}\dot{\varphi}_n \cos\theta_i \tag{10}$$

$$\frac{d}{dt}\frac{\partial f_{1k}}{\partial \varphi_m} = \sum_{i=1}^{N_k} -\delta_{im}[\delta_{in}\dot{\varphi}_n \sin\theta_i + L_i\dot{\varphi}_m \cos\theta_i] \tag{11}$$

$$\frac{d}{dt}\frac{\partial f_{2k}}{\partial \varphi_m} = \sum_{i=1}^{N_k} \delta_{im}[\delta_{in}\dot{\varphi}_n \cos\theta_i - L_i\dot{\varphi}_m \sin\theta_i] \tag{12}$$

3.1.2 Time-Dependent Motion Generators and Their Partial Derivatives

The time-dependent motion generators constraints equations are general functions of time. An example of this type of constraints is the motion of the input link of the mechanism shown in Figure (5) which could be given as

$$g(\varphi_1,t)= \omega_1 t + \frac{1}{2}\alpha_1 t^2 + \varphi_{1_o} - \varphi_1 = 0$$

where ω_1 and α_1 are the angular velocity and angular acceleration of the input link and φ_{1_o} is the initial value of φ_1.

The partial derivatives of these constraints equations with respect to time is specified by the user. Therefore

$$\frac{\partial g}{\partial \varphi_1} = -1$$

and

$$\frac{\partial g}{\partial \varphi_r} = 0 , \qquad r = 2,\ldots,M$$

Also

$$\frac{\partial g}{\partial t} = \omega_1 + \alpha_1 t$$

The total time derivatives of the partial derivatives are also specified by the user. For the illustrative example they are

$$\frac{d}{dt}\frac{\partial g}{\partial \varphi_1} = 0$$

$$\frac{d}{dt}\frac{\partial g}{\partial t} = \alpha_1$$

3.1.3 Spatial and Length Constraints and Their Partial Derivatives

The path and length constraints equations are general functions relating the varying coordinates with each other. An example for the path constraint is the specified motion of the end effector of a robot. Figure (6) shows a 2r robot with its end effector tracing an elliptical path. The coordinates φ_3 and φ_4 are related by

$$S(\varphi_3,\varphi_4) = \frac{(\varphi_3 - X_0)^2}{a^2} + \frac{(\varphi_4 - Y_0)^2}{b^2} - 1 = 0$$

The derivatives of the path constraints equations with respect to the coordinates and the total time derivatives of those partial derivatives are specified by the user. For the example shown in Figure (6) the partial derivatives are

$$\frac{\partial S}{\partial \varphi_3} = 2b^2(\varphi_3 - X_0)$$

$$\frac{\partial S}{\partial \varphi_4} = 2a^2(\varphi_4 - Y_0)$$

In the case of path and length constraints there is no partial time derivatives. The total time derivatives are

$$\frac{d}{dt}\frac{\partial S}{\partial \varphi 3} = 2b^2\dot{\varphi}_3$$

$$\frac{d}{dt}\frac{\partial S}{\partial \varphi_4} = 2a^2\dot{\varphi}_4$$

The length constraints equations are like those given in the example in Chapter II where the varying coordinates of the mechanism shown in Figure (3) is related with 3 constraint equations.

These are the three types of constraints considered in the kinematic analysis. The resulting set of equations relating the varying coordinates and their derivatives are arranged in matrix form to solve for the positions, velocities, and accelerations of the various links of the mechanism.

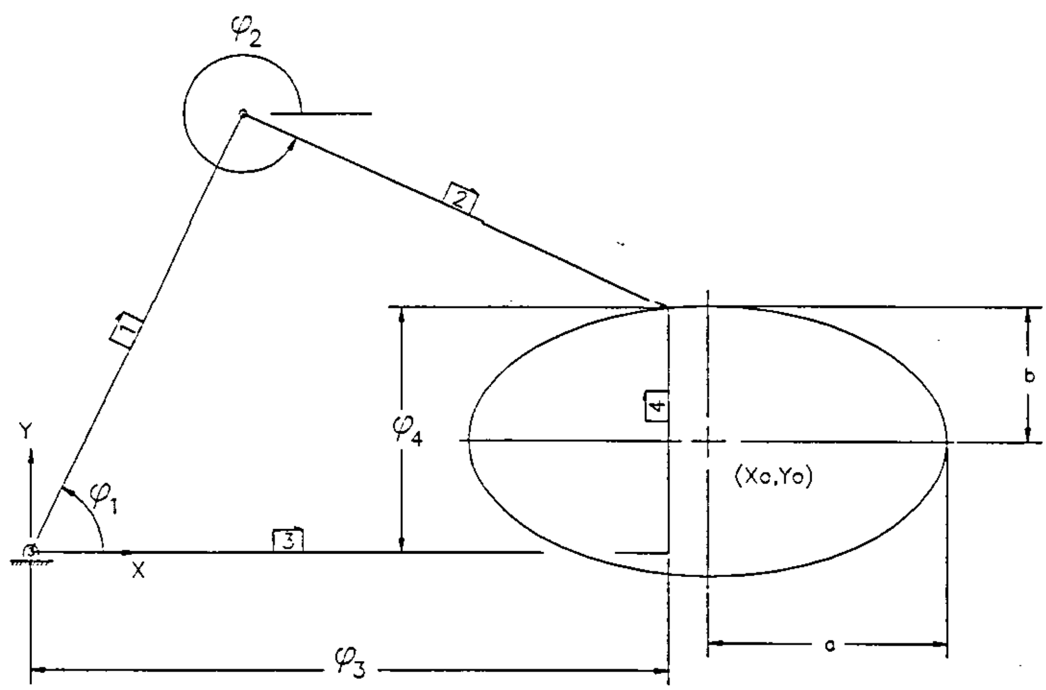

Figure (6) A 2R Robot to Illustrate the Use of Path Constraints and Virtual Elements [1].

3.1.4 Displacement Analysis

The equations describing the constraints of a general mechanism are non-linear algebraic equations. An iterative procedure is used to solve these equations for the values of the varying coordinates. Newton-Raphson technique in solving non-linear equations is applied. This technique requires the formation of a matrix containing the partial derivatives of the residual equations with respect to each unknown parameter. This matrix is called the Jacobian matrix $[J]$. Arrange the constraints equations in the following order; closed-loop, time-dependent motion generators, and spatial and length constraints. Then the Jacobian matrix becomes

$$[J]_{M\times M}=\begin{bmatrix}
\frac{\partial f_{11}}{\partial\varphi_1} & \frac{\partial f_{11}}{\partial\varphi_2} & -- & -- & \frac{\partial f_{11}}{\partial\varphi_M}\\
\frac{\partial f_{21}}{\partial\varphi_1} & \frac{\partial f_{21}}{\partial\varphi_2} & -- & -- & \frac{\partial f_{21}}{\partial\varphi_M}\\
 & & & & \\
\frac{\partial f_{1k}}{\partial\varphi_1} & \frac{\partial f_{1k}}{\partial\varphi_2} & -- & -- & \frac{\partial f_{1k}}{\partial\varphi_M}\\
\frac{\partial f_{2k}}{\partial\varphi_1} & \frac{\partial f_{2k}}{\partial\varphi_2} & -- & -- & \frac{\partial f_{2k}}{\partial\varphi_M}\\
 & & & & \\
\frac{\partial f_{1q}}{\partial\varphi_1} & \frac{\partial f_{1q}}{\partial\varphi_2} & -- & -- & \frac{\partial f_{1q}}{\partial\varphi_M}\\
\frac{\partial f_{2q}}{\partial\varphi_1} & \frac{\partial f_{2q}}{\partial\varphi_2} & -- & -- & \frac{\partial f_{2q}}{\partial\varphi_M}\\
\frac{\partial g_{1}}{\partial\varphi_1} & \frac{\partial g_{1}}{\partial\varphi_2} & -- & -- & \frac{\partial g_{1}}{\partial\varphi_M}\\
 & & & & \\
\frac{\partial g_{L}}{\partial\varphi_1} & \frac{\partial g_{L}}{\partial\varphi_2} & -- & -- & \frac{\partial g_{L}}{\partial\varphi_M}\\
 & & & & \\
\frac{\partial g_{h}}{\partial\varphi_1} & \frac{\partial g_{h}}{\partial\varphi_2} & -- & -- & \frac{\partial g_{h}}{\partial\varphi_M}\\
\frac{\partial S_{1}}{\partial\varphi_1} & \frac{\partial S_{1}}{\partial\varphi_2} & -- & -- & \frac{\partial S_{1}}{\partial\varphi_M}\\
 & & & & \\
\frac{\partial S_{r}}{\partial\varphi_1} & \frac{\partial S_{r}}{\partial\varphi_2} & -- & -- & \frac{\partial S_{r}}{\partial\varphi_M}\\
 & & & & \\
\frac{\partial S_{w}}{\partial\varphi_1} & \frac{\partial S_{w}}{\partial\varphi_2} & -- & -- & \frac{\partial S_{w}}{\partial\varphi_3}
\end{bmatrix} \qquad (13)$$

where q is the number of closed-loops, h is the number of time- dependent motion generators, w is the number of path and length constraints, and M is the number of varying coordinates. To have a determinate system of equations, the following equation must be satisfied

$$M = 2q + h + w \tag{14}$$

The elements of the Jacobian matrix are evaluated using an initial estimate for the unknowns. Also, the residuals are evaluated using the same estimates. The corrected values of the unknowns are found using the following iteration equations

$$[J]^{(j)}\{\Delta\varphi\} = -\{F\}^{(j)} \tag{15}$$

and

$$\{\varphi\}^{(j+1)} = \{\varphi\}^{(j)} + \{\Delta\varphi\} \tag{16}$$

where the vector $\{\Delta\varphi\}$ is the correction vector, $\{F\}^{(j)}$ is the residuals evaluated at the j-th iteration, $\{\varphi\}^{(j)}$ is the unknown vector at the j-th iteration, and $\{\varphi\}^{(j+1)}$ is the corrected unknown vector. At the initial iteration j=0. The iteration continues using equation 15 and 16 until all the magnitudes of the elements of the correction vector $\{\Delta\varphi\}$ are less than an allowed tolerance.

In this iterative procedure the initial estimates of the varying coordinates decides which geometric inversion the mechanism will follow in the analysis. Different initial estimates must be used to analyze different geometric inversions. These estimates are obtained by sketching the mechanism at the desired geometric inversion.

3.1.5 Velocity Analysis

The equations used for the velocity analysis are linear algebraic equations. The resulting system of equations is

$$[J]_{M\times M}\{\dot{\varphi}\}_{M\times 1} = \begin{Bmatrix} \{0\}_{2q\times 1} \\ \left\{-\dfrac{\partial g}{\partial t}\right\}_{h\times 1} \\ \{0\}_{w\times 1} \end{Bmatrix} \tag{17}$$

The system of equations 17 is linear and can be solved by Gauss-Jordan elimination for the velocity vector $\{\dot{\varphi}\}$.

3.1.6 Acceleration Analysis

The acceleration vector $\{\ddot{\varphi}\}$ can be obtained by taking the total time derivatives of both sides of equation 17. Thus, the acceleration vector is given by

$$[\dot{J}]_{M\times M}\{\dot{\varphi}\}_{M\times 1}+[J]_{M\times M}\{\ddot{\varphi}\}_{M\times 1}=\begin{Bmatrix}\{0\}_{2q\times 1}\\ -\dfrac{d}{dt}\left\{\dfrac{\partial g}{\partial t}\right\}_{h\times 1}\\ \{0\}_{w\times 1}\end{Bmatrix} \qquad (18)$$

where the elements of $[\dot{J}]$ are the total time derivatives of the elements of the Jacobian Matrix. Rearrange equation 18 to obtain

$$[J]_{M\times M}\{\ddot{\varphi}\}_{M\times 1}=\begin{Bmatrix}\{0\}_{2q\times 1}\\ -\dfrac{d}{dt}\left\{\dfrac{\partial g}{\partial t}\right\}_{h\times 1}\\ \{0\}_{w\times 1}\end{Bmatrix}-[\dot{J}]_{M\times M}\{\dot{\varphi}\}_{M\times 1} \qquad (19)$$

The acceleration vector $\{\ddot{\varphi}\}$ can be solved for using equation 19 by Gauss-Jordan elimination technique.

3.2 Inverse Dynamic Analysis [1]

In inverse dynamic analysis [6] the functional relationships between the motion of the input and time are known. These relationships can be introduced in the constraints equations under the time-dependent motion generators. Thus, the configuration at any point on time is defined and the coordinates' positions, velocities, and accelerations are known. After defining the coordinates of the mechanism, the line element model of the mechanism is used to solve for the joint deflections and internal forces [7]. The following sections present the details of the inverse dynamic analysis.

3.2.1 Line Element [1]

The line element used to model the links of the mechanism have constant cross-sectional properties. The elements are connected by joints. Each joint experiences three freedoms consisting of two linear freedoms and one rotational freedom. Figure (7) shows an element with its initial and terminal joints and their freedoms. There are two types of freedoms, active freedoms and inactive freedoms. The active freedoms describe the possible elastic displacements of the mechanism. The inactive freedoms specify the motion restraints at the element end. Inertial and external loads

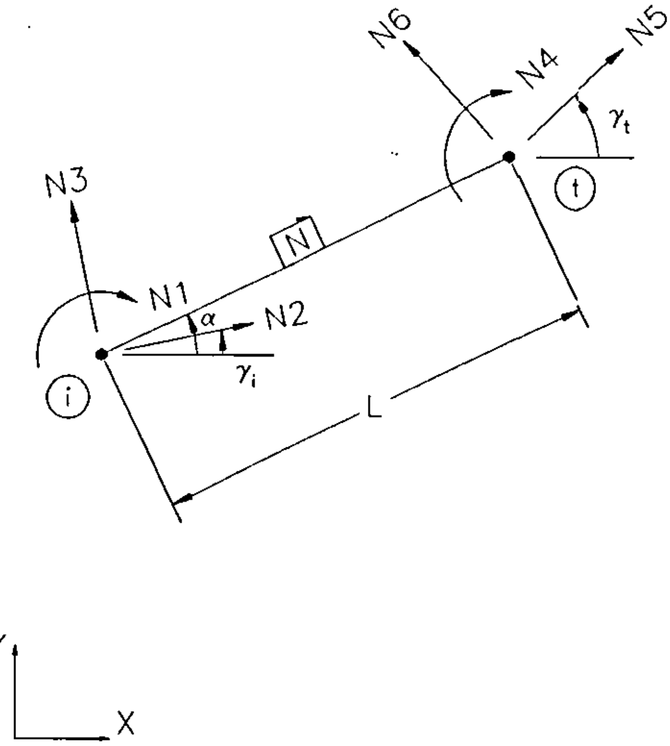

Figure (7) The Parameters of a General Element [1].

are applied in the direction of the active freedoms. The freedoms of each element are given global numbers $N_1, N_2, \ldots, N_6$ in the order shown in Figure (7). The coordinate systems describing the initial and terminal joint freedoms can have any orientation with respect to the fixed reference coordinate system.

The properties and orientation of the elements relates displacements in the direction of the freedoms and the loads applied at the two ends of the element. This relation is summarized in the following equation:

$$\{Pe\}_{6\times1} = [Ke]_{6\times6}\{Xe\}_{6\times1} \tag{20}$$

where $\{Pe\}$ is the vector containing the six loads applied in the direction of the six freedoms of the element, $[Ke]$ is the element stiffness matrix, and $\{Xe\}$ is the vector containing the six displacements in the direction of the six freedoms. The stiffness matrix $[Ke]$ is the product of three matrices. Hence

$$[Ke]_{6\times6} = [A]_{6\times3}[S]_{3\times3}[A]^T_{3\times6} \tag{21}$$

where $[A]$ is the element statics matrix and $[S]$ is the element internal stiffness matrix. $[A]$ and $[S]$ are defined as

$$[A]_{6\times3} = \begin{bmatrix} 0 & 1 & 0 \\ -\cos(\alpha-\gamma_i) & \dfrac{\sin(\alpha-\gamma_i)}{L} & \dfrac{\sin(\alpha-\gamma_i)}{L} \\ -\sin(\alpha-\gamma_i) & \dfrac{-\cos(\alpha-\gamma_i)}{L} & \dfrac{-\cos(\alpha-\gamma_i)}{L} \\ 0 & 1 & 0 \\ \cos(\alpha-\gamma_t) & \dfrac{-\sin(\alpha-\gamma_t)}{L} & \dfrac{-\sin(\alpha-\gamma_t)}{L} \\ \sin(\alpha-\gamma_t) & \dfrac{\cos(\alpha-\gamma_t)}{L} & \dfrac{\cos(\alpha-\gamma_t)}{L} \end{bmatrix} \tag{22}$$

and

$$[S]_{3\times3} = \begin{bmatrix} \dfrac{EA}{L} & 0 & 0 \\ 0 & \dfrac{4EI}{L} & \dfrac{4EI}{L} \\ 0 & \dfrac{4EI}{L} & \dfrac{4EI}{L} \end{bmatrix} \tag{23}$$

where α is the orientation of the element with respect to the reference frame, γ_i and γ_t are the orientations of the initial and terminal ends freedoms coordinate systems with respect to the reference frame, L is the length of the element, A and I are the cross-sectional area and area moment of inertia of the element, and E is modulus of elasticity of the element material. For more detailed discussion of these element matrices, examine reference [2].

3.2.2 The Global System

The stiffness matrices of all the line elements contained in the mechanism are superimposed on each other through the global freedom numbers of each element $(N_1, N_2, \ldots, N_6)$. Also the elements loading vectors and elements displacements vector are superimposed on each other. This superimposition procedure produces the following global system of equations:

$$\{P\}_{N_p \times 1} = [K]_{N_p \times N_p} \{X\}_{N_p \times 1} \tag{24}$$

where $\{P\}$ is the global loading vector, $[K]$ is the global stiffness matrix, and $\{X\}$ is the global elastic displacement vector. The loading vector $\{P\}$ is a known vector. This is true because the loads of the individual elements are superimposed on each other at their respective joints and at the same time all the joints are in equilibrium. Therefore, the elements' loads cancel each other at their common joints and the only remaining loads are those loads that are externally applied on the mechanism or the inertial loads due to the motion of the mechanism. The global stiffness matrix is also a known matrix since it is the sum of the known individual element stiffness matrices. Equation 24 can be solved for $\{X\}$ using Gauss-Jordan elimination.

3.2.3 Element Forces and Moments

After finding the global elastic displacements vector $\{X\}$, the element elastic displacements $\{Xe\}$ can be obtained. They are extracted from $\{X\}$ by their global freedom numbers. Knowing the individual element elastic displacement vector, the elemental loads $\{Pe\}$ can be found using equation 20. Through these elemental forces and moments, the joint reactions and the unknown driving forces and torques are obtained.

3.3 Static Analysis [1]

The static analysis considered in this program relates the applied loads, the nodal deflections, and elemental internal forces and moments. This type of analysis is used for structures where there are no inertial loads and the global stiffness matrix is constant. The analysis is performed by generating the global stiffness matrix of the structure using its line element model and the procedure outlined in Section 3.2.2. The global loading vector $\{P\}$ is defined by the loading condition of the structure. Equation 24 can be directly used to solve for the global deflection vector $\{X\}$. The element internal forces and moments are solved for using the procedure explained in Section 3.2.3 and equation 20.

3.4 Forward Dynamic Analysis [1]

In forward dynamic analysis [6], the driving forces and torques are known, but the motion they generate in unknown. To solve this problem, a coordinate integration procedure is used . In the case of forward dynamic analysis there are two types of coordinates, ordinary and primary coordinates. The solution starts with known initial positions and velocities for the primary coordinates. The initial positions for the ordinary coordinates are estimated and iteratively computed. Using the initial state of the mechanism, the initial accelerations of the primary coordinates are computed. These accelerations are then integrated through the time step of the analysis to find the new positions and velocities of the coordinates at the beginning of the next time step. The same procedure is repeated through the time steps until the time of the analysis expires. The procedure through which the initial accelerations of the primary and ordinary coordinates are obtained at the beginning of each time step is discussed in the following steps:

Step 1: Express the accelerations of all the varying coordinates in terms of the primary coordinates accelerations.

Consider equation 19 in Section 3.1.6 and suppose that there are m_o ordinary coordinates and m_p primary coordinates. The number of available constraints equations is only m_o and the number of coordinates is $M = m_o + m_p$. The size of the

Jacobian matrix $\left[J\right]$ and its total time derivatives, matrix $\left[\dot{J}\right]$ is m_oxM. Numbering the primary coordinates last and partitioning equation 19 into ordinary and primary coordinates, one obtains

$$\left[\left[J_o\right]_{m_o \times m_o}\left[J_p\right]_{m_o \times m_p}\right]\begin{Bmatrix}\ddot{\varphi}_{o_{m_o \times 1}}\\ ---\\ \ddot{\varphi}_{p_{m_p \times 1}}\end{Bmatrix}=\begin{Bmatrix}0\\ -\dfrac{d}{dt}\dfrac{\partial g}{\partial t}\\ 0\end{Bmatrix}_{m_o \times 1}-\left[\dot{J}_p\right]_{m_o \times m_p}\left\{\dot{\varphi}_p\right\}_{m_p \times 1}-\left[\dot{J}_o\right]_{m_o \times m_o}\left\{\dot{\varphi}_o\right\}_{m_o \times 1} \tag{25}$$

where $\left[J_o\right]_{m_o \times m_o}$ contains the first m_o columns of the Jacobian matrix of the system and corresponds to the ordinary coordinates, $\left[J_p\right]_{m_o \times m_p}$ contains the last m_p columns of the Jacobian matrix and corresponds to the primary coordinates, $\left\{\ddot{\varphi}_o\right\}_{m_o \times 1}$, $\left\{\dot{\varphi}_o\right\}_{m_o \times 1}$, are the accelerations and velocities of the ordinary coordinates, $\left\{\ddot{\varphi}_p\right\}_{m_p \times 1}$, $\left\{\dot{\varphi}_p\right\}_{m_p \times 1}$, are the accelerations and velocities of the primary coordinates, and $\left[\dot{J}_o\right]_{m_o \times m_o}$ and $\left[\dot{J}_p\right]_{m_o \times m_p}$ are the total time derivatives of $\left[J_o\right]_{m_o \times m_o}$ and $\left[J_p\right]_{m_o \times m_p}$ respectively. Rearranging equation 25 to express the acceleration of all the coordinates in terms of the primary coordinates acceleration one obtains

$$\left\{\ddot{\varphi}\right\}_{m \times 1}=\begin{Bmatrix}\left[J_o\right]^{-1}_{m_o \times m_o}\left\{E_o\right\}_{m_o \times 1}\\ \left\{0\right\}_{m_p \times 1}\end{Bmatrix}+\begin{bmatrix}-\left[J_o\right]^{-1}_{m_o \times m_o}\left[J_p\right]_{m_p \times m_p}\\ \left[I\right]_{m_p \times m_p}\end{bmatrix}\left\{\ddot{\varphi}_p\right\}_{m_p \times 1} \tag{26}$$

where

$$\left\{E_o\right\}_{m_o \times 1}=\begin{Bmatrix}0\\ -\dfrac{d}{dt}\left\{\dfrac{\partial g}{\partial t}\right\}\\ 0\end{Bmatrix}-\left[\dot{J}_p\right]\left\{\dot{\varphi}_p\right\}-\left[\dot{J}_o\right]\left\{\dot{\varphi}_o\right\} \tag{27}$$

Step 2: Use the initial displacements and velocities of the varying coordinates to express the inertial loads of the mechanism in terms of the primary coordinates accelerations.

The accelerations of the links inertias are expressed by the coordinates' positions, velocities, and accelerations. The inertial loads of each link are expressed by two terms. One term corresponds to the inertial loads that do not depend on the acceleration of the primary coordinates and the other term corresponds to the inertial loads which are linear functions of the primary coordinates accelerations. Using the elastic freedoms numbers as identifiers for the inertial loads, the inertial load vector is defined as

$$\{P_I\}_{N_p \times 1} = \{P_{I_1}\}_{N_p \times 1} + [P_{I_2}]_{N_p \times m_p} \{\ddot{\varphi}_p\}_{m_p \times 1} \tag{28}$$

where $\{P_{I_1}\}_{N_p \times 1}$ corresponds to the inertias' accelerations derived from the first term of the right side of equation 26 and $[P_{I_2}]_{N_p \times m_p}$ corresponds to the inertias' accelerations derived from the coefficient matrix of the second term of the right side of equation 26.

Step 3: Form the global loading vector {P} as a function of the primary coordinate accelerations.

The global loading vector $\{P\}$ is a combination of the inertial loads and the externally applied loads. The external loads are expressed in known functions of time and coordinates positions and velocities. Thus, they are added to the first term of the right side of equation 28 to define the global loading vector as

$$\{P\}_{N_p \times 1} = \{P_{g1}\}_{N_p \times 1} + [P_{I_2}]_{N_p \times m_p} \{\ddot{\varphi}_p\}_{m_p \times 1} \tag{29}$$

where

$$\{P_{g1}\}_{N_p \times 1} = \{P_L\}_{N_p \times 1} + \{P_{I_1}\}_{N_p \times 1} \tag{30}$$

and $\{P_L\}_{N_p \times 1}$ is the external loading vector.

Step 4: Use the global stiffness matrix [K] to solve for the displacement vector {X} as a function of the primary coordinate acceleration.

The global stiffness matrix $[K]$ is defined by the elastic elements properties and the positions of the varying coordinates. Therefore, it is defined at the beginning of each time step. Using equation 24, the global elastic displacement vector $\{X\}$ is computed as

$$\{X\}_{N_p \times 1} = [K]^{-1}_{N_p \times N_p} \{P_{g1}\}_{N_p \times 1} + [K]^{-1}_{N_p \times N_p} [P_{I_2}]_{N_p \times m_p} \{\ddot{\varphi}_p\}_{m_p \times 1} \tag{31}$$

or

$$\{X\}_{N_p \times 1} = \{X_{g1}\}_{N_p \times 1} + [X_{g2}]_{N_p \times m_p} \{\ddot{\varphi}_p\}_{m_p \times 1} \tag{32}$$

where

$$\{X_{g1}\}_{N_p\times 1} = [K]^{-1}_{N_p\times N_p}\{P_{g1}\}_{N_p\times 1} \tag{33}$$

and

$$[X_{g2}]_{N_p\times M_p} = [K]^{-1}_{N_p\times N_p}\{P_{I_2}\}_{N_p\times m_p} \tag{34}$$

Step 5: For the elements where the driving forces and torques are applied, find the elemental load vector {Pe} as a function of primary coordinates accelerations.

For any element, the load vector $\{Pe\}$ is defined by equation 20. The element stiffness matrix $[Ke]$ is already defined and the elastic displacement vector $\{Xe\}$ can be extracted from the global elastic displacement vector $\{X\}$. Thus, the vector $\{Xe\}$ is given as

$$\{Xe\}_{6\times 1} = \{Xe_1\}_{6\times 1} + [Xe_2]_{6\times m_p}\{\ddot{\varphi}_p\}_{m_p\times 1} \tag{35}$$

where $\{Xe_1\}$ and $[Xe_2]$ correspond to $\{Xg_1\}$ and $[Xg_2]$ respectively. Now, using equation 20 the elemental load vector $\{Pe\}$ is computed as

$$\{Pe\}_{6\times 1} = [Ke]_{6\times 6}\{Xe_1\}_{6\times 1} + [Ke]_{6\times 6}[Xe_2]_{6\times m_p}\{\ddot{\varphi}_p\}_{m_p\times 1} \tag{36}$$

Step 6: Equate the elemental load components which are expressed in terms of the primary coordinates accelerations to the corresponding driving forces and torques.

The Known driving forces and torques are applied in the direction of the elemental freedoms. Therefore, they are considered as elemental loads. By writing equation 36 for the elements associated with these driving forces and torques and extracting the equations that correspond to the freedoms where these loads are applied, one obtains

$$\{P_d\}_{m_p\times 1} = \{P_{d_1}\}_{m_p\times 1} + [P_{d_2}]_{m_p\times m_p}\{\ddot{\varphi}_p\}_{m_p\times 1} \tag{37}$$

where $\{P_d\}_{m_p\times 1}$ is the known driving forces and torques vector, $\{P_{d_1}\}_{m_p\times 1}$ is the known elements' loads extracted from the first term in equation 36, and $[P_{d_2}]_{m_p\times m_p}$ is the coefficient matrix extracted from the second term in equation 36. The only unknown in equation 37 is the primary coordinates acceleration vector.

Step 7: Solve for the primary coordinates acceleration vector.

Rearrange equation 37 as

$$\left[P_{d_2}\right]_{m_p\times m_p}\left\{\ddot{\varphi}_p\right\}_{m_p\times 1}=\left\{P_d'\right\}_{m_p\times 1} \tag{38}$$

where

$$\left\{P_d'\right\}_{m_p\times 1}=\left\{P_d\right\}_{m_p\times 1}-\left\{P_{d_1}\right\}_{m_p\times 1} \tag{39}$$

The primary coordinates acceleration vector can be directly solved for using equation 38 by Gauss-Jordan elimination technique.

Step 8: Substitute for the values of the primary coordinates accelerations to find the accelerations of the ordinary coordinates, the global loading vector, the global elastic displacements vector, and the elemental loading vectors.

Equations 26, 29, 32, and 36 express, respectively, the accelerations of varying coordinates, the global loading vector, the global elastic displacements vector, and the elemental loading vectors in terms of the primary coordinates accelerations vector. Substitute for the computed value $\left\{\ddot{\varphi}_p\right\}$ in these equations to completely define the state of the mechanism at the beginning of the time step.

The initial positions and velocities of the varying coordinates at the beginning of the n-th time step are approximated as

$$\{\varphi\}^{(n)}=\{\varphi\}^{(n-1)}+\Delta t\{\dot{\varphi}\}^{(n-1)}+\frac{1}{2}(\Delta t)^2\{\ddot{\varphi}\}^{(n-1)} \tag{40}$$

and

$$\{\dot{\varphi}\}^{(n)}=\{\dot{\varphi}\}^{(n-1)}+\Delta t\{\ddot{\varphi}\}^{(n-1)} \tag{41}$$

These approximations are sufficient if the time step Δt corresponds to small displacements of the mechanism's coordinates.

3.5 Static Equilibrium Position Analysis

In static equilibrium position analysis, the static configuration of a system of rigid bodies, subjected to a set of constraints and acted upon by static loads, is unknown. This problem is solved based on the principle of virtual work and using an iterative procedure. In this type of analysis, like the inverse dynamic analysis, there are two types of coordinates, ordinary and primary coordinates. The solution starts with an initial guess for the positions coordinates.

3.5.1 Derivation of the Static Equilibrium Equations :

The derivation is based on the principle of virtual work. This principle was stated by Bernoulli for a system in static equilibrium [8]. First we consider a single particle whose position is given by $\bar{R}_{p_s}(\bar{q})$ and subjected to a force $\bar{P}_s$, and let $r_{p_s}(\bar{q})$ be the component of the vector $\bar{R}_{p_s}(\bar{q})$ in the direction of the applied load P_s. If the particle is given a virtual displacement δr_{p_s} in the direction of P_s, after noting that $\bar{P}_s = 0$ for static equilibrium, the virtual work is

$$\delta W_{p_s} = P_s \cdot \delta r_{p_s} = 0$$

but

$$\delta r_{p_s} = \sum_{i=1}^{n} \frac{\partial r_{p_s}}{\partial q_i} \delta q_i$$

therefore the virtual work done by P_s is

$$W_{p_s} = P_s \cdot \delta r_{p_s} = P_s \sum_{i=1}^{n} \frac{\partial r_{p_s}}{\partial q_i} \delta q_i$$

for a mechanical system of h points of load application, the total virtual work done is

$$\sum_{s=1}^{h} W_{p_s} = \sum_{s=1}^{h} P_s \sum_{i=1}^{n} \frac{\partial r_{p_s}}{\partial q_i} \delta q_i = 0 \qquad (42)$$

by rearranging and collecting terms, equation (42) can be written as

$$\sum_{i=1}^{n} C_i(\bar{q}) \delta q_i = 0 \qquad i = 1,2,\ldots,n$$

or

$$\{C\}^T \{\delta q\} = 0 \qquad (43)$$

where

$n =$ number of coordinates

$\bar{q} = \langle q_1, q_2, \ldots, q_n \rangle$

These are the varying coordinates that describe the systems equilibrium configuration. These coordinates are related through the systems constraints.

$C_i(\bar{q})$ is the load component associated with the virtual displacement δq_i.

It is desired to express equation (42) in terms of independent virtual displacements as

$$\delta W = \sum_{j=1}^{m} D_j(\bar{q}) \delta q_{d_j} = 0$$

or

$$\{D\}^T \{\delta q_d\} = 0 \tag{44}$$

where

$m =$ number of independent coordinates

q_{d_j} are the independent coordinates.

The number of constraints equations is given by

$k = n - m$

The equilibrium configuration is given by the following set of equations

$$D_j(\bar{q}) = 0 \qquad j = 1,2,\ldots,m$$

and (45)

$$f_l(\bar{q}) = 0 \qquad l = 1,2,\ldots,k$$

where $f_l(\bar{q}) = 0$ are the constraints equations.

So far we have $(m + k = n)$ equations with (n) unknowns, but to calculate $D_j(\bar{q})$ we have to find the load components $C_i(\bar{q})$ and then transform $C_i(\bar{q})$ to $D_j(\bar{q})$. The load components $C_i(\bar{q})$ can be found by looking at equations (42) and (43) and can be given by

$$C_k = \sum_{s=1}^{h} P_s \sum_{j=1}^{n} \frac{\partial r_{p_s}}{\partial q_k} \tag{46}$$

3.5.1.2 Transforming {C} to {D} :

Considering holonomic constraints, the constraints equations can be written as

$$f_l(\bar{q}) = 0 \qquad l = 1,2,\ldots,k$$

The variation of the constraint equation is always zero, then

$$\delta f_l = \sum_{i=1}^{n} \frac{\partial f_l}{\partial q_i} \delta q_i = 0$$

or

$$[J]_{k\times n}\{\delta q\}_{n\times 1} = \{0\}_{k\times 1} \qquad (47)$$

where $[J]_{k\times n}$ is the Jacobian matrix and its elements are given by

$$J_{li} = \frac{\partial f_l}{\partial q_i} \qquad l = 1,2,\ldots,k \text{ and } i = 1,2,\ldots,n$$

Partitioning the $[J]$ matrix in equation (47) as

$$\left[[J_o]_{k\times k} \quad [J_d]_{k\times m} \right] \begin{Bmatrix} \{\delta q_o\}_{k\times 1} \\ \{\delta q_d\}_{m\times 1} \end{Bmatrix} = \{0\} \qquad (48)$$

where $\bar{q}_d$ are the independent coordinates to be used in equation (44), $\bar{q}_o$ are the remaining coordinates.

By manipulating equation (48) , we can have

$$\{\delta q\}_{n\times 1} = [B]_{n\times m}\{\delta q_d\}_{m\times 1} \qquad (49)$$

where

$$[B]_{n\times m} = \begin{bmatrix} -[J_o]^{-1}_{k\times k} & & [J_d]_{k\times m} \\ & [I]_{m\times m} & \end{bmatrix}$$

Equation (49) expresses all the variations in terms of the independent variations.

So, we can write, using equations (43) and (49)

$$\{C\}^T\{\delta\bar{q}\} = \{C\}^T[B]\{\delta\bar{q}_d\} = 0$$

but, from equation (44)

$$\{D\}^T\{\delta q_d\} = 0$$

then, we conclude that

$$\{D\}^T = \{C\}^T[B]$$

or $\quad \{D\} = [B]^T\{C\} \qquad (50)$

and this equation provides the required transformation.

3.5.2 The numerical solution procedure :

Step 1 : Assume initial guess values for all the coordinates

$\{q_i\}$ $i = 1,2,\ldots,n$

Step 2 : Find the exact values of the dependent coordinates $\{q_o\}$ corresponding to the assumed values of the independent coordinates $\{q_d^o\}$

Step 3 : Find the $[B]$ and $\{C\}$ matrices, and calculate the resulting $\{D^o\}$vector from equation (50).

Step 4 : Numerically differentiate D_i with respect to q_j using 6-th order differentiation formula

$$\frac{\partial D_i}{\partial q_j} \approx \frac{1}{60\Delta q_j}\left(D_{i_3} - 9D_{i_2} + 45D_{i_1} - 45D_{i_{-1}} + 9D_{i_{-2}} - D_{i_{-3}}\right)$$

where $D_{i_3} = D_i\left(q_j + 3\Delta q_j\right)$,

$D_{i_2} = D_i\left(q_j + 2\Delta q_j\right)$,

$D_{i_1} = D_i\left(q_j + \Delta q_j\right)$,

$D_{i_{-1}} = D_i\left(q_j - \Delta q_j\right)$,

$D_{i_{-2}} = D_i\left(q_j - 2\Delta q_j\right)$,

and $D_{i_{-3}} = D_i\left(q_j - 3\Delta q_j\right)$

then find

$$[J_d] = \left[\frac{\partial D_i}{\partial q_j}\right]$$

Step 5 : Then solve the expression

$$[J_d]\{\Delta q_d\} = -\{D^o\}$$

for $\{\Delta q_d\}$, (these values when added to $\{q_d\}$ should force the $\{D\}$ vector to converge to $\{0\}$.

Step 6 : Increment the vector $\{q_d\}$ by the values $\{\Delta q_d\}$ obtained in step 5. i.e. let $\{q_d\} = \{q_d^o\} + \{\Delta q_d\}$ and repeat steps 2,3,4,5 and 6 until convergence is obtained.

CHAPTER IV

INPUT DATA PREPARATION AND PROGRAM STRUCTURE

This program is capable of performing dynamic and static analysis of general planar mechanisms. For the program to perform the analysis, the mechanism must be modeled and transferred into numerical data and mathematical expressions, which the program can interpret and analyze. This has to be done by the user of the program. After the input data is ready, the user can execute the program and input the data directly.

4.1 The Input Data

To prepare the input data, the user must develop a complete model for the mechanism. This should be done by following the necessary steps discussed in Chapter II. As an example consider the six-bar mechanism shown in Figure (8). The input data can be defined directly from the model. It consists of the following items:

4.1.1 The Characteristic Data

The characteristic data provides general information about the mechanism. It consists of the following items which are defined by their name as it appears in the computer program:

1) nel = number of elements

2) nl = number of independent closed loops

3) nco = number of varying coordinates (M)

4) nj = number of joints

5) dt = time increment

6) tt = total time of the analysis

7) kd = indicate which analysis is needed, kd=0 for kinematic analysis, kd=1 for dynamic analysis, kd=2 for static analysis, kd=3 for static equilibrium position analysis.

8) ntf = number of elastic freedoms (N_p)

9) gr = local acceleration of gravity

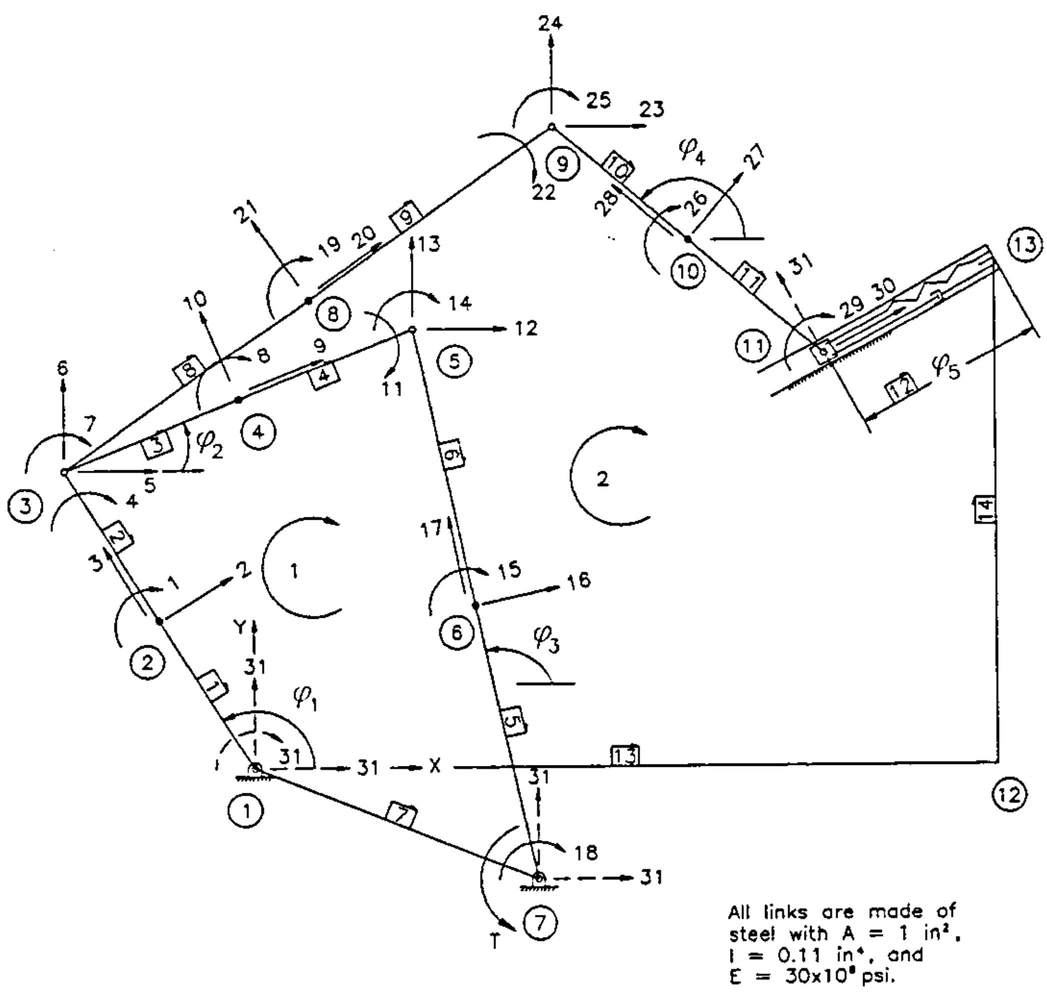

Figure (8) A Six-bar Mechanism to Illustrate the Preparation of The Input Data [1].

10) el = modulus of elasticity

11) nvm = number of variable messes

12) npc = number of primary coordinates

These items can be entered through the data editor in the DAMES program. For the mechanism shown on Figure (8), these items are

14, 2, 5,13, .05, 2., 1, 30, 386, 30000000, 0, 0

where the time increment is .05 second and the total time of the analysis is 2.0 seconds.

4.1.2 The Element Data

The element data describe each element in the mechanism. The units in which these information are provided must be consistent. The length and the orientations of the element and its freedom coordinate systems are defined by two parameters. The first parameter is the constant component and the second parameter is the number of the varying coordinate representing the varying component. The following items give the name of the parameters in the program and its definition.

1) ne = the element number

2) cl(ne) = constant component of length

3) ivl(ne) = variable component coordinate number of length

4) alc(ne) = constant component of element orientation

5) ialv(ne) = variable component coordinate number of orientation

6) nij(ne) = the number of the initial joint

7) ntj(ne) = the number of the terminal joint

8) np(ne,j) (j=1,..,6) = the global freedom numbers in the order $(N_1, N_2, \ldots, N_6)$ as shown on Figure (7).

9) gic(ne) = the constant component of the orientation of the initial joint freedoms coordinate system

10) igiv(ne) = variable component coordinate number

11) gtc(ne) = the constant component of the orientation of the terminal joint freedoms coordinate system

12) igtv(ne) = variable component coordinate number

13) ar(ne) = the cross-sectional area

14) ami(ne) = the cross-sectional area moment of inertia

The element data for element 1 in Figure (8) is 1,1.,0,0.,1,1,2,31,31,1,2,3,0.,0,0.,0,1.,0.1

For element 11

11,3.,0,0.,1,1,2,31,31,31,1,2,3,0.,0,0.,0,1.,0.1

and for element 13

13,0.,5,30 ,0,13,11,0,0,0,0,0,0,0.,0,0.,0,0.,0.

4.1.3 Closed Loop Data

The closed loop data is an important part in describing the topology of the mechanism. It consists of the following items for each loop.

1) ln = loop number

2) nle(ln) = number of elements in the loop

3) lpe(ln,j) (j=1,..,nle(ln)) = elements numbers in the loop when it is traced in the clockwise direction.

For the mechanism shown on Figure (8) the input data for loop 1 is

1,7,1,2,3,4,-6,-5,-7

and for loop 2 is

2,9,1,2,8,9,10,11,-13,-14,-12

4.1.4 Joint Data

The joint data describe the path connecting a joint to the origin of the base frame and the inertias lumped at the joint. It is arranged in the following order for each joint.

1) jn = joint number

2) npe(jn) = number elements in the path

3) mpe(jn,j) (j=1,..,npe(jn)) = elements numbers included in the path

4) xma(jn) = the mass lumped at the joint

5) xmoi(jn) = the mass moment of inertia

Consider joint 1 in the example mechanism of Figure (8), its data is given by

1,0,0.,0.

where there are no elements in the path of joint 1.

The data for joint 10 is

10,5,1,2,8,9,10,.0003,.00015

For joint 6 the joint data is

6,2,7,5,.0002,.0001

4.1.5 Variable Mass Data

The variable mass data is only considered if conditions like those discussed in step 8 of the modeling technique are present. The variable mass data includes the following items.

1) mnv(i) = the joint number at which the variable mass is lumped.

2) im1(i) = the element number at one side of the variable mass.

3) im2(i) = the element number at the other side of the variable mass.

4) den(i) = the linear density of the link material.

To illustrate the input data for variable masses, consider the mechanism shown on Figure (3). There are two variable masses, one at joint 3 and the other at joint 5. The data for the first one is

3,2,3,r

where r is the linear density of the moving slider path.

4.1.6 The primary Coordinates Data

This set of data is considered when inverse dynamic analysis is performed. It consists of the following items for each primary coordinate.

1) k = the number of the primary coordinate being considered.

2) ti(k) = its initial position.

3) tiv(k) = its initial velocity.

4) npf(i) = the global freedom number associated with the primary coordinate.

5) nedc(i) = number of elements associated with the primary coordinate.

6) nedec(i,j) (j=1,..,nedc(i)) = the elements' numbers in item #5.

To clarify each item in the primary coordinates data, consider the mechanism shown in Figure (9). The known torque T drives the mechanism. The primary coordinate is numbered last and it is φ_3 . The global freedom numbers associated with it

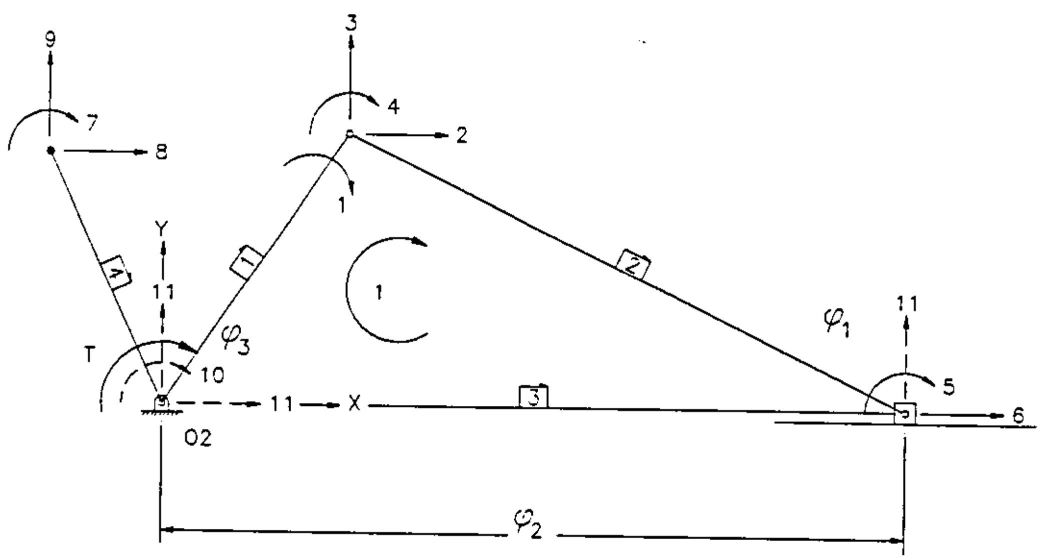

Figure (9) A Slider-Crank Mechanism Driven With Known Torque at O_2 [1].

is 10 and the torque is applied in its positive direction. There are two elements associated with freedom 10. They are elements 1 and 4. Thus, the data for this primary coordinate is

$$3, \varphi_{3_o}, \dot{\varphi}_{3_o}, 10, 2, 1, 4$$

where φ_{3_o} and $\dot{\varphi}_{3_o}$ are the initial position and velocity of the primary coordinate φ_3 .

4.1.7 Initial Condition-Estimate-Data

To start the iteration in the displacement analysis, initial estimates of the coordinates must be provided. These estimates do not have to be accurate, but close to the correct value. They may be obtained by a sketch of the mechanism. This data set consists of the estimates of the initial positions of the mechanism coordinates excluding the primary coordinates. For the mechanism shown in Figure (8), there are 5 coordinates (no primary coordinates) and their data set is

$$\varphi_{1_o}, \varphi_{2_o}, \varphi_{3_o}, \varphi_{4_o}, \varphi_{5_o}$$

where φ_{1_o} is the initial estimate of φ_1 .

4.2 Data Input for the Constraints, Loads and Driving Forces and Torques

There are three parts of mathematical expressions the user must supply. The first includes the user constraints, which are in equation form describing the time-dependent motion generators and spatial and length constraints. The second defines the externally applied loads on the mechanism. The Last one defines the known forces and torques which drive the mechanism. The user can input the expressions into the program directly, since the program has a text editor. The program can interpret the mathematical expressions directly without any need for re-compilation, since an equation parser has been developed and implemented into the program. Of course during the development of this parser, a certain syntax and general rules have been defined. The general format for the user constraints, the user loads and the driving forces sections is shown below. (Note: the input data is shown in italic characters)

#uscons

input the expressions for the user constraints here *return*

#end

#usload

input the expressions for the user loads here

return

#end

#drive

input the expressions for the driving forces and torques here

return

#end

the variable names used in the mathematical expressions are described in the following sections. But for now, here are the rules for the data input

1- If the user want the program to ignore a line, he can use a semicolon to define the line as a comment.

Example:

#uscons

; this is a comment

2- The user cannot use variable names other than those listed in the following sections.

3- If you want to assign a value to a variable, type the variable name on a new line, follow it by an equal sign '=', then input the value or expression.

Example:

#uscons

ntc=1

nsc=2

...

4- The user must not input more than one statement in a single line.

Example:

#uscons

nsc=1 ntc=2 X this is wrong

...

5- To access an item in an array-type variable name, use the variable name followed by the array index enclosed in round brackets, and if the array is two dimensional, use the variable name followed by the two index numbers separated by a comma and enclosed in round brackets.

Example:

#uscons

ntc=1

nsc=2

*tc(1)=110.-5*tm-ti(1)*

ptc(1,2)=-1.0

...

6- The user can use a goto statement. It must be followed by an integer specifying a label, which program execution can move to. The label must exist and should be followed by a colon.

Example:

goto 20

...

20:

...

7- The user can use the if(...) conditional statement in the following manner (the else statement is optional)

if(condition)

expression1

expression2

...

else

expression1

expression2

...

endif

the following conditional operators can be used

<table>
<tr><th>Operator</th><th>Meaning</th></tr>
<tr><td>=</td><td>equal</td></tr>
<tr><td>></td><td>greater than</td></tr>
<tr><td><</td><td>less than</td></tr>
<tr><td>>=</td><td>greater than or equal</td></tr>
<tr><td><=</td><td>less than or equal</td></tr>
<tr><td><></td><td>not equal</td></tr>
<tr><td>&</td><td>and</td></tr>
<tr><td>|</td><td>or</td></tr>
</table>

Example:

if(nst=2)

goto 10

endif

...

10:

8- The following operators can be used in the expressions, they are ordered according to their priority in execution from the highest to the lowest.

Operator	Meaning
( )	Open & Close brackets
* /	Multiply and divide
+ -	Plus and Minus

9- The following mathematical functions can be used (Note: Angles used in trigonometric functions are in radians).

Function	Meaning
cos(x)	cosine of x
sin(x)	sine of x
tan(x)	sin(x)/cos(x)
acos(x)	the inverse cosine of x
asin(x)	the inverse sine of x
atan(x)	the inverse of tan x
power(x,y)	returns e (2.718281828) to the power x
log(x)	returns the natural logarithm of x to the base e
sqr(x)	returns x to the power 2 (x^2)
sqrt(x)	returns the square root of x ($\sqrt{x}$)

4.2.1 The User Constraints

The functions that relate the coordinates with time and with each other and the derivative of these functions are supplied in *"#uscons"* section. The user only supplies the equations in the format described previously. For time-dependent motion generators, the variable names used and their corresponding mathematical expressions are

$$tc(i) = g_i$$
$$ptc(i) = \frac{\partial g_i}{\partial \varphi_i}$$
$$pttc(i) = \frac{\partial g_i}{\partial t}$$
$$dptc(i,j) = \frac{d}{dt}\frac{\partial g_i}{\partial \varphi_i}$$
$$dpttc(i) = \frac{d}{dt}\frac{\partial g_i}{\partial t}$$

The variable names for the coordinates positions, velocities, accelerations and time are as follows

$$\varphi_i = ti(i)$$
$$\dot{\varphi}_i = tiv(i)$$
$$\ddot{\varphi}_i = tia(i)$$

time = tm

To illustrate the use of these names, suppose the input crank of the mechanism shown in Figure (8) is driven with the following function

$$\varphi_1(t) = \varphi_{1_o} + \omega \times tm + \frac{1}{2}\alpha \times tm^2$$

The residual equation is

$$g_1(\varphi_1) = \varphi_{1_o} + \omega \times tm + \frac{1}{2} \times \alpha \times tm^2 - \varphi_1$$

Then

$$tc(1) = g_1 = \varphi_{1_o} + \omega \times tm + 0.5 \times \alpha \times tm^2 - ti(1)$$

where φ_{1_o} , ω, and α are known constraints. The partial derivatives of g_1 are as follows:

$$ptc(1,1) = \frac{\partial g_1}{\partial \varphi_1} = -1$$

$$pttc(1) = \frac{\partial g_1}{\partial t} = \omega + \alpha \times tm$$

and the total time derivatives are

$$dptc(1,1) = 0$$

$$dpttc(1) = \alpha$$

For the spatial and length constraints, the variable names are

$$sc(i) = S_i$$

$$psc(i,j) = \frac{\partial S_i}{\partial \varphi_j}$$

$$dpsc(i,j) = \frac{d}{dt}\frac{\partial S_i}{\partial \varphi_j}$$

Consider the 2R robot shown in Figure (6), the residual form of the constraints equation is

$$Sc(1) = S_1(\varphi_3, \varphi_4) = \frac{(ti(3) - x_o)^2}{a^2} + \frac{(ti(4) - y_o)^2}{b^2} - 1$$

where x_o , y_o ,a , and b are known constraints. The partial derivatives are:

$$psc(1,3) = \frac{\partial S_1}{\partial \varphi_3} = 2.0 \times b^2 \times (ti(3) - x_o)$$

and

$$psc(1,4) = \frac{\partial S_1}{\partial \varphi_4} = 2.0 \times a^2 \times (ti(4) - y_o)$$

The total time derivatives are:

$$dpsc(1,3) = 2.0 \times b^2 \times tiv(3)$$

and

$$dpsc(1,4) = 2.0 \times a^2 \times tiv(4)$$

The user also must specify the number of existing constraints equations for each type. For the time-dependent constraints the variable name for the number of equations is ntc. Similarly the variable name for the spatial constraints equations is nsc. These numbers are

$$ntc = 1$$

and

$$nsc = 1$$

for the two examples discussed earlier. The following is a copy of *"#uscons"* section and the location of the user supplied expressions.

#uscons

ntc=1 ← Number of time dependent constraints

nsc=1 ← Number of spatial constraints

if(nst=2)

goto 10

endif

*tc(1)=6.2832*sin(.3927*tm)+1.5708-ti(3)*

ptc(1,3)=-1 $\leftarrow \frac{\partial g_L}{\partial \varphi_3}$

*pttc(1)=2.4674*cos(.3927*tm)* $\leftarrow \frac{\partial g_L}{\partial t}$

sc(1)=-ti(4) $\leftarrow S_r(\varphi_4)$

psc(1,4)=-1. $\leftarrow \frac{\partial S_r}{\partial \varphi_4}$

goto 20

10:

*dpttc(1)=-.9689*sin(.3927*tm)* $\leftarrow \frac{d}{dt}\left(\frac{\partial g_L}{\partial t}\right)$

20:

return

#end

4.2.2 The User Loads

In dynamic analysis, the forces and moments are expressed as a function of time or coordinates positions and velocities. To permit the use of general expressions for those externally applied loads, the user can specify the loads by mathematical expressions. The external forces and moments can be applied in the direction of any active elastic freedom. They are identified by the numbers of these elastic freedoms. The section containing the expressions of the externally applied loads is called *"#usload"* . The variable name for the applied load is

$$usl(i)$$

where i is the freedom number in which direction the load is applied. Suppose that there is a torque T applied at O_4 in the mechanism shown on Figure (8). Let T act in a counter clockwise direction and be given by

$$T = 10.0 + 5 \times \left(\varphi_3 - \varphi_{3_o}\right)$$

To introduce this torque in the program, the following statement must be supplied in the *"usload"* section:

$$usl(13) = -10.0 - 5 \times \left(ti(3) - \varphi_{3_o}\right)$$

where φ_{3_o} is the known initial value of φ_3 . The signs are reversed because the torque is applied in the opposite direction of the rotational freedom 18. Also, in the mechanism shown in Figure (8), consider the spring and damper forces at the slider. To introduce these forces, the following statements must appear in the *"usload"* section

$$usl(30) = k \times \left(ti(5) - \varphi_{5_o}\right) + c \times tiv(5)$$

where k and c are known constants and φ_{5_o} is the initial value of φ_5 when the spring is not stretched or compressed. The *"usload"* section and the location of the user supplied expressions can be listed as:

```
#usload
usl(13)=-10.-5.*(ti(3)-2.)
usl(30)=3.*(ti(5)-1.5)+0.5*tiv(5)
return
#end
```

4.2.3 The Known Driving Forces and Torques

In the inverse dynamic analysis, the mechanism is driven by known forces and torques. The section *"#drive"* identifies these driving forces and torques. These can be functions of time or a combination of time and coordinates' positions and velocities. The variable names for these functions is $fpc(i)$. The index i $(i = 1,2,...,m_p)$ corresponds to the order by which the input data for a given driving load is supplied. Suppose that the mechanism shown in Figure (9) is driven by the known torque T, and T is defined by

$$T = 100.0 + 30 \times (\varphi_3 - \varphi_{3_o})$$

This driving torque is introduced in the program by the following statement which must be supplied in this section *"#drive"*

$$fpc(1) = 100.0 - 30 \times (ti(3) - \varphi_{3_o})$$

where φ_{3_o} is a known constant. The section *"#drive"* is shown in the following listing where the location of the user supplied statements is indicated

```
#drive
fpc(1)=100.-30*(ti(3)-2.0)
return
#end
```

4.3 Program Structure

The DAMES program for the Dynamic Analysis of MEchanical Systems is based on the theory discussed in Chapter III. The input to the program was discussed and illustrated in Section 4.2. The flow of the program is shown in Figures (10.A), (10.B) and (10.C). The program is divided into subroutines, and each subroutine is explained in the flow-chart. The program is written in "C" language, and is developed under the most popular Graphical User Interface environment on a PC computer "Microsoft Windows". The program is too large to be listed in this thesis, it takes more than 200 pages, so only part of it is listed and supplied in appendix C. The listing shows the part of the program that relates to the analysis, file handling, and the main functions. The output of the program is stored in 6 data files. The contents of each are given in the following:

a. Files 1,2, and 3 contain the positions, velocities, and accelerations of all the coordinates respectively.

b. File 4 contains the x and y components of the positions, velocities, and accelerations of all the joints.

c. File 5 contains the elastic displacement vector {Xg}.

d. File 6 contains the element forces vector {Pe} for all the elastic elements of the mechanism. If static analysis is performed, the results are stored in this file in table form.

The file names for the above are given as

FNAME.BK#

where

FNAME stands for the project name, as the user specifies it when saving the data,

stands for the file number as listed above.

For example if the user saved the input data to a project name *ROBOT*, then the file names for the output will be in consecutive order

ROBOT.BK1, *ROBOT.BK2*, *ROBOT.BK3*, *ROBOT.BK4*, *ROBOT.BK5* AND *ROBOT.BK6* . The information in these files are listed at each time step along with the corresponding time, and in case of static equilibrium position analysis, the time is replaced by the iteration number and only the data for the coordinates and joint positions are saved.

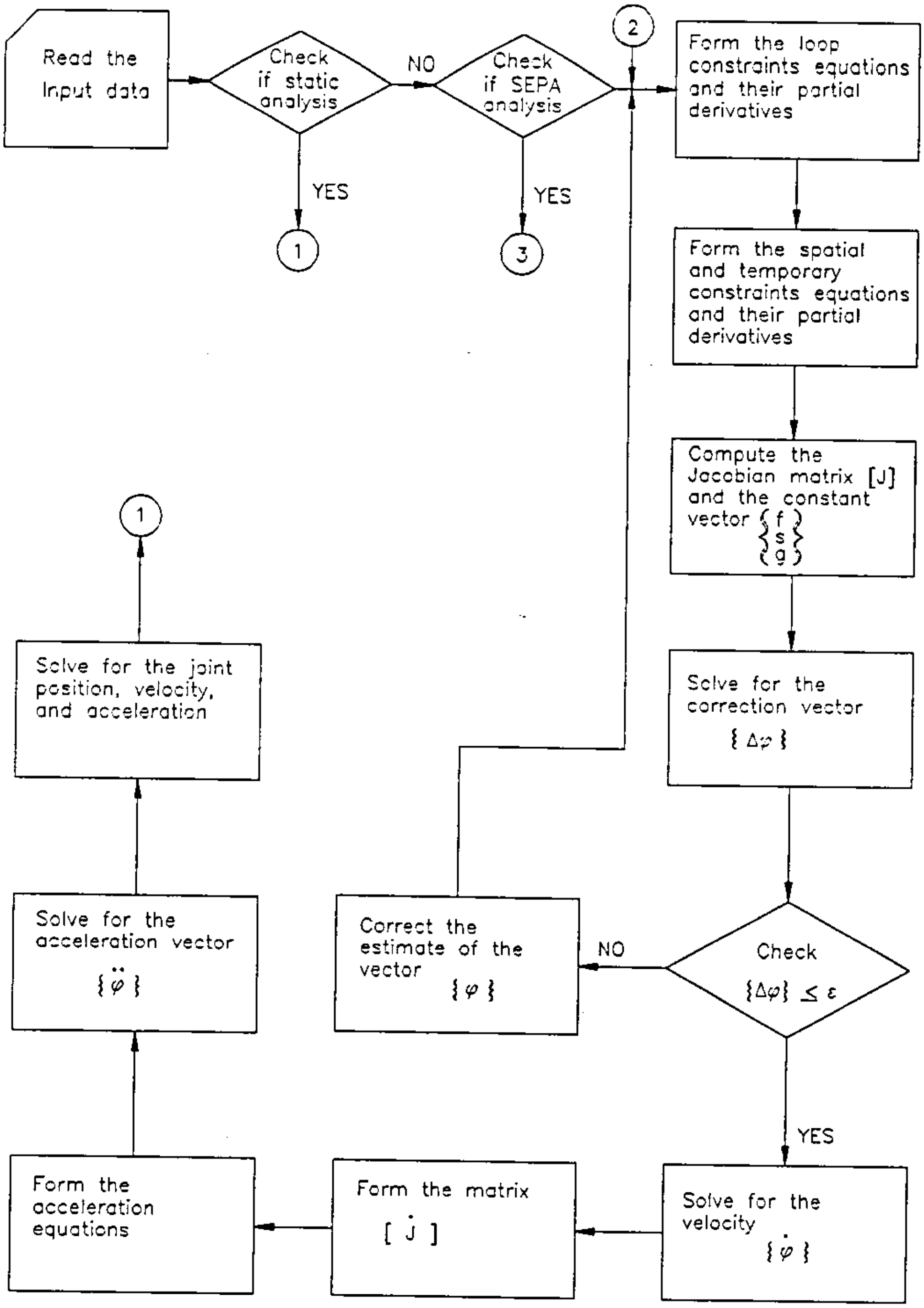

Figure (10.A) The Flow Chart of the Computer Program.

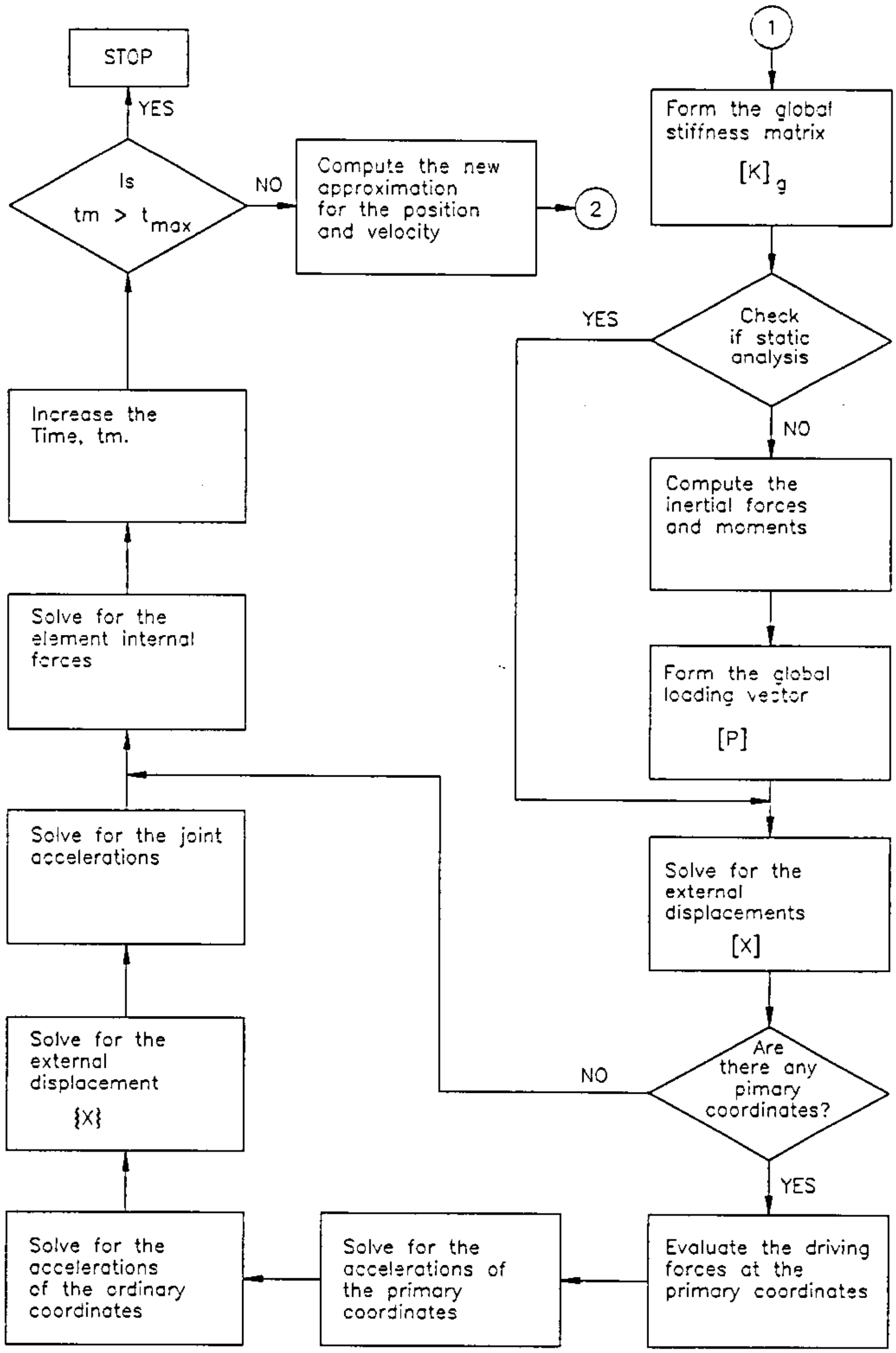

Figure (10.B) The Flow Chart of the Computer Program (Continued).

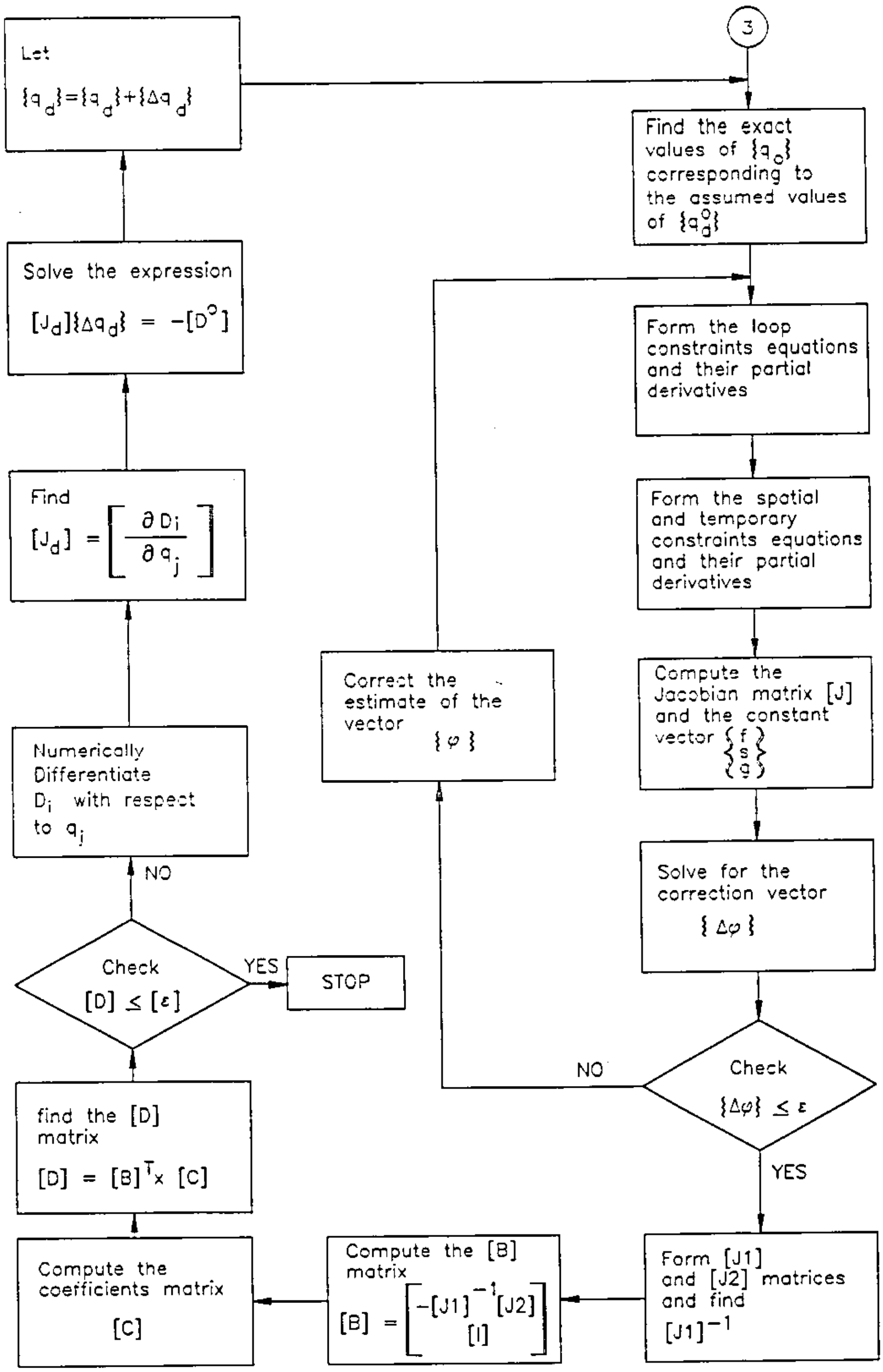

Figure (10.C) The Flow Chart of the Computer Program (Continued).

The main window in the program has a tool bar which contains seven buttons as shown in Figure (11.A), the first is to open a previously saved project, the second is to save the current data, the third is to activate the numerical data input editor as shown in Figure (11.B), the fourth button is to activate the input data editor for user supplied constraints, loads, and driving forces and torques, the fifth button is to begin the analysis, the sixth button is to activate the presentation of the results where new buttons appear on the right side of the screen as shown in Figure (11.C), the user can have a plot of coordinates' positions vs. time, coordinates' velocities vs. time, coordinates' accelerations vs. time, elements internal forces, external joints deflections, joints' x-positions and y-positions vs. time, and the path of each joint. The user can have a printout of any chart or table displayed, by pressing the "PRINT" button. The seventh button on the tool bar activates the animated simulation of the mechanism movements as shown in Figure (11.D).

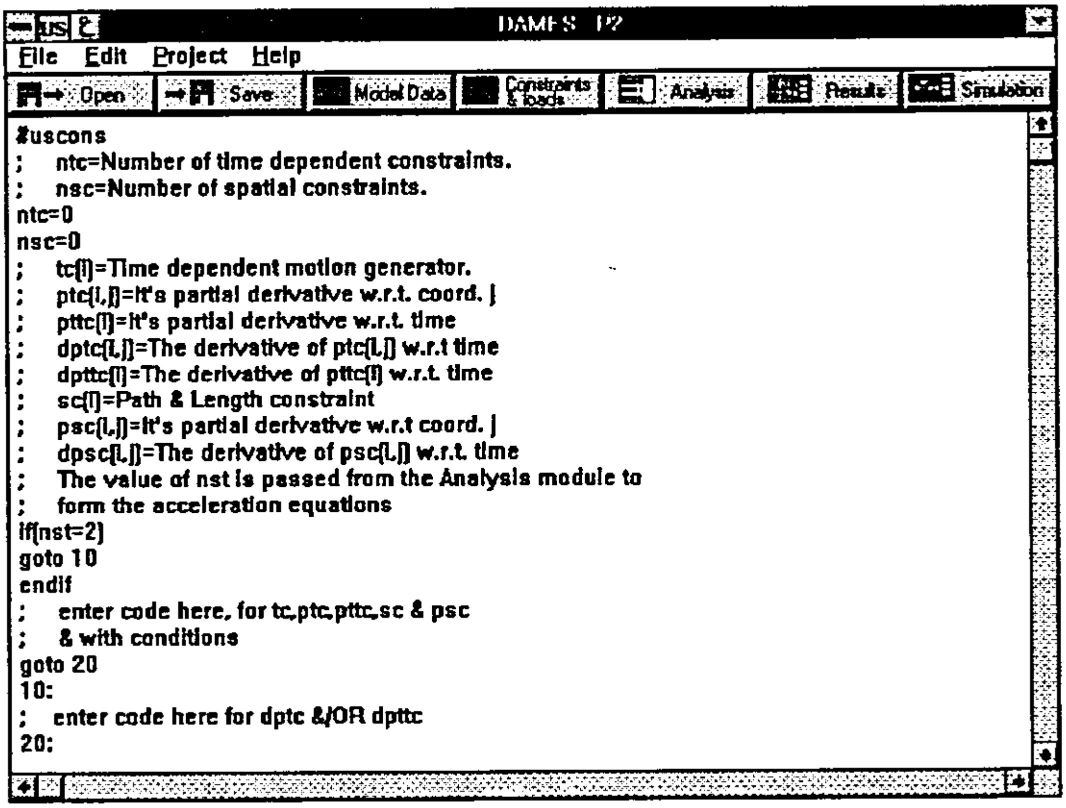

Figure (11.A) The Main Window of the DAMES Program.

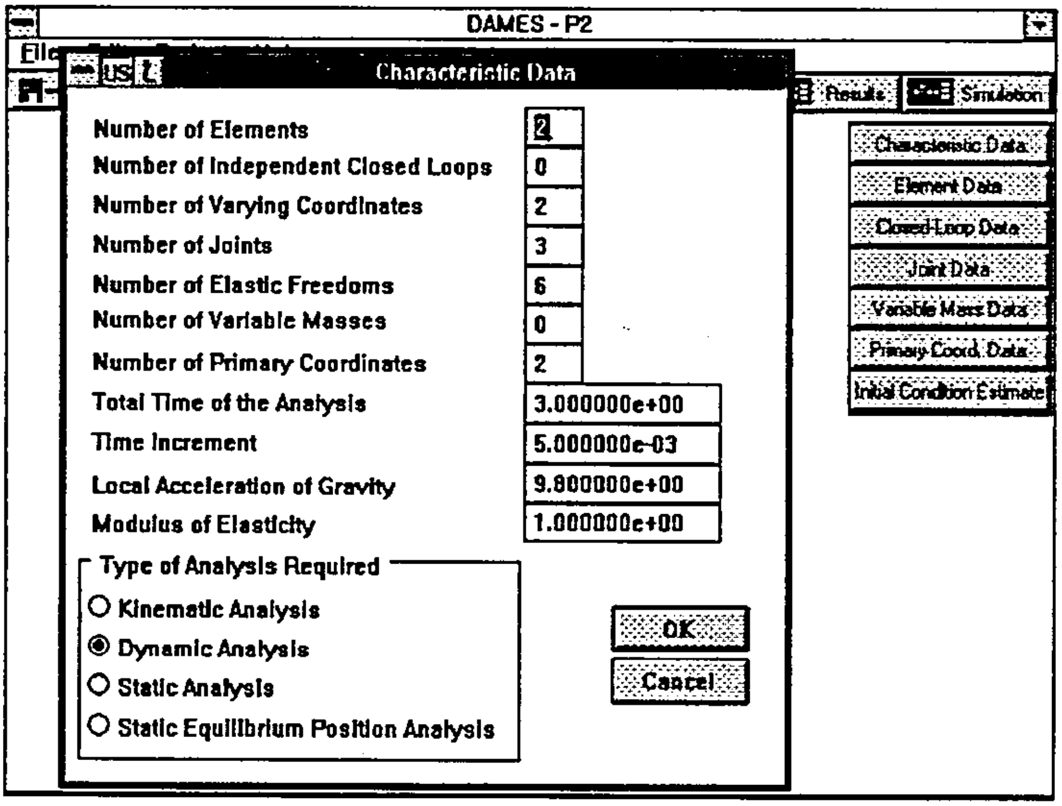

Figure (11.B) The Numerical Data Input Editor of the DAMES Program.

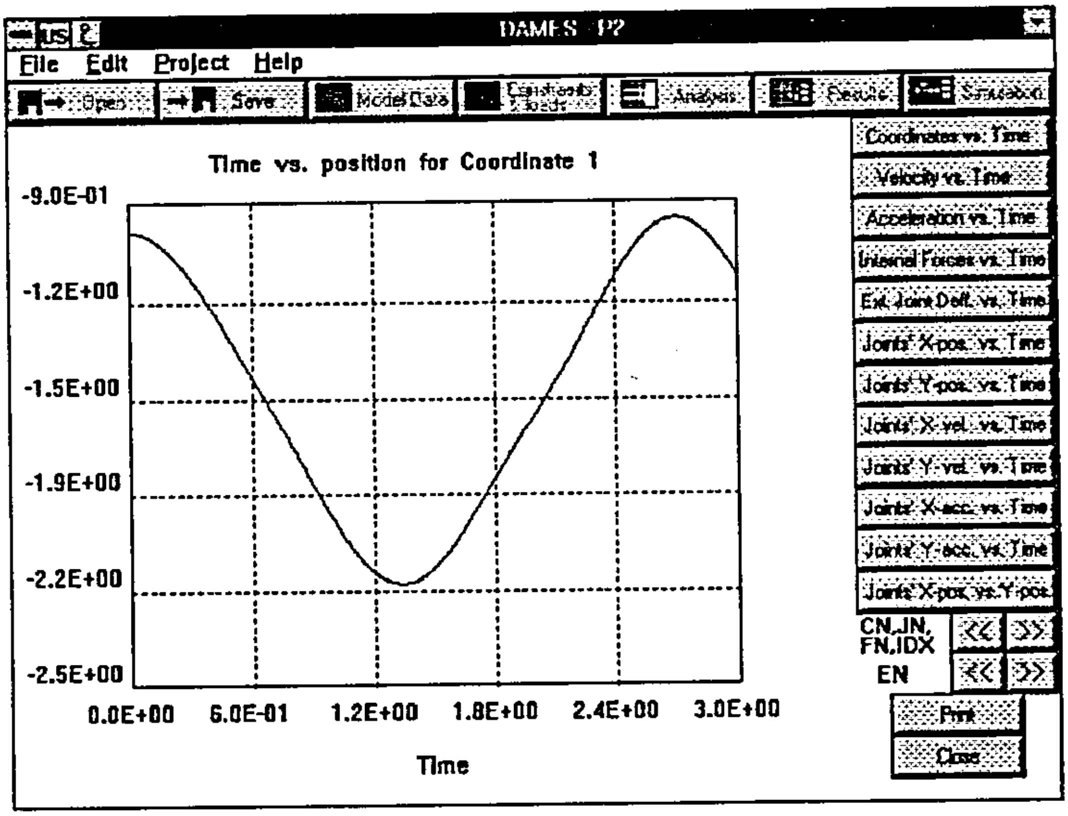

Figure (11.C) The Results Presentation Window of the DAMES Program.

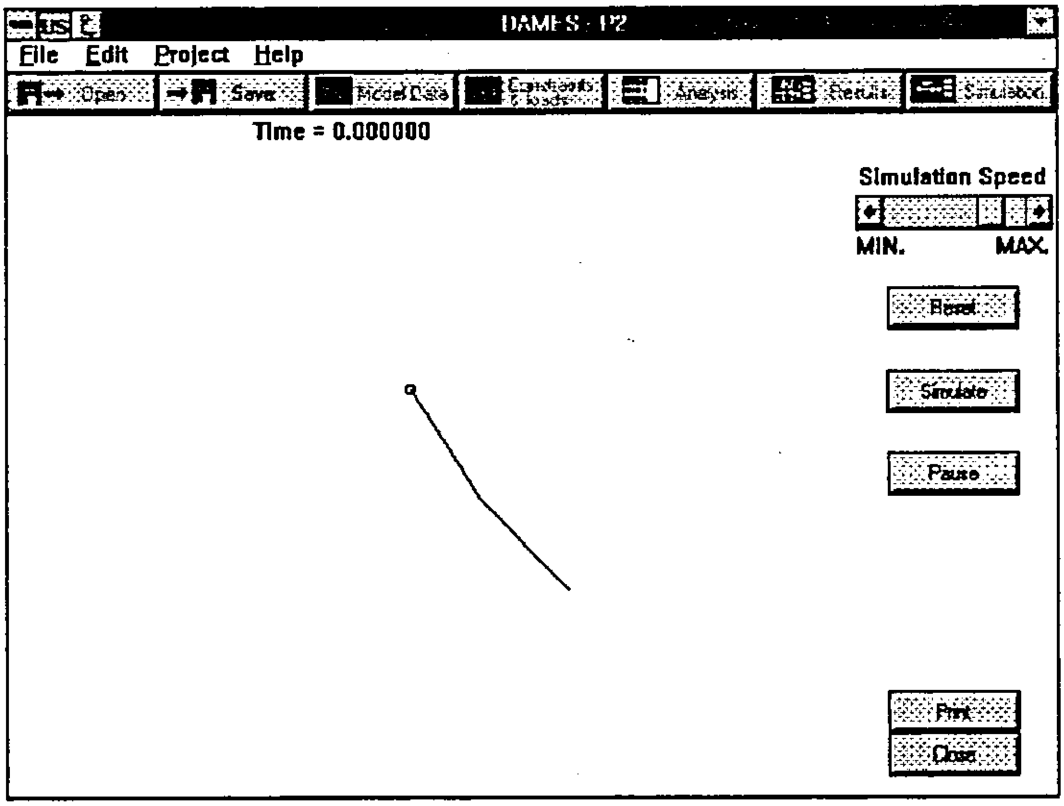

Figure (11.D) The Simulation Window of the DAMES Program.

CHAPTER V

ILLUSTRATIVE EXAMPLES

In this chapter several examples are solved. Those examples cover various areas in the analysis of mechanical systems. Some examples are solved for the purpose of verifying that the results obtained by this program are correct, while other examples are solved for the purpose of illustration. The first three examples are forward dynamic analysis of different mechanisms. The first example is a simple pendulum. This example is solved in three ways, once with free oscillation, second with half critical damping and third with critical damping. The results of this example are verified by comparing them with the analytical solution. The second example is a double pendulum. In the third example the dynamic response of an automobile moving on a sinusoidal road is determined. The results of this example are compared with the results obtained by the DYMAC program [4]. The fourth and fifth examples are inverse dynamic analysis, where a disk cam with radial flat-faced follower and a 3-R robot with mixed-loop configuration are considered. The fourth example is compared with the analytical solution. The 6th, 7th, and 8th examples are on static equilibrium position analysis. The 6th example is solved manually for the purpose of comparison. The 7th example is a 3-R robot. And the 8th example is to determine the static configuration of the lumbar region of the human spine, subjected to certain loading conditions. The following is a detailed discussion for each example and its results.

Example 1

A Simple Pendulum

In this example the oscillation of a simple pendulum is analyzed using the DAMES program and compared to the exact solution for the purpose of checking the program. This example is solved in three different ways, once with free oscillations, second with half critical damping and third with critical damping. The finite element model for the pendulum is shown in Figure (12). The data for the pendulum are as follows:

Mass lumped at joint (2) = 1 Kg

Length of the hanging cord = 1 meter

The initial angle the cord forms with the vertical = 10^0

There is one primary coordinate associated with the pendulum. The coordinate is φ_1 . The primary freedom related to the primary coordinate is freedom 4. The load generated by this primary coordinate is

$$fpc(\ ) = C \times \dot{\varphi} = C \times \quad (\ )$$

where C is the damping coefficient.

The input data for this example is shown in Appendix A.

This example is solved in three ways:

1. The pendulum with free oscillations:

The exact solution for the period of one oscillation is

$$\tau = \frac{2k}{\pi}\left(2\pi\sqrt{\frac{l}{g}}\right) = \frac{2k}{\pi}\left(2\pi\sqrt{\frac{1}{9.81}}\right) = 2.010158 \text{ seconds}$$

where k = 1.574 for a 10^0 angle [9].

and the result obtained by the DAMES program is 2.01 seconds as can be seen in Figure (13) for a complete cycle.

Taking into account that the time increment for the analysis was 0.005 second, the obtained result is a good approximate of the exact solution.

(2) The pendulum with half critical damped oscillations:

For a simple pendulum, the critical damping coefficient is

$$C_c = 2\sqrt{\frac{g}{l}} = 6.264184$$

and the circular frequency with damping is

$$q = \frac{2\pi}{\tau}\sqrt{1-\left(\frac{C}{C_c}\right)^2}$$

where τ is the period of small oscillation without damping

and for $\dfrac{C}{C_c} = 0.5$

$q = 2.70695$ rad/sec, and the period of oscillation is

$\tau = \dfrac{2\pi}{q} = 2.32113$ seconds

and the result obtained by the DAMES program is 2.32 seconds as can be seen in Figure (14) for a complete cycle, which remains within a good approximation of the solution.

<u>(3) The pendulum with critically damped oscillations:</u>

For a simple pendulum, the critical damping coefficient is

$$C_c = 2\sqrt{\frac{g}{l}} = 6.264184$$

and the circular frequency with damping is

$$q = \frac{2\pi}{\tau}\sqrt{1-\left(\frac{C}{C_c}\right)^2}$$

where τ is the period of small oscillation without damping, and for $\dfrac{C}{C_c} = 1.0$

$q = 0.0$ rad/sec, and the period of oscillation is

$\tau = \dfrac{2\pi}{q} \rightarrow$ infinity, or there will be no oscillations.

and looking at Figure (15) obtained by DAMES program, we can see that there are no oscillations, and the result is compatible with the solution.

A simulation of the pendulum can be seen using the DAMES program.

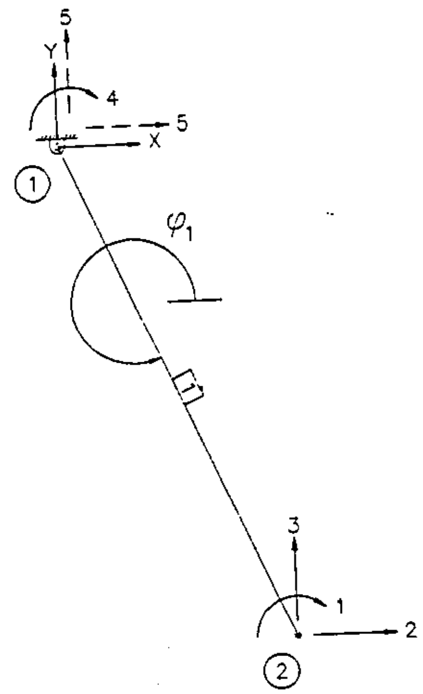

Figure (12) A Simple Pendulum.

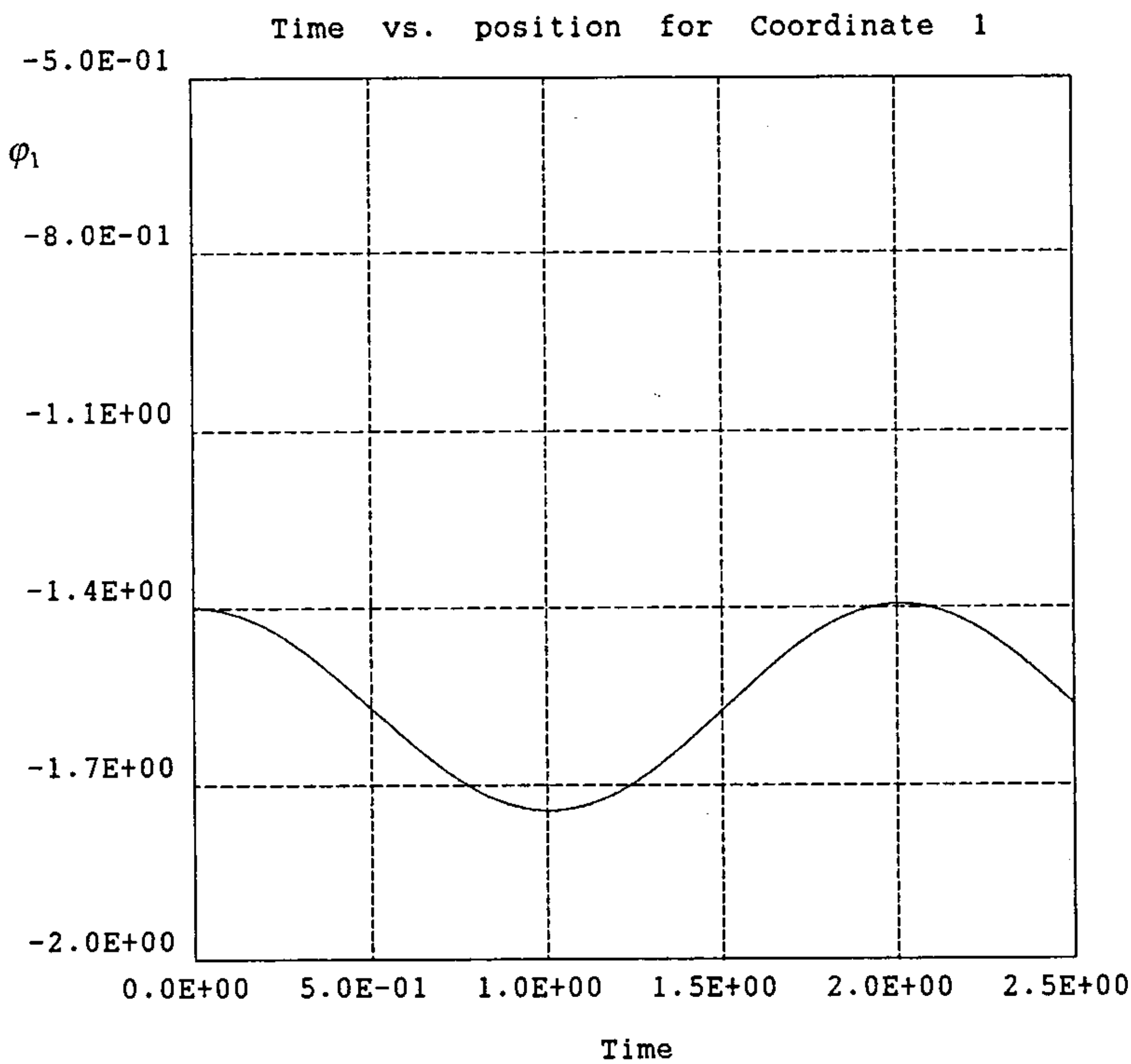

Figure (13) The Time History of the Coordinate φ_1 without Damping.

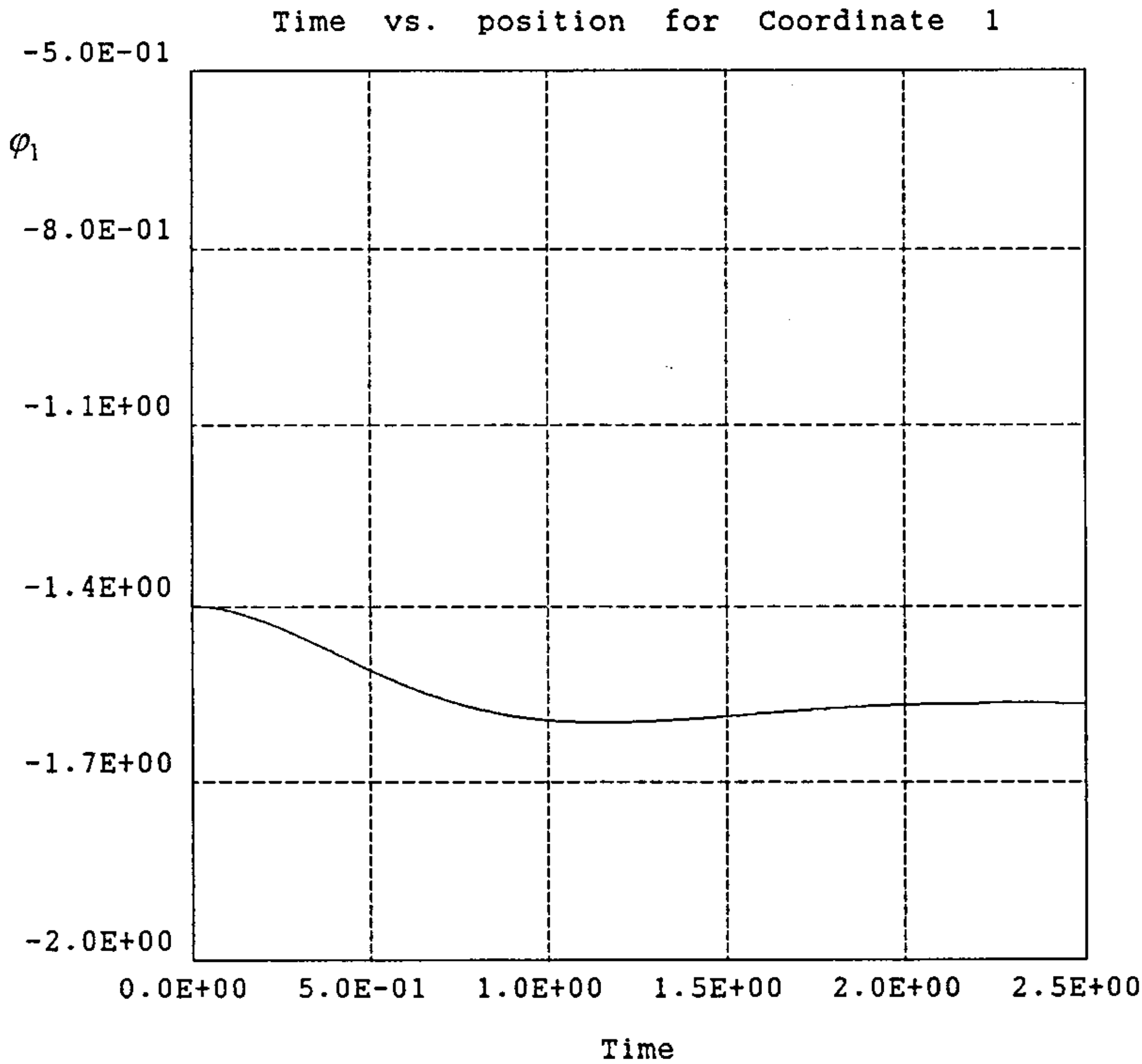

Figure (14) The Time History of the Coordinate φ_1 with Half-Critical Damping.

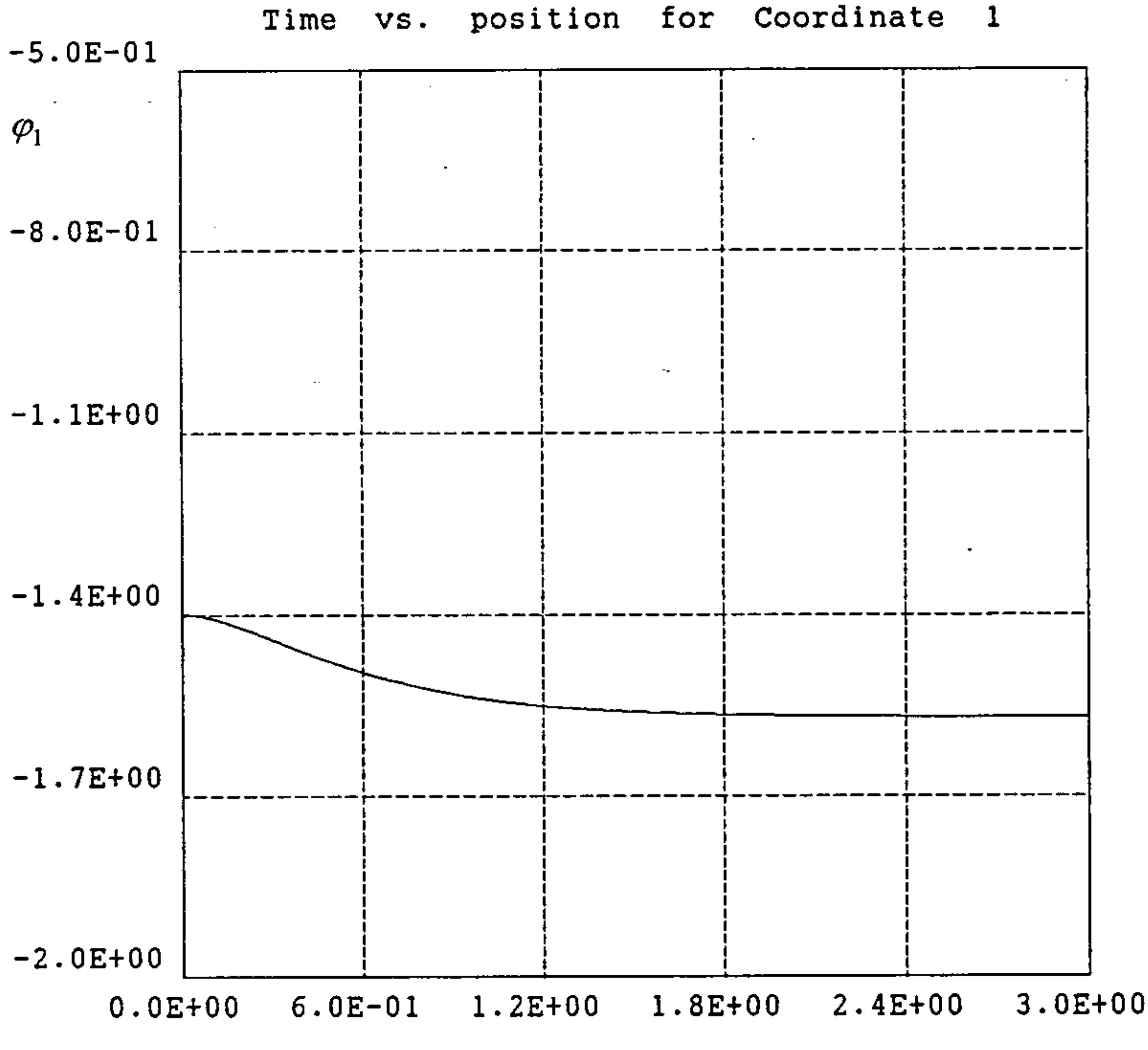

Figure (15) The Time History of the Coordinate φ_1 with Critical Damping.

Example 2

A Double Pendulum

In this example, forward dynamic analysis is performed for the system shown in Figure (16). The system is composed of a simple pendulum with another pendulum suspended on its free end. The system is solved for with free oscillations and the positions, velocities and accelerations of the two coordinates are found. The finite element model for the system is shown in Figure (16). The data for the system are as follows:

Mass lumped at joint (2) = 1 Kg

Length of the element 1 = 1 meter

Mass lumped at joint (3) = 1 Kg

Length of the element 2 = 1 meter

The initial angle that element 1 forms with the vertical = 10^{o}

The angle that element 2 forms with the vertical = 15^{o}

The system has two primary coordinates associated with it. They are coordinates φ_1 and φ_2. The primary freedom associated with the first primary coordinate is freedom 8, and it is freedom 7 for the other primary coordinate. Since we have free oscillations, i.e. no damping, we set the primary forces to zero. And the loads generated by those primary coordinates will be:

$$fpc(1) = 0$$

$$fpc(2) = 0$$

The input data for this example is shown in Appendix A.

The natural frequency of the system for this normal mode vibration is given by the equation [10]

$$\omega = \sqrt{\frac{g}{l}\left(2-\sqrt{2}\right)}$$

$$\omega = \sqrt{\frac{9.81}{1}\left(2-\sqrt{2}\right)}$$

$$= 2.3972 \quad \text{rad/sec}$$

and the period of small oscillations is

$$\tau = \frac{2\pi}{\omega}$$

$$= 2.621 \text{ sec.}$$

and the result obtained by the DAMES program is 2.65 seconds as can be seen in Figures (17)&(18) for a complete cycle.

Taking into account that the time increment for the analysis was 0.005 second, the obtained result is a good approximate of the exact solution.

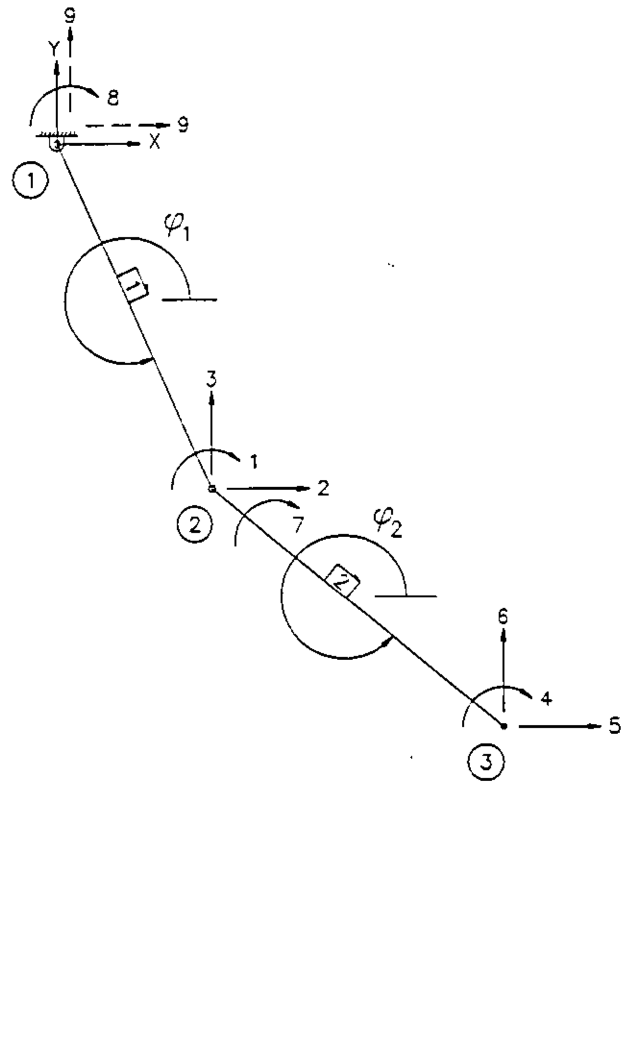

Figure (16) The Double Pendulum.

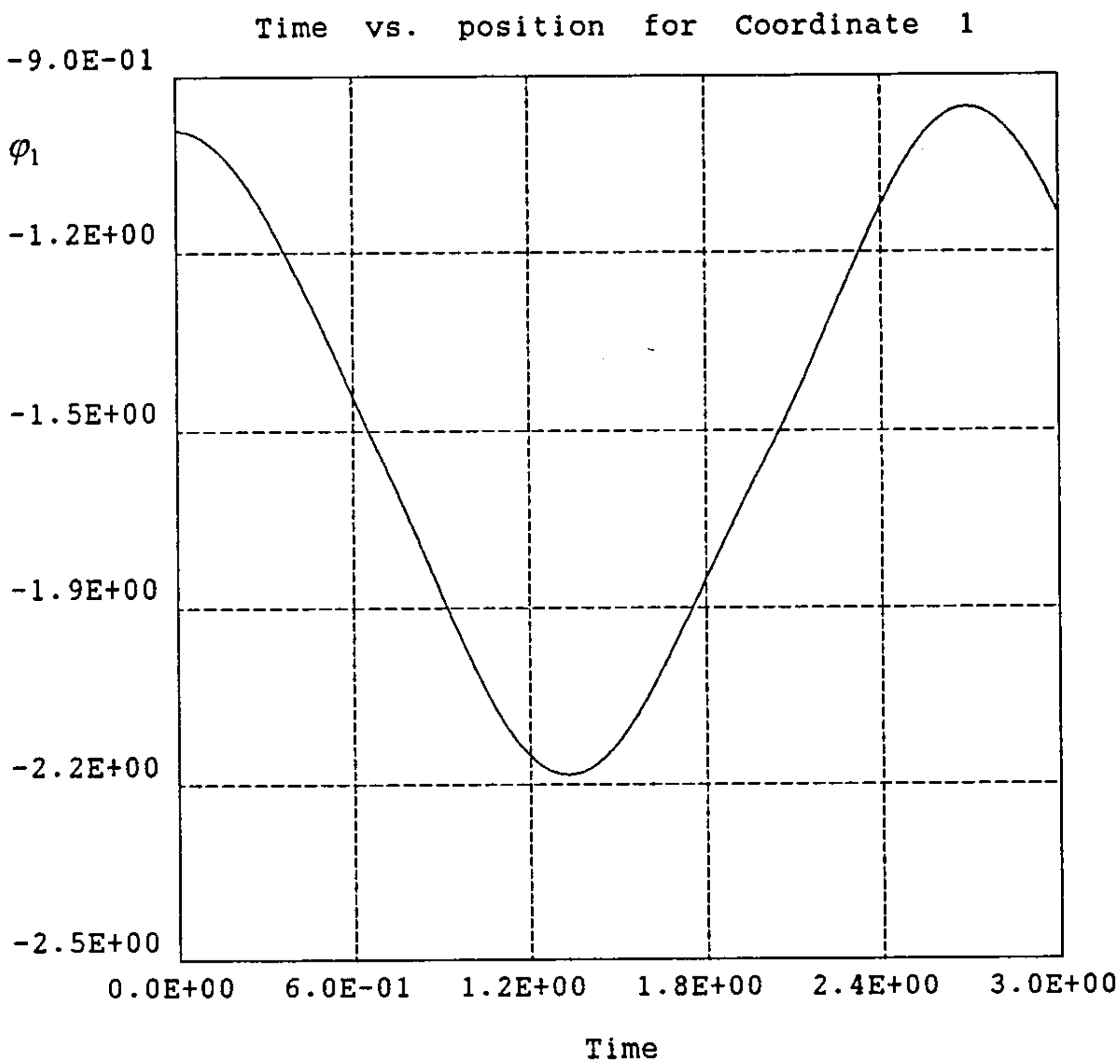

Figure (17) The Time History of the Coordinate φ_1.

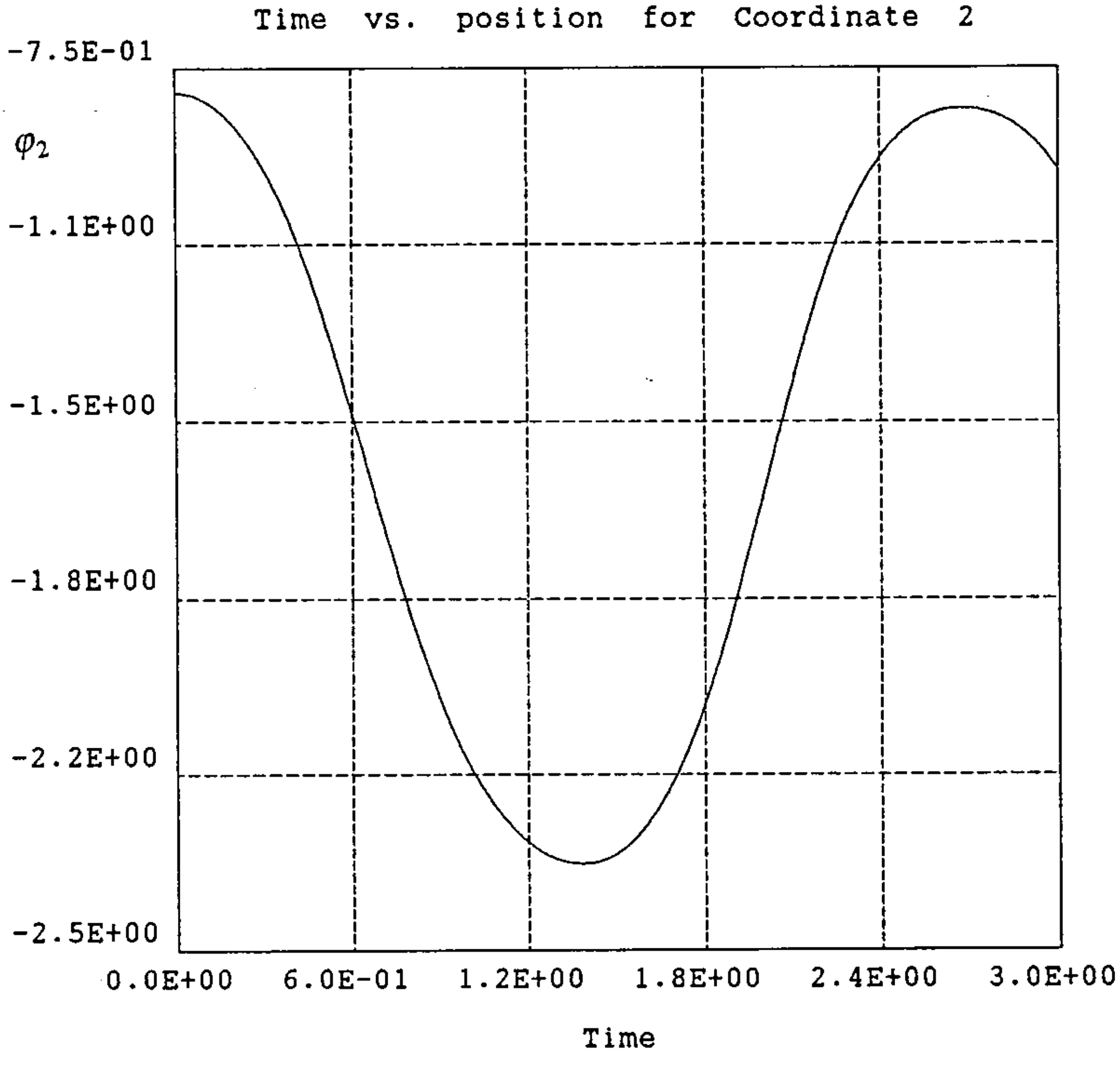

Figure (18) The Time History of the Coordinate φ_2.

Example 3

Automobile on a Rough Road [1]

In this example a two dimensional model of a car is used to simulate its motion over a sinusoidal road. Figure (19.A) shows the relation of the car to the model and Figure (19.B) shows the finite element model. The two wheel axles through points A and E move parallel to the road profile. The axes of the two springs are always normal to the line BD which is fixed to the car chassis. The mass and mass moment of inertia of the car are lumped at point C. The following data is used to define the parameters of the problem.

m = 647.67 lb (car mass)

I_1 = 12000 lb-in^2

$K_1 = K_2$ = 32,500 lb/in

$\varphi_{6_o} = \varphi_{7_o}$ = 20 in

BC = CD = 54 in

C1 = C2 = 1,000 lb-sec/in

The car is moving with a constant speed φ_1 = 550 in/sec. At time t=0 the front wheels are at the beginning of the curve y=f(x).

There are 3 time dependent constraints. The value of φ_1 is defined by the car speed and the initial position. Let the origin of the x-y coordinate system be at the beginning of the curve y=f(x). The following constraint equations can be written for φ_1

$$g(1) = 550 \times t - 108 - \varphi_1 = 0\,,$$

$$\frac{\partial g(1)}{\partial \varphi_1} = -1\,,$$

$$\frac{\partial g(1)}{\partial t} = 550\,,$$

$$\frac{d}{dt}\frac{\partial g(1)}{\partial \varphi_1} = 0\,, \text{ and}$$

$$\frac{d}{dt}\frac{\partial g(1)}{\partial t} = 0$$

The values of φ_2 and φ_5 are determined by the road profile and the value of φ_1. In case of φ_2 , the constraint equations are divided into two parts. First, when $\varphi_1 \le 0$, φ_2 is constant and equal to zero. Therefore the constraint equations are

$$g(2) = 0 - \varphi_2 = 0.0 ,$$

$$\frac{\partial g(2)}{\partial \varphi_2} = -1 ,$$

$$\frac{\partial g(2)}{\partial t} = 0.0 ,$$

$$\frac{d}{dt}\frac{\partial g(2)}{\partial \varphi_2} = 0.0 , \text{ and}$$

$$\frac{d}{dt}\frac{\partial g(2)}{\partial t} = 0.0$$

when $\varphi_1 > 0$, the constraint equation for φ_2 are

$$g(2) = 6 \times (1 - \cos(c\varphi_1)) - \varphi_2 = 0.0,$$

$$\frac{\partial g(2)}{\partial \varphi_1} = 6 \times c \times \sin(c\varphi_1) ,$$

$$\frac{\partial g(2)}{\partial \varphi_2} = -1,$$

$$\frac{\partial g(2)}{\partial t} = 0.0 ,$$

$$\frac{d}{dt}\frac{\partial g(2)}{\partial \varphi_1} = 6 \times c^2 \times \dot{\varphi}_1 \cos(c\varphi_1) ,$$

$$\frac{d}{dt}\frac{\partial g(2)}{\partial \varphi_2} = 0.0 , \text{ and}$$

$$\frac{d}{dt}\frac{\partial g(2)}{\partial t} = 0.0$$

The constraint equations for φ_5 are obtained as

$$g(3) = 6 \times (1 - \cos(c \times (\varphi_1 + \varphi_4))) - \varphi_5 = 0.0 ,$$

$$\frac{\partial g(3)}{\partial \varphi_1} = 6 \times c \times \sin(c \times (\varphi_1 + \varphi_4)) ,$$

$$\frac{\partial g(3)}{\partial \varphi_4} = 6 \times c \times \sin(c \times (\varphi_1 + \varphi_4)) ,$$

$$\frac{\partial g(3)}{\partial \varphi_5} = -1 ,$$

$$\frac{\partial g(3)}{\partial t} = 0.0 ,$$

$$\frac{d}{dt}\frac{\partial g(3)}{\partial \varphi_1} = 6 \times c^2 \times (\dot{\varphi}_1 + \dot{\varphi}_4)\cos(c \times (\varphi_1 + \varphi_4)),$$

$$\frac{d}{dt}\frac{\partial g(3)}{\partial \varphi_4} = 6 \times c^2 \times (\dot{\varphi}_1 + \dot{\varphi}_4)\cos(c \times (\varphi_1 + \varphi_4)),\ \text{and}$$

$$\frac{d}{dt}\frac{\partial g(3)}{\partial \varphi_5} = 0.0$$

These constraints equations are supplied in the *#uscons* section of the input data. A listing of the input data is given in Appendix B.

There are two primary coordinates associated with the two suspension springs and dampers. These coordinates are φ_6 and φ_7. The primary freedoms related to these primary coordinates are freedoms 7 and 8. These freedoms are in the directions of the axes of the two springs. The loads generated by these two primary coordinates are

$$fpc(1) = 325 \times (20 - \varphi_6) - 10 \times \dot{\varphi}_6$$

and

$$fpc(2) = 325 \times (20 - \varphi_7) - 10 \times \dot{\varphi}_7$$

These loads represent the spring and damping forces. They are supplied in the direction of freedoms 7 and 8 respectively. These primary forces are supplied in the *#drive* section of the input data given in Appendix B. The results of the dynamic response analysis of the car is given in Figures (20)-(24). Figure (20) shows the path of joint 5 in the XY plane. The time history of coordinates 6 and 7 are given in Figures (21) and (22). The forces produced by the springs and dampers at points B and D are shown in Figures (23) and (24). Figure (25) shows the path of joint 5 as given by the computer program DYMAC [4]. Comparing this Figure with Figure (20), it is clear that the two solutions are identical.

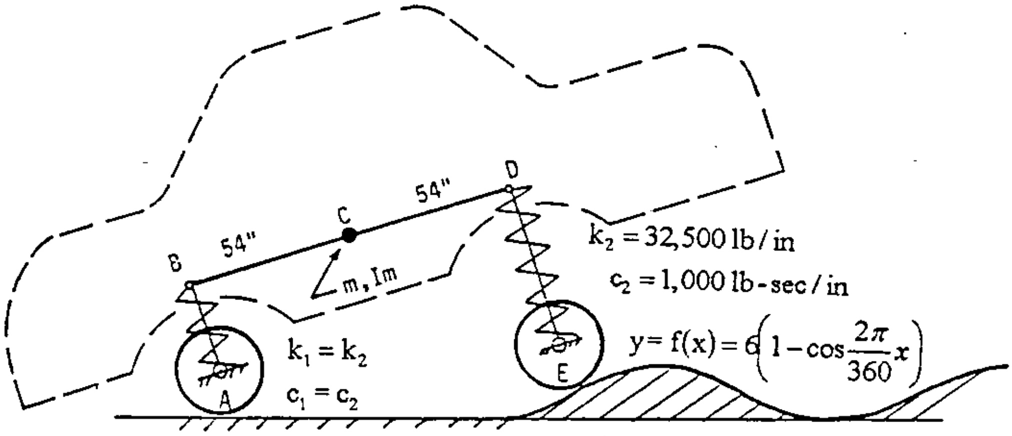

Figure (19.A) The Car and its 2-D Model[1].

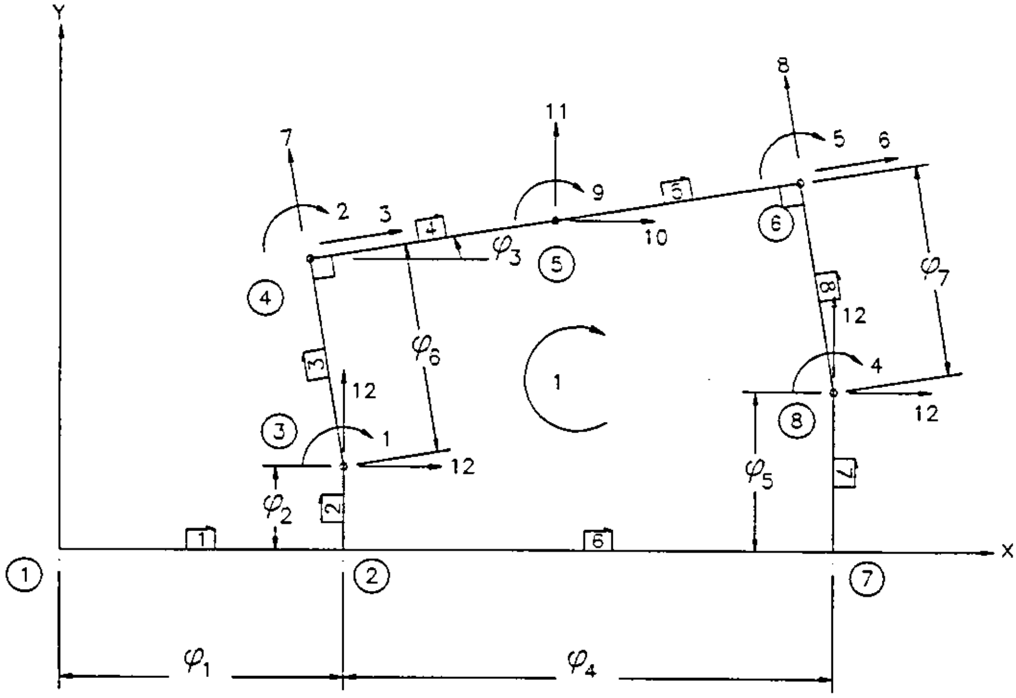

Figure (19.B) The Finite Element Model of the Car [1].

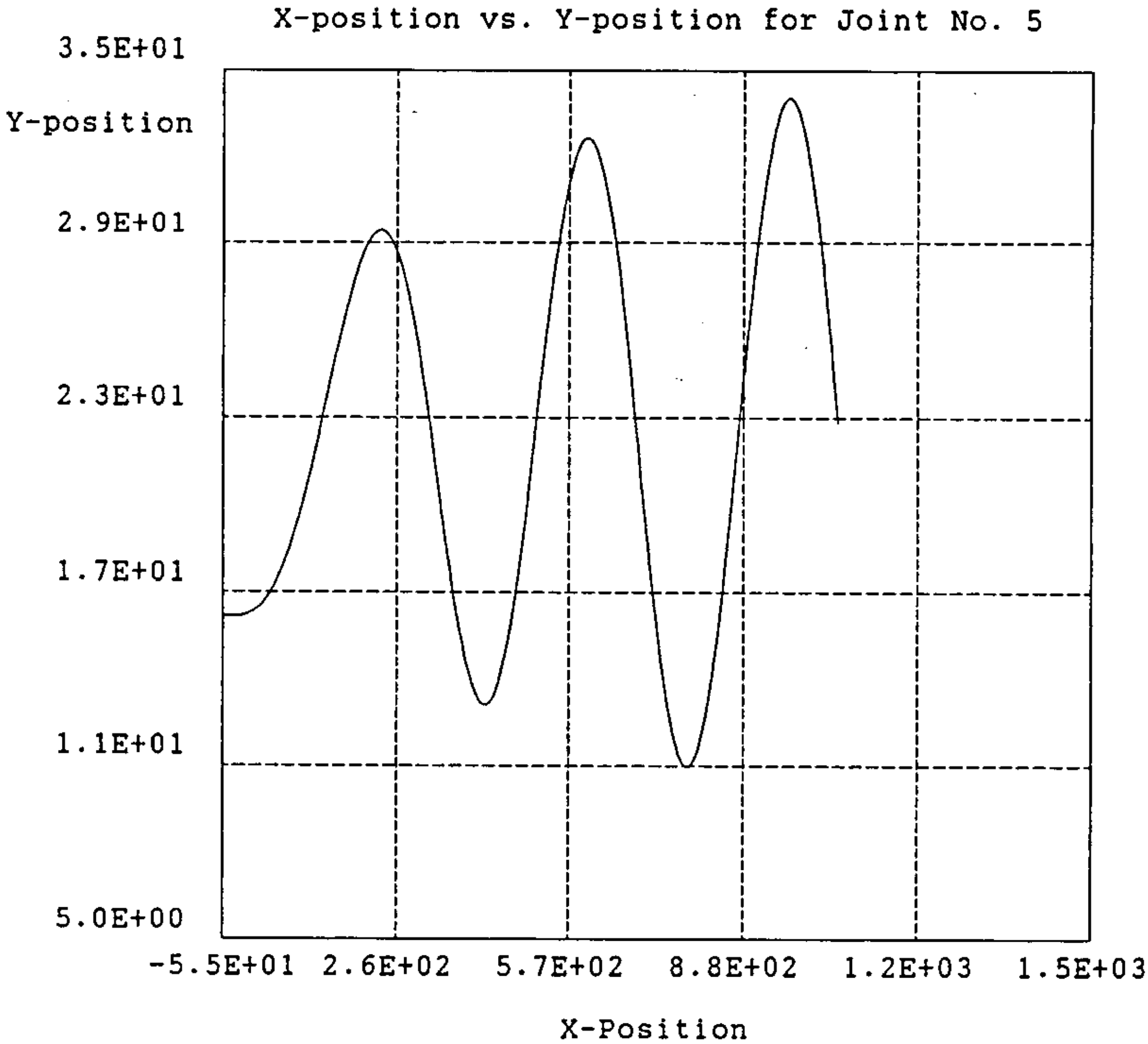

Figure (20) The Path of Joint 5 in the XY Plane.

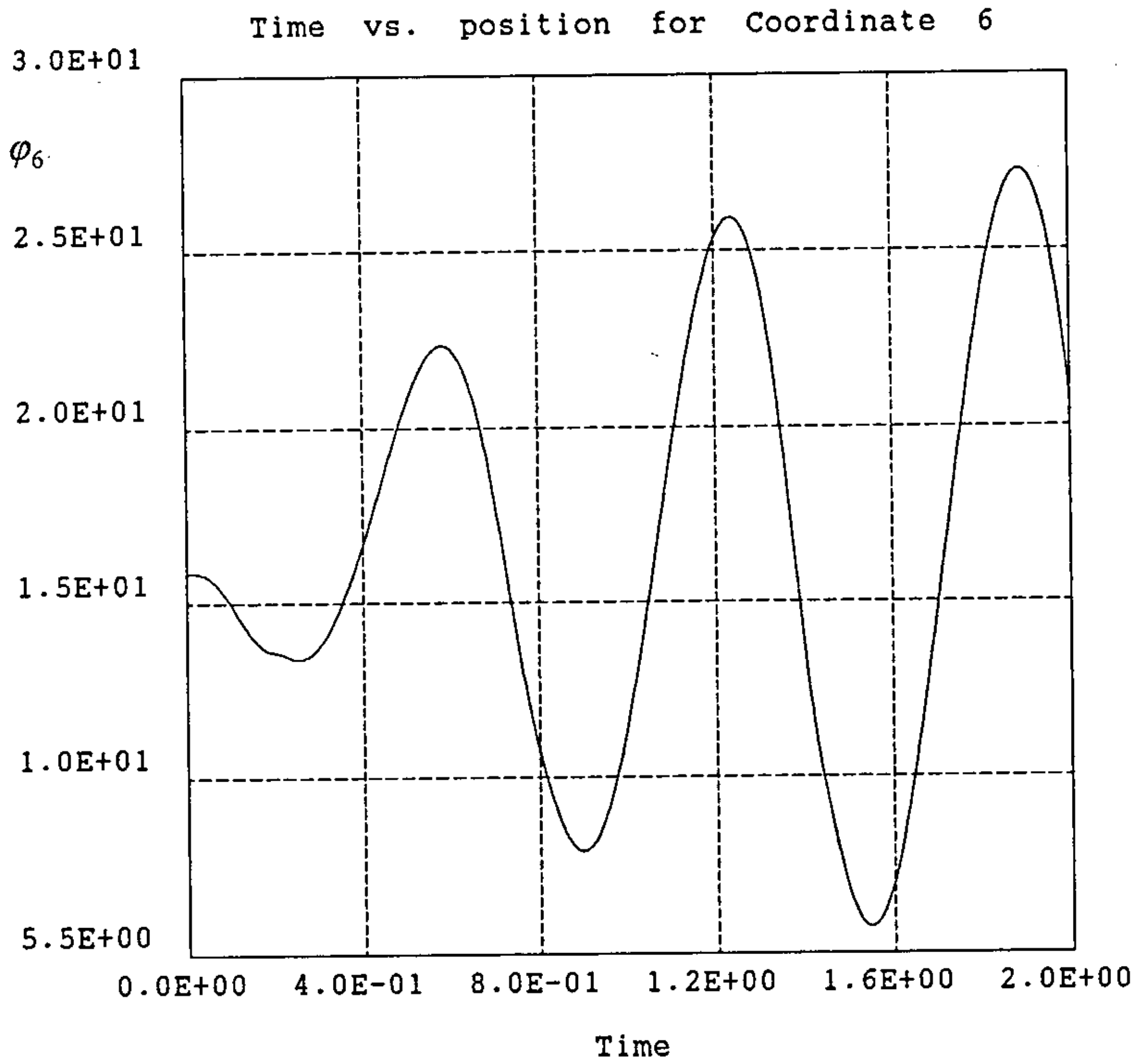

Figure (21) The Time History of the Coordinate φ_6.

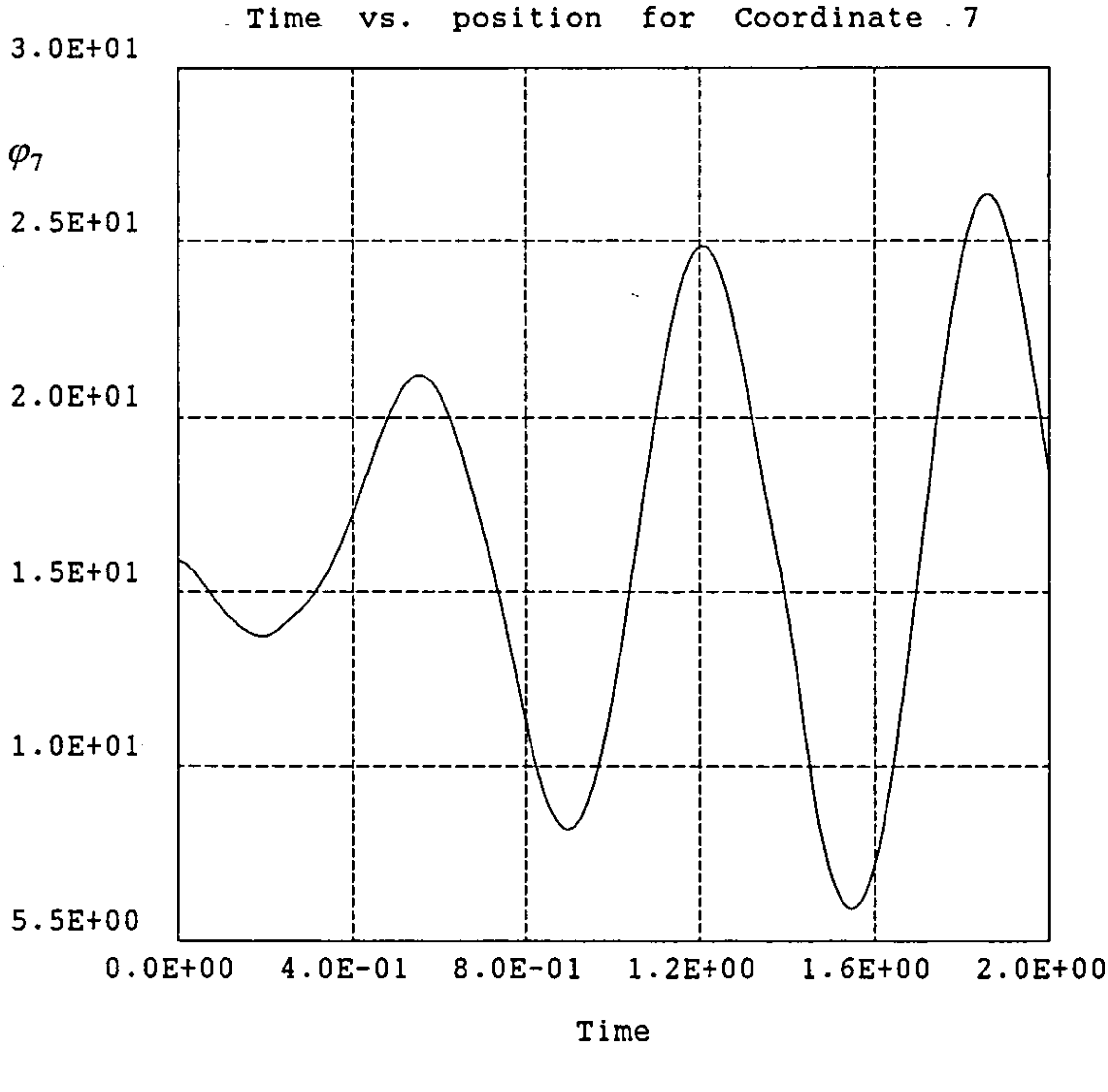

Figure (22) The Time History of the Coordinate φ_7.

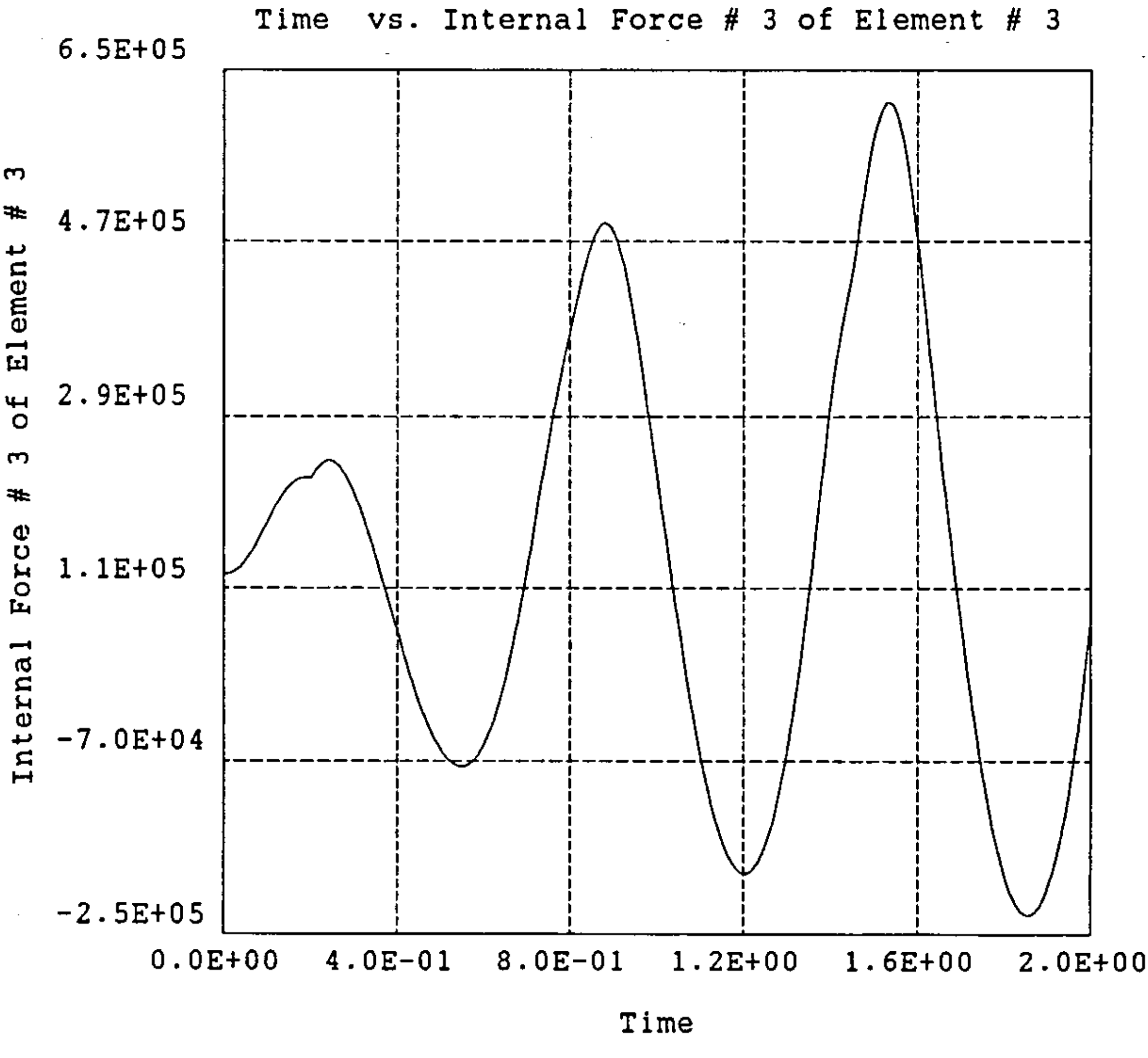

Figure (23) The Time History of the Axial Force of Element No.3.

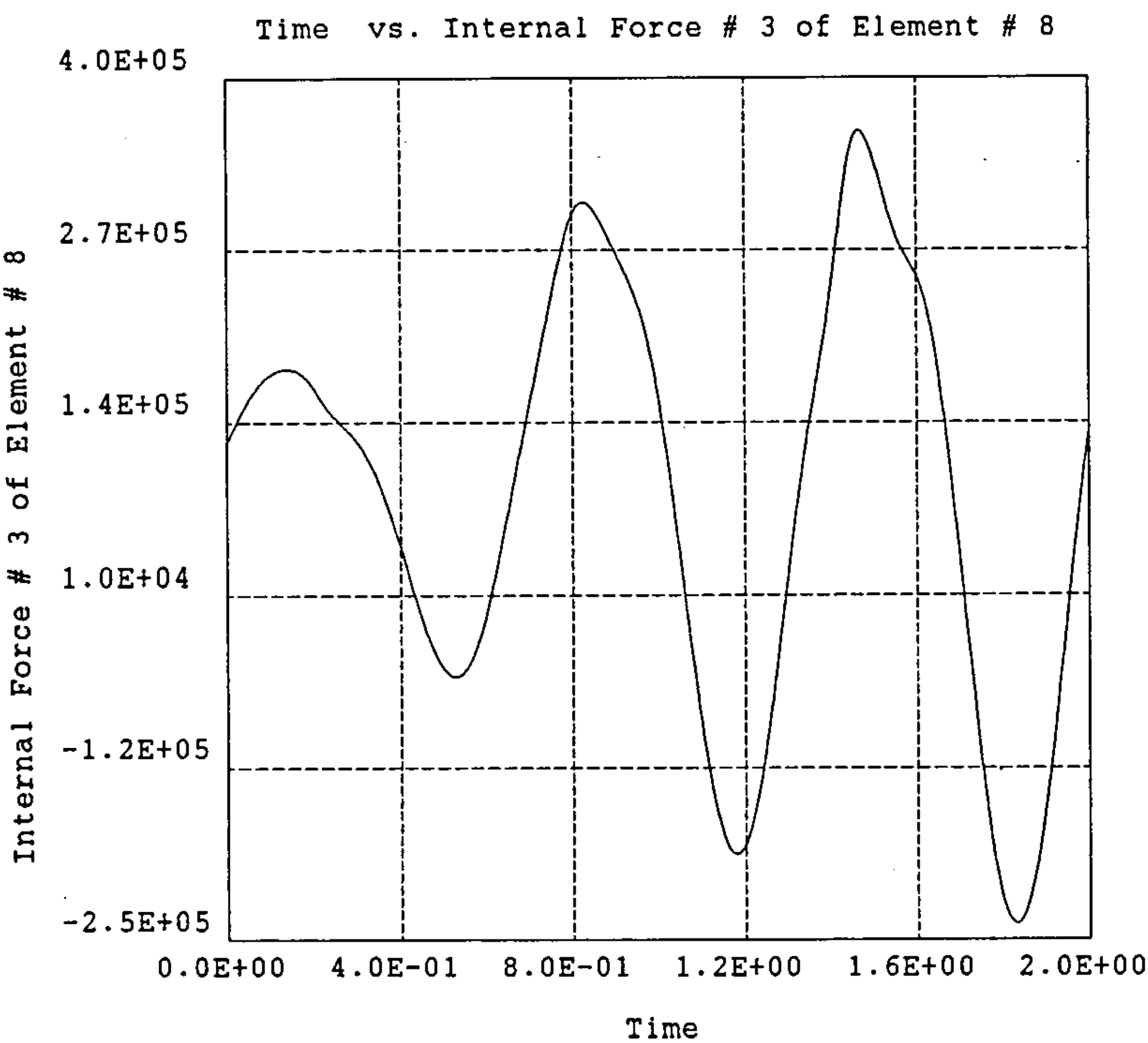

Figure (24) The Time History of the Axial Force of Element No.8.

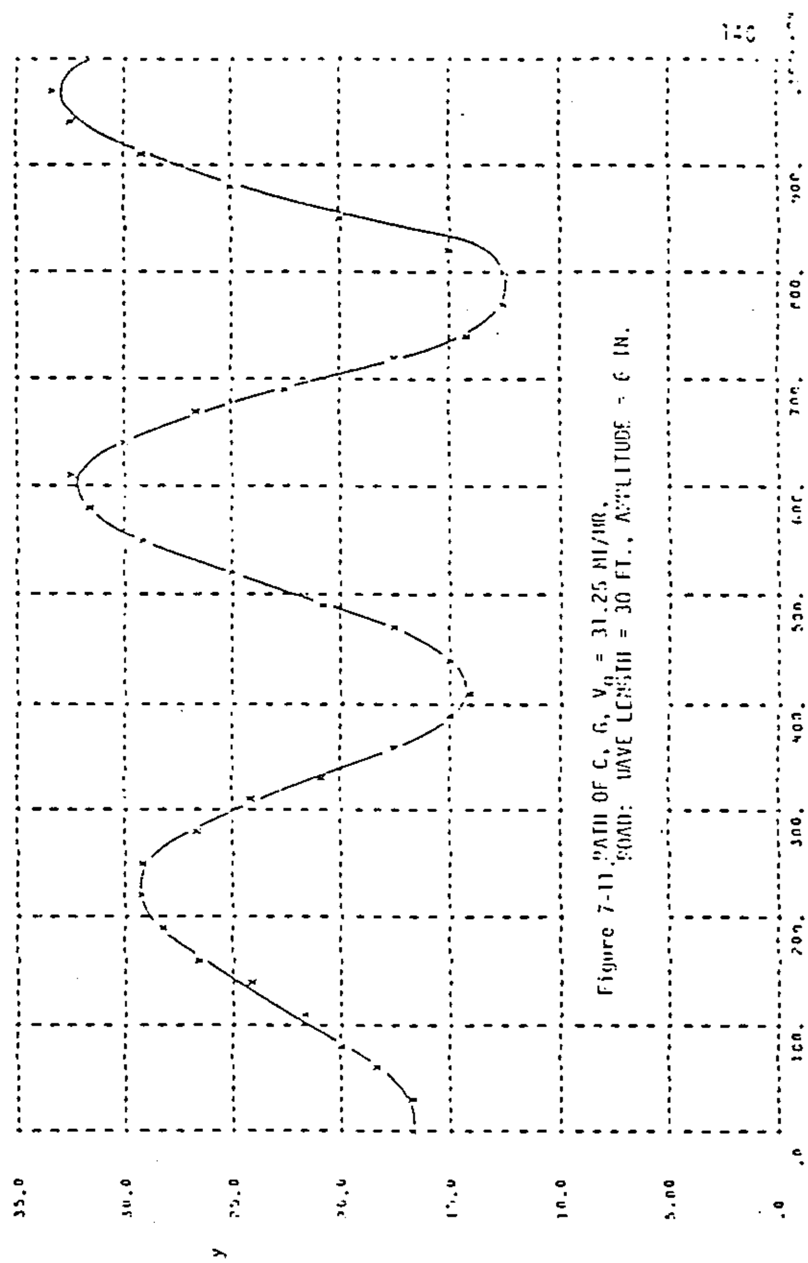

Figure (25) The Path of Joint 5 as Given by DYMAC (Compare This with Figure (20)) [1].

Example 4

Disk Cam with Radial Flat-Faced Follower

Figure (26.A) shows a disk cam with radial flat-faced follower. The cam rotates with constant angular velocity. The contact point between the cam and follower is at x,y, which is a distance l from the radial center line of the follower. The displacement of the follower from the origin is given by the following equation:

$$R = C + f(\theta)$$

Where the minimum radius of the cam is represented by C, and $f(\theta)$ represents the desired motion of the follower as a function of the angular displacement of the cam.

For this example, the flat-faced follower is driven through a total displacement of 38.1 mm. At the start of the cycle (zero displacement), the follower dwells for $\frac{\pi}{2}$ rad. It then moves 38.1 mm with cycloidal motion in $\frac{\pi}{2}$ rad. The follower dwells for $\frac{\pi}{2}$ rad and returns 38.1 mm with cycloidal motion in $\frac{\pi}{2}$ rad. A sketch of the displacement diagram is shown in Figure (27) [11]. The desired motion of the follower is expressed as a function of the angular displacement of the cam, and is given in the following equations [11]

$$f(\theta)_{O-A} = 0\,,$$

$$f(\theta)_{A-B} = \frac{76.2}{\pi}\left(\theta - \frac{\pi}{2}\right) - \frac{76.2}{4\pi}\sin(4\theta - 2\pi)$$

$$f(\theta)_{B-C} = 38.1$$

$$f(\theta)_{C-D} = 152.4 - \frac{76.2 \times \theta}{\pi} - \frac{76.2}{4\pi}\sin(4\theta - 6\pi)$$

The finite element model of the system is shown in Figure (26.B). The following data is used to define the parameters of the problem.

C = 32 mm (minimum radius of the cam)

m = 1 Kg (the mass lumped at joint 2)

k = 5 N/m (the spring constant)

time for complete cycle of the cam = 1 sec

initial length of element 1 = 0.032

initial length of element 2 = 0.108

total length of elements 1 & 2 = 140 mm

The system has two time dependent constraints. One for the length of element 1 and the other for the length of element 2. These constraints equations are as follows for the time $0 < tm \leq 0.25$ sec, the first constraint is for coordinate φ_1 and is given by

$$tc(1) = 0.032 - \varphi_1 = 0.032 - ti(1) = 0\ ,$$

for the time $0.25 < tm < 0.5$ sec, the first constraint equation is

$$tc(1) = \frac{0.0762}{\pi} \times \left(2 \times tm \times \pi - \frac{\pi}{2}\right) - \frac{0.0762}{4\pi} \times \sin(8 \times tm \times \pi - 2 \times \pi) - ti(1) + 0.032$$

for the time $0.5 \leq tm \leq 0.75$ sec, the first constraint equation is

$$tc(1) = 0.0381 - ti(1) + 0.032$$

for the time $0.75 < tm \leq 1.0$ sec, the first constraint equation is

$$tc(1) = 0.0381 \times \left(1.0 - \frac{\left(2 \times tm \times \pi - \frac{3\pi}{2}\right)}{\frac{\pi}{2}} + \frac{1}{2\pi} \sin(8 \times tm \times \pi - 6 \times \pi)\right) - ti(1) + 0.032$$

The other constraint is for coordinate 2 , and is given as follows

$$tc(2) = 0.140 - \varphi_1 - \varphi_2 = 0.140 - ti(1) - ti(2) = 0$$

The spring force is introduced in the *#usload* section of the input data as an external load, and is given by

$$usl(1) = -5.0 \times (\varphi_2 - 0.108)$$

A listing of the input data for this example is given in Appendix A and Appendix B.

The results for time history for coordinate 1, is given in Figure (28). By comparing this Figure (28) with Figure (27), taking into consideration the minimum radius of the cam, it is clear that those figures are identical. Figure (29) shows the force generated by the spring and the inertia of the mass.

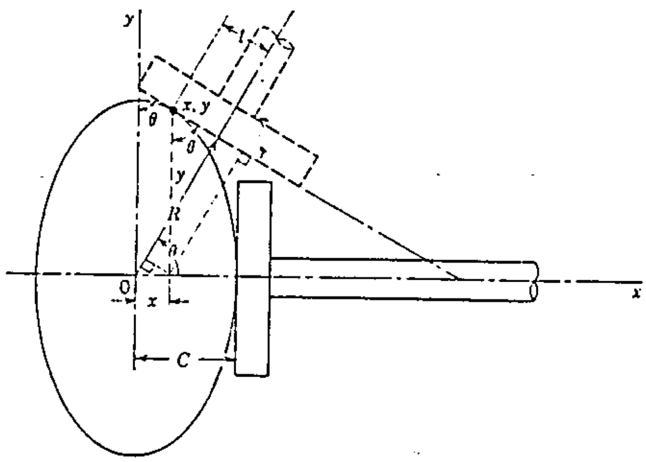

Figure (26.A) The Disk Cam with Radial Flat-Faced Follower. [11]

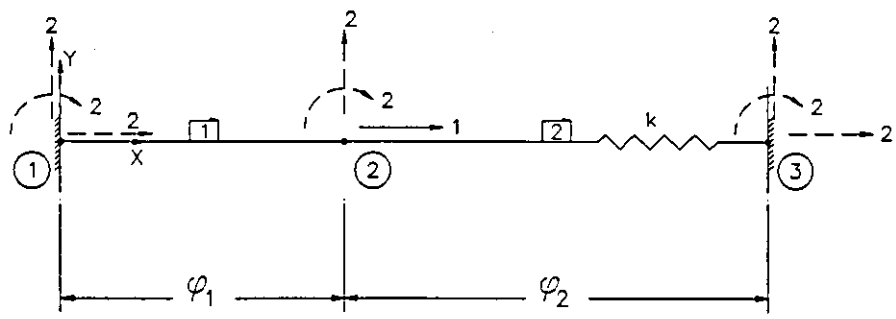

Figure (26.B) The Finite Element Model of the Disk Cam with the Follower.

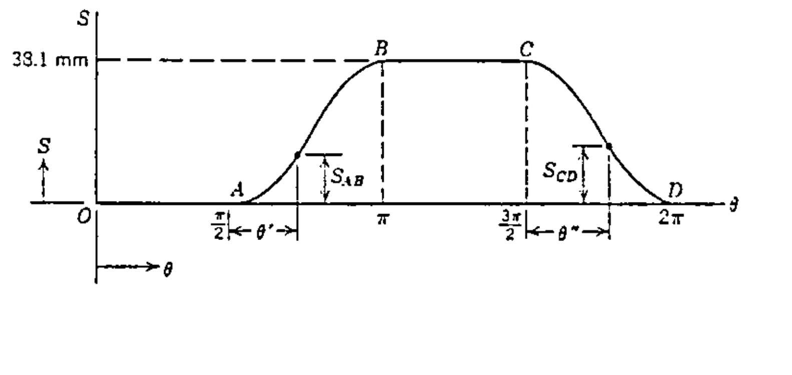

Figure (27) Sketch of the Displacement Diagram. [11]

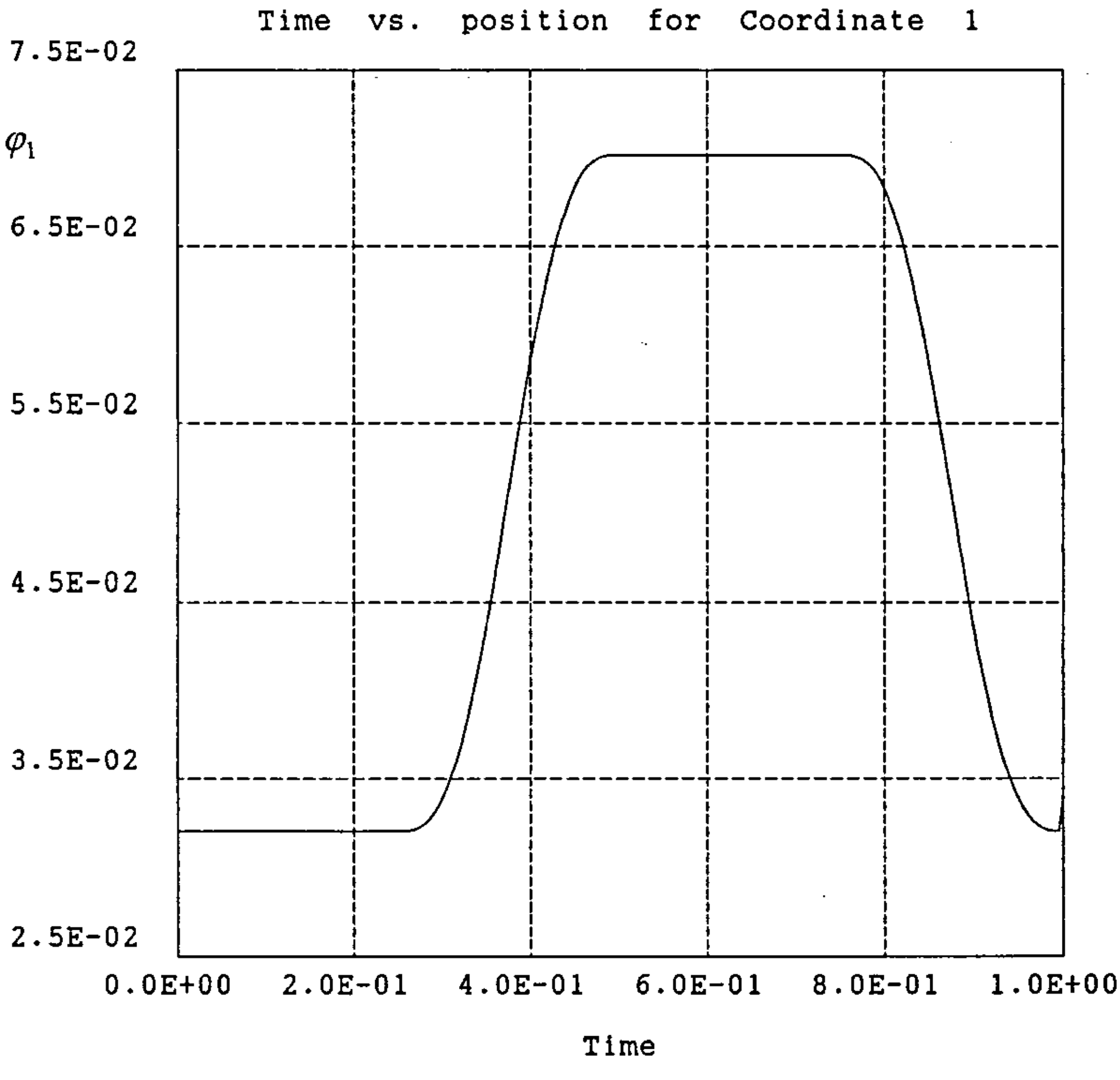

Figure (28) The Time History of the Coordinate φ_1.

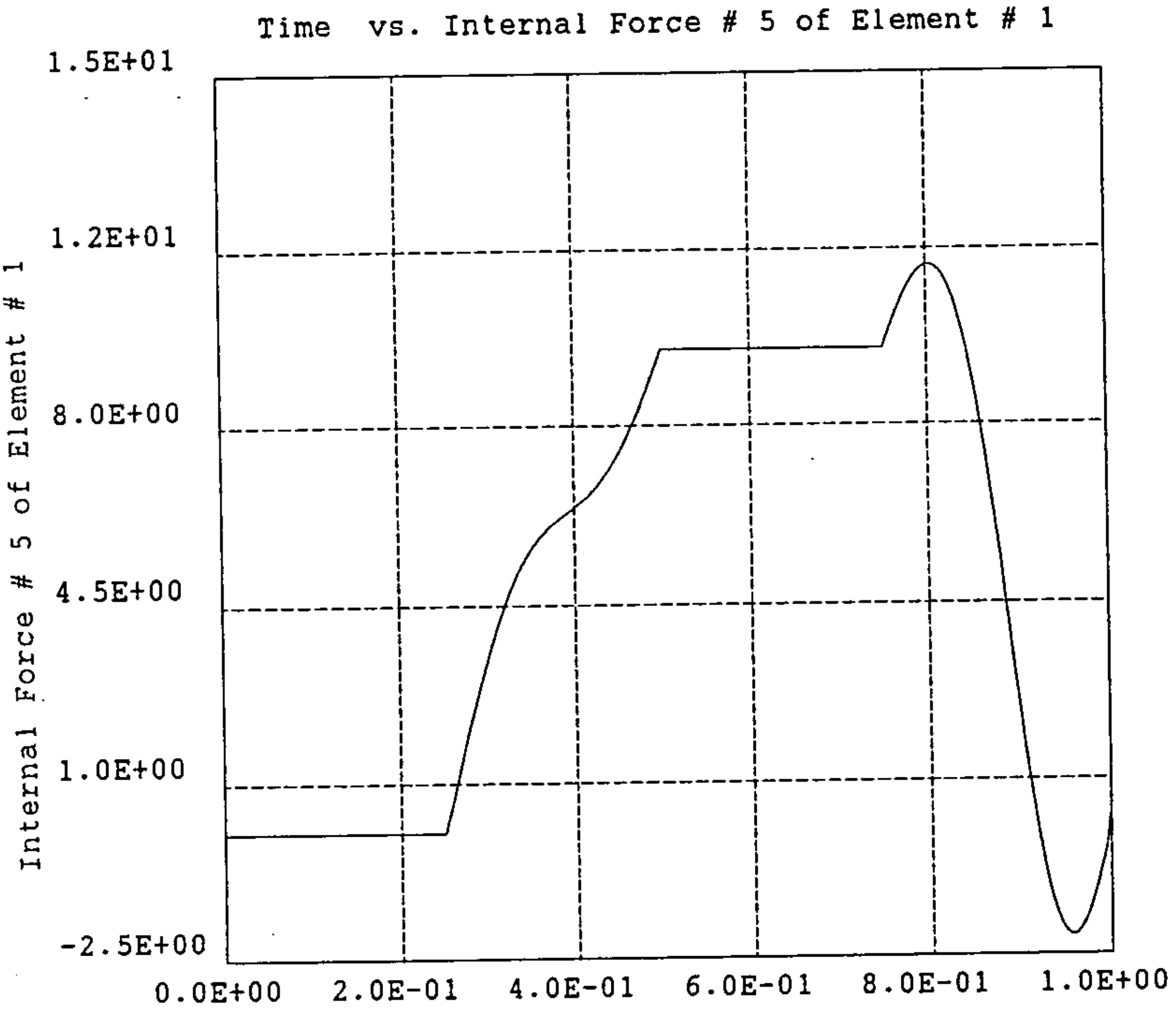

Figure (29) The Time History of the Axial Force of Element No.1.

Example 5

A 3-R Mixed-Loop Planar Robot [1]

Figure (30) shows a 3-R mixed-loop planar robot. The end effector at joint 7 must trace the circular path shown with the defined displacement function S. The orientation of element 6 must remain constant at an angle $\varphi_4 = 0$. There are three input actuators, one at joint 1 which defines the orientation of element 1 and the other two are at the double joint 3 and they define the orientations of elements 3 and 7. The inertias of the rest of the elements are neglected. A load of 40 lbm is lumped at joint 7 for the end effector assembly. The combined mass and mass moment of inertia of the two actuators at joint 3 are 60 lbm and 2.8 $lb_f.in.sec^2$.

For the dimensions and the inertias shown on Figure (30), we need to find the positions, velocities, accelerations, and input torques for the three actuators.

The input data for this example is taken directly from Figure (30). It is listed in Appendix A. There are no applied loads or known driving forces or torques. There are two constraints equations. One time-dependent equation for φ_3 defining the orientation of element 9 using the function S. This constraint equation is

$$tc(1) = g(\varphi_3, t) = \frac{S(t)}{r} - \varphi_3 = 0$$

The other equation is path constraint and it defines the orientation of the end effector. It is given by

$$ts(1) = S(\varphi_4) = \varphi_{4_o} - \varphi_4 = 0$$

Finally, the derivatives of these two constraints equations are

$$ptc(1,3) = \frac{\partial g}{\partial \varphi_3} = -1 ,$$

$$pttc(1) = \frac{\partial g}{\partial t} = \frac{1}{r}\frac{\partial S(t)}{\partial t} ,$$

$$dptc(1,3) = \frac{d}{dt}\frac{\partial g}{\partial \varphi_3} = 0 ,$$

$$dpttc(1) = \frac{d}{dt}\frac{\partial g}{\partial t} = \frac{1}{r}\frac{d}{dt}\frac{\partial S(t)}{\partial t} ,$$

$$pts(1,4) = \frac{\partial S}{\partial \varphi_4} = -1 ,$$

$$dpts(1,4) = \frac{d}{dt}\frac{\partial S}{\partial \varphi_4} = 0 \ .$$

The results of the dynamic analysis are shown on Figures (31)-(38). The time history of the position for φ_1 and φ_6 are shown on Figures (31)-(32). The time history of their velocities are shown on Figures (33)-(34), and their accelerations are shown on Figures (35)-(36). The time history of the driving torques for two actuators are shown on Figures (37)-(38).

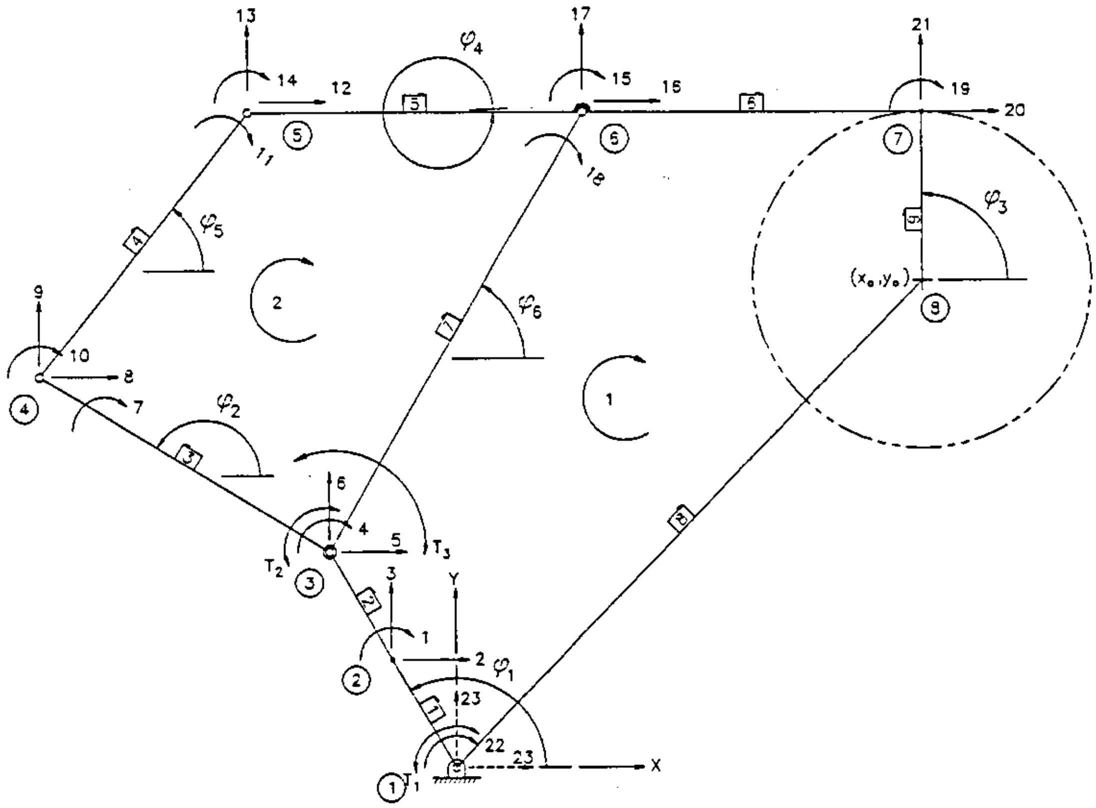

Figure (30) A 3R Mixed-Loop Robot. [1]

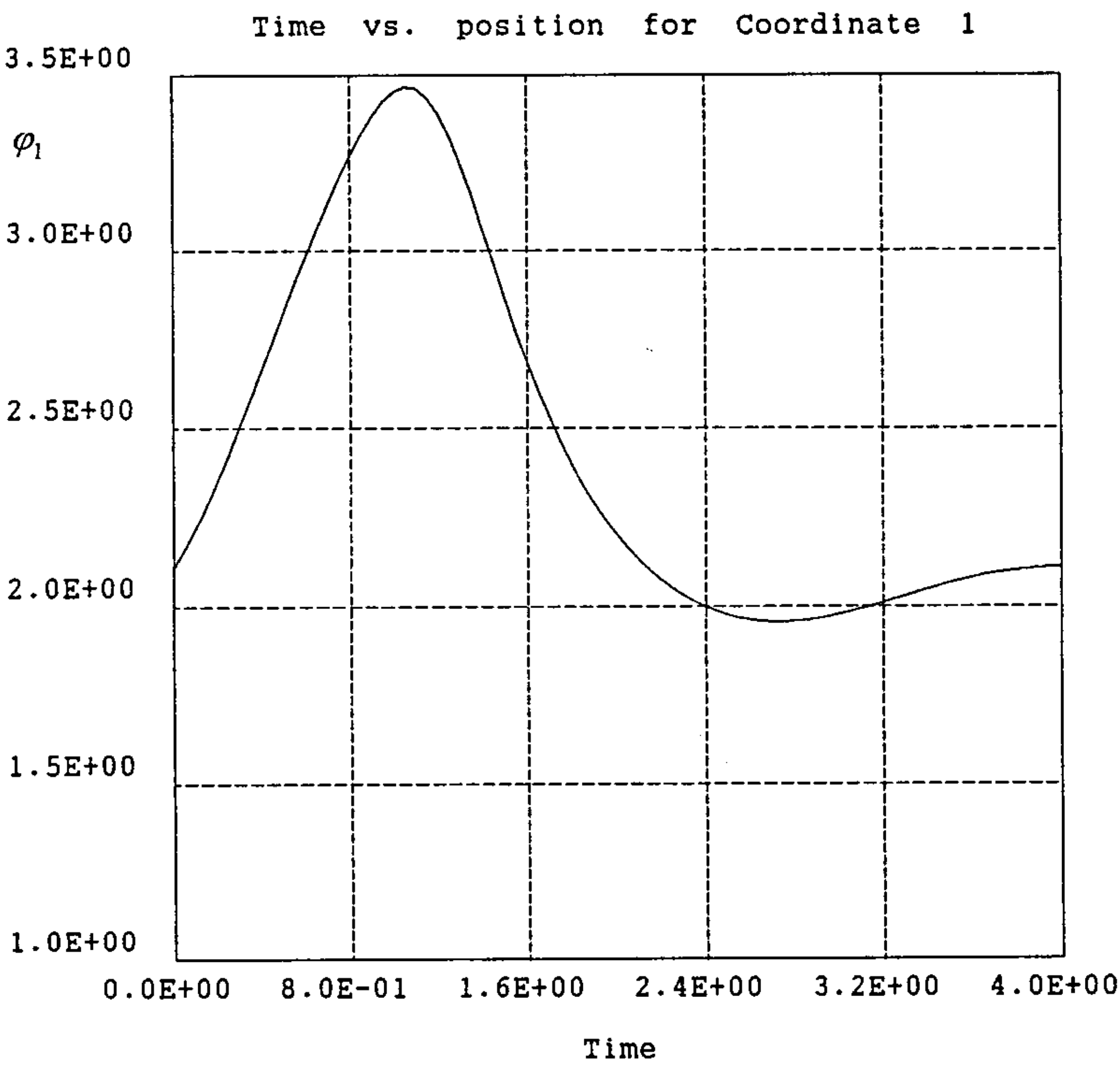

Figure (31) The Time History of the Coordinate φ_1.

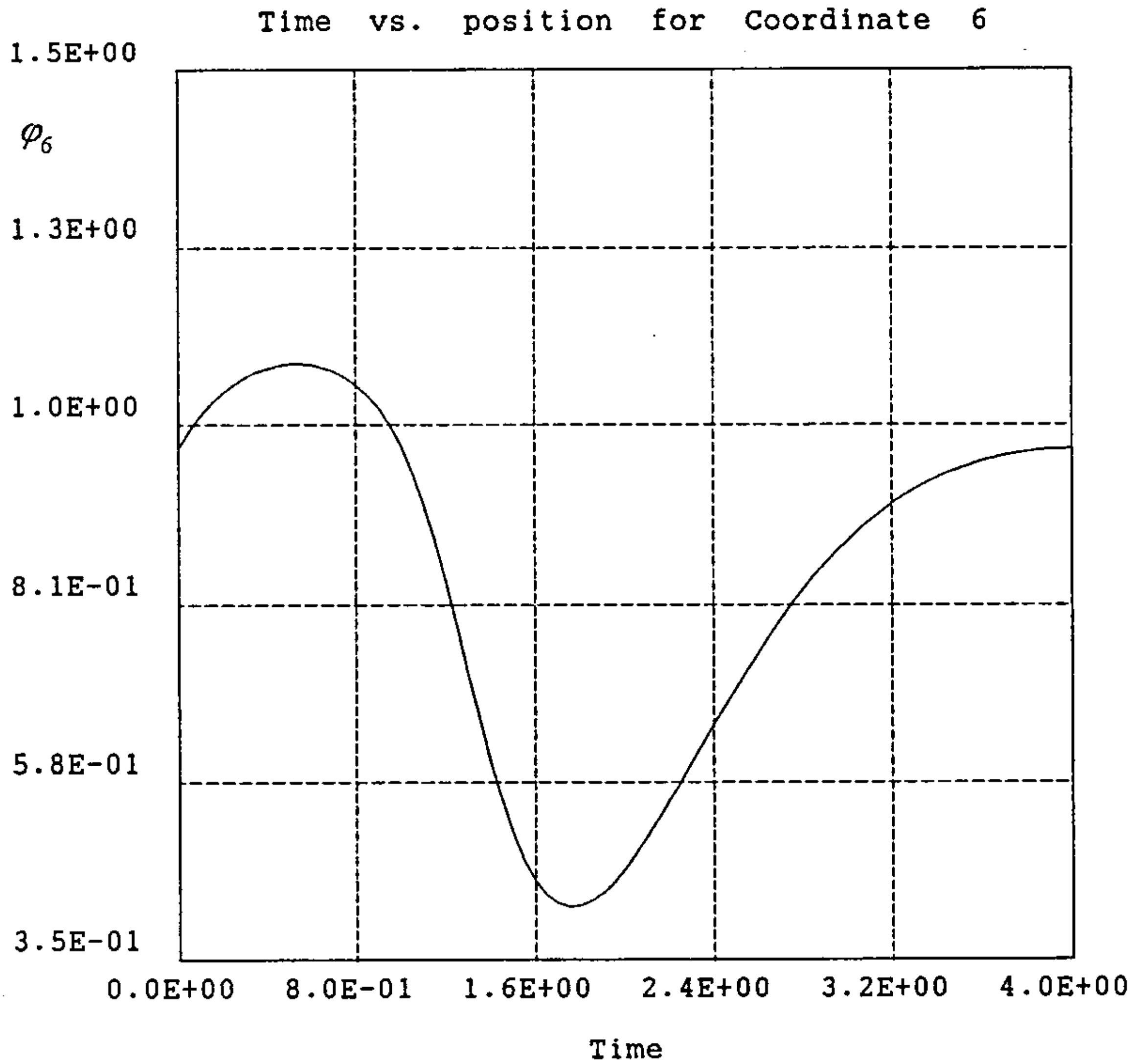

Figure (32) The Time History of the Coordinate φ_6.

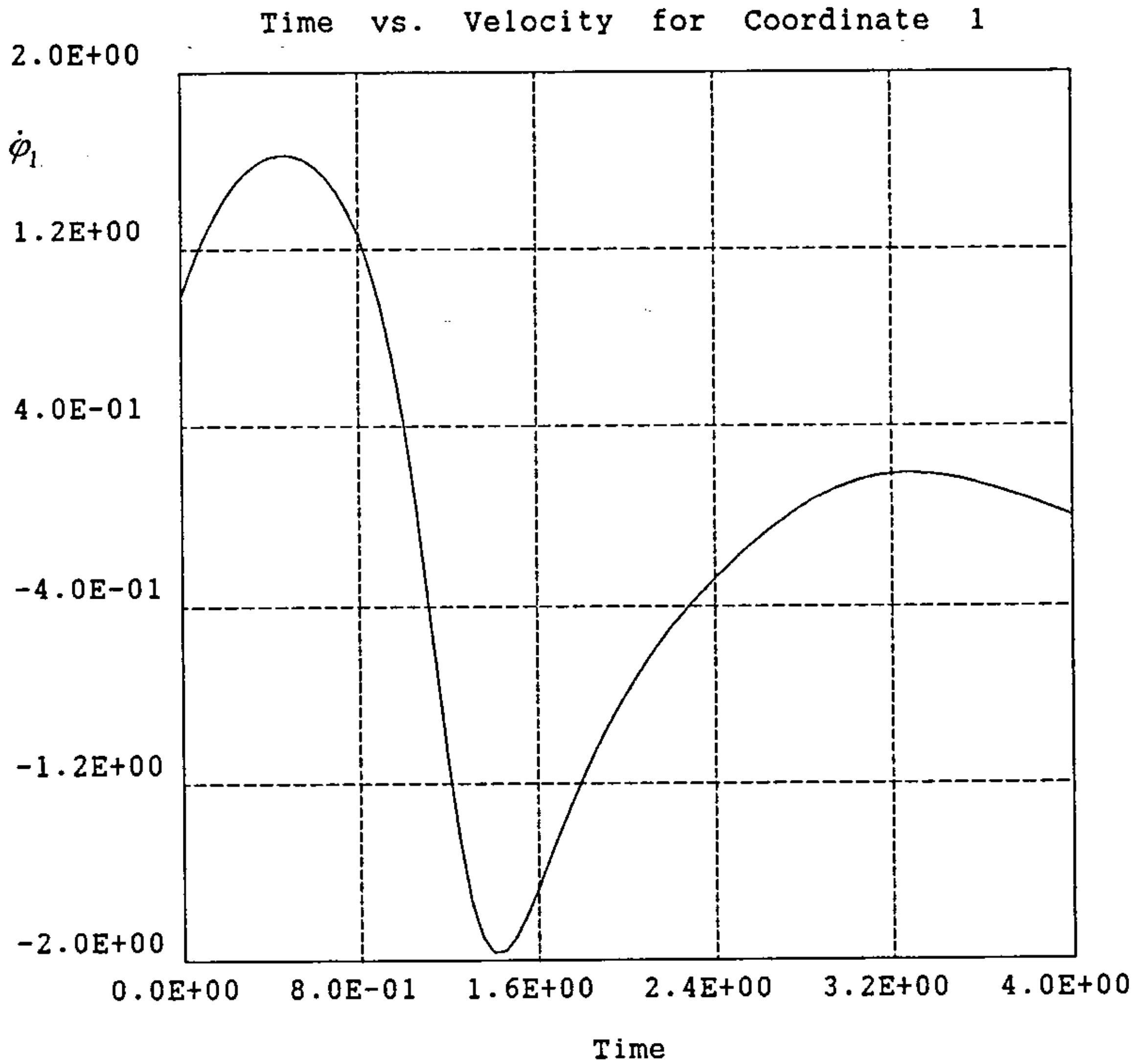

Figure (33) The Time History of the Velocity of $\varphi_1(\dot{\varphi}_1)$.

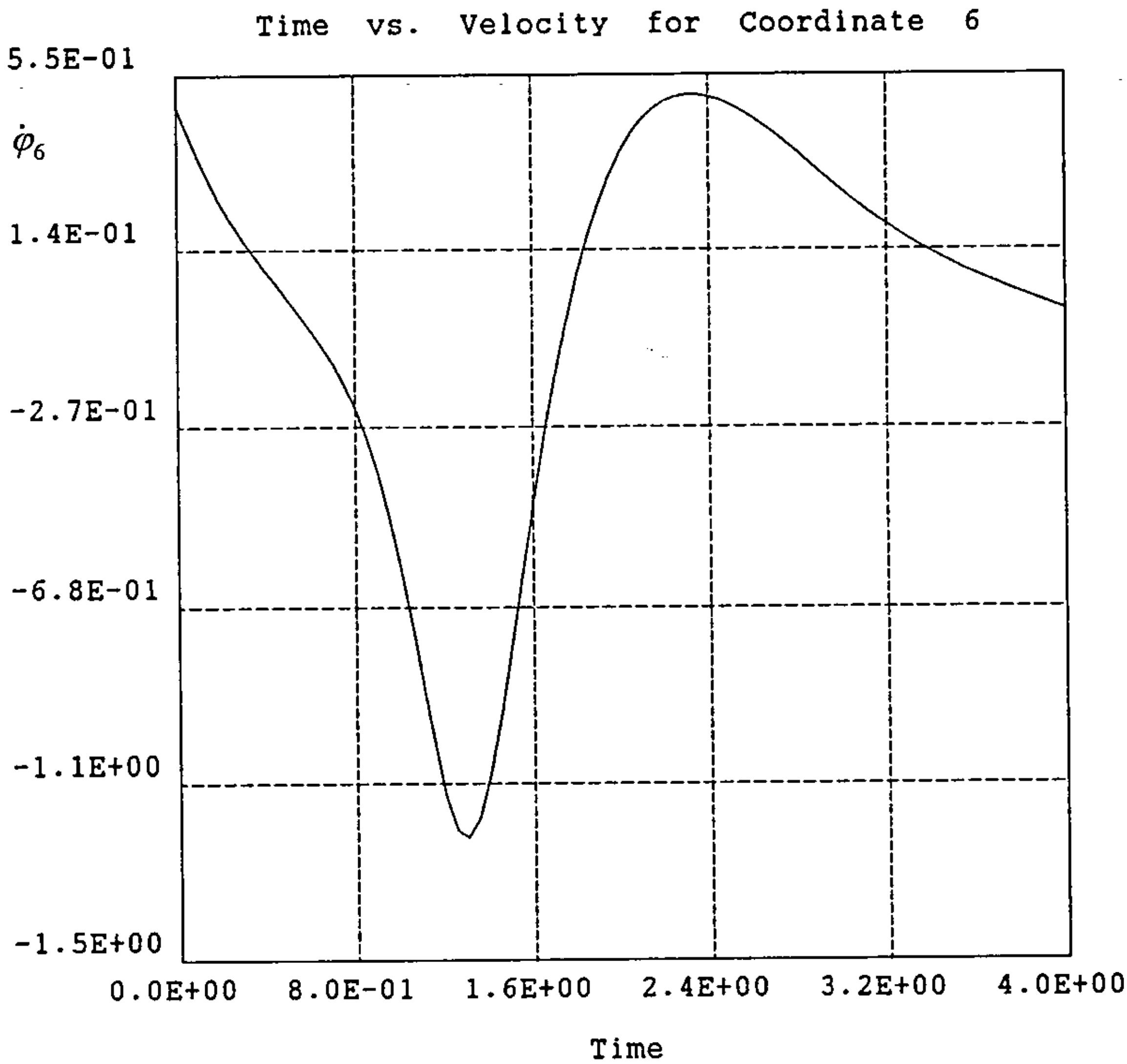

Figure (34) The Time History of the Velocity of $\varphi_6(\dot{\varphi}_6)$.

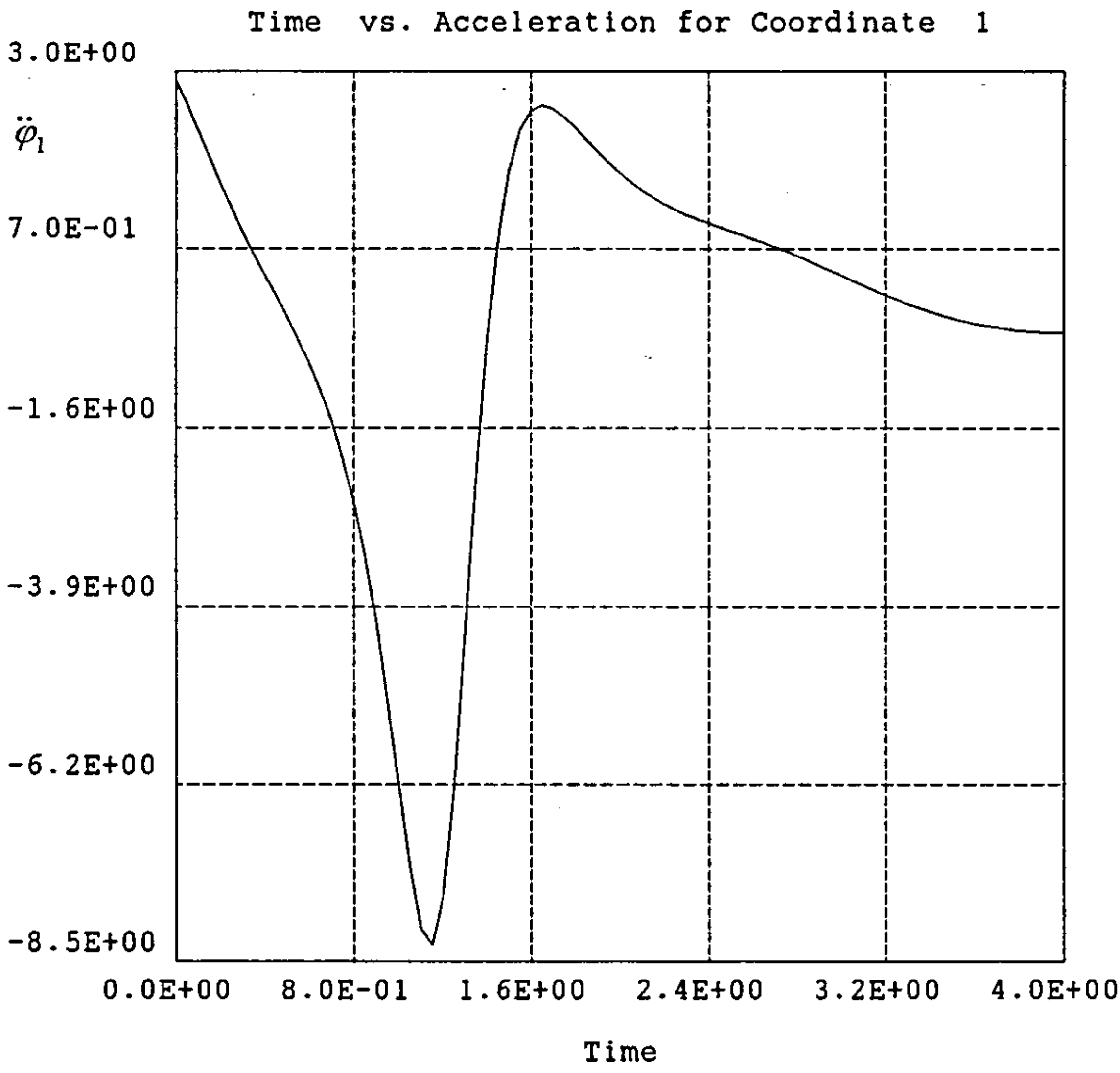

Figure (35) The Time History of the Acceleration of $\varphi_1(\ddot{\varphi}_1)$.

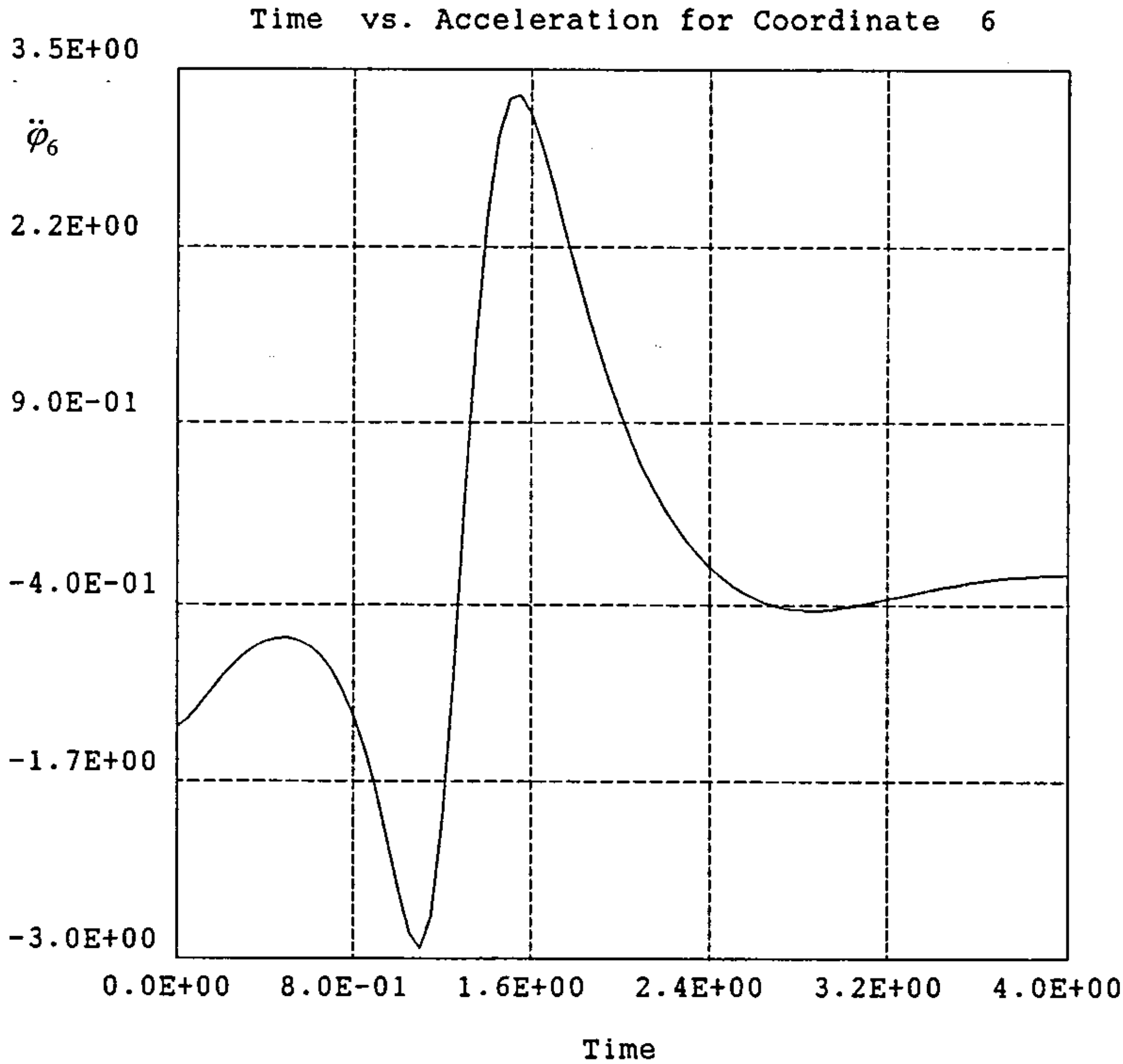

Figure (36) The Time History of the Acceleration of $\varphi_6(\ddot{\varphi}_6)$.

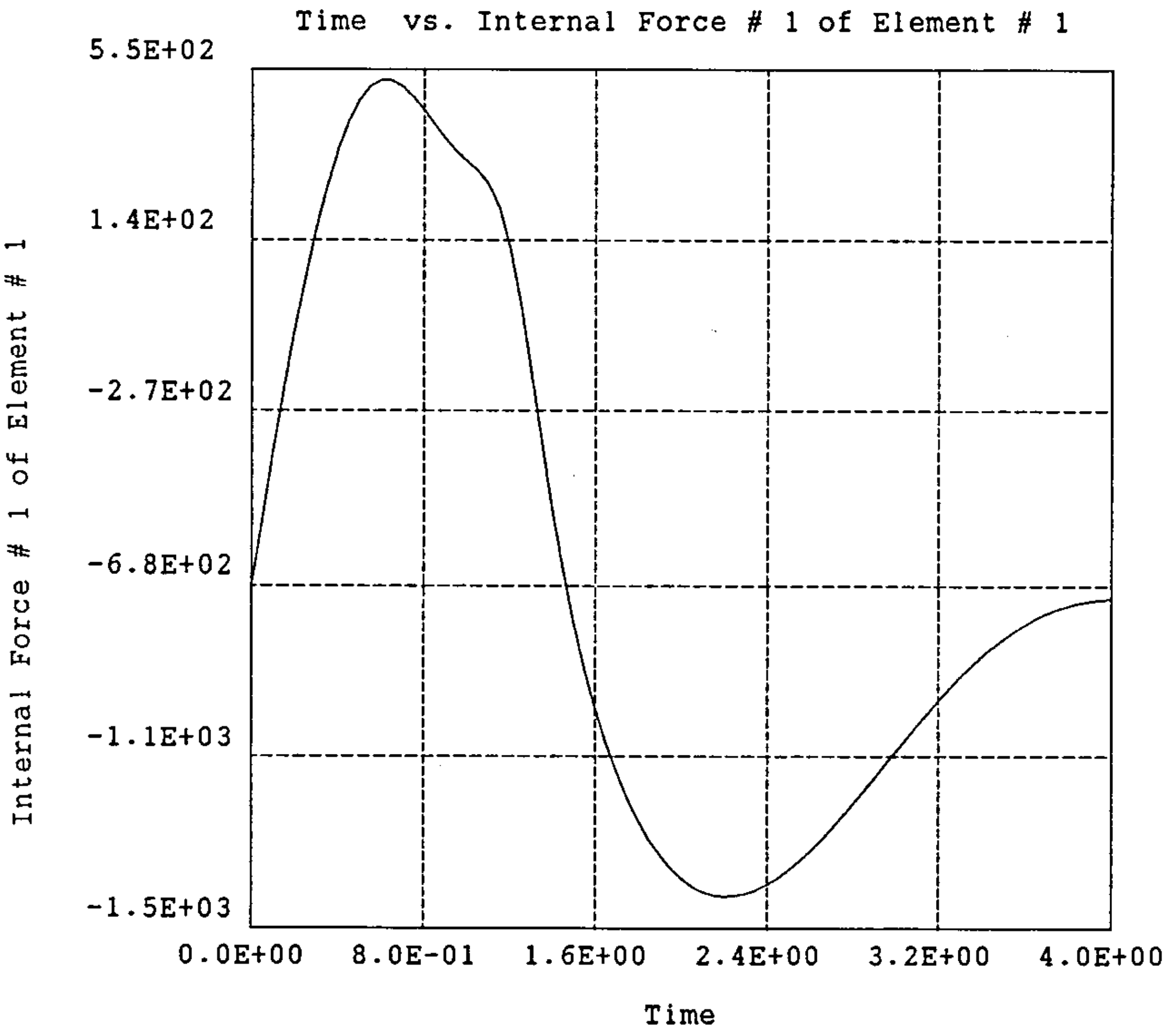

Figure (37) The Time History of the Driving Torque T_1.

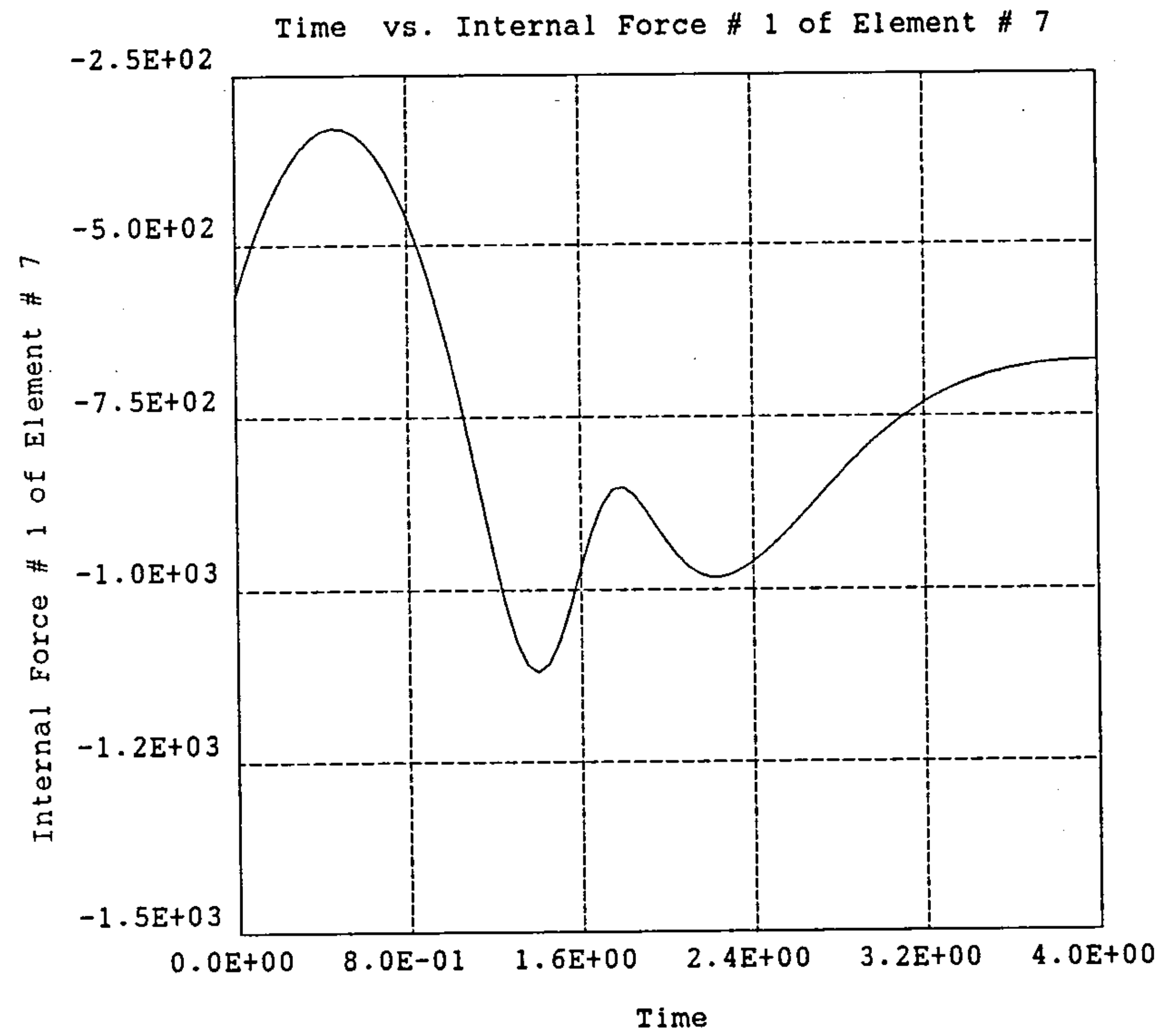

Figure (38) The Time History of the Driving Torque T_3.

Example 6

A Slider Crank Mechanism

In this example we are to find the static equilibrium configuration of the slider crank mechanism shown in Figure (39) under the action of gravity.

Manual Solution:

Step(1): Find the virtual work expression

(a) - Find the position vector

- For joint 2

$$R_{x_2} = \frac{a}{2}\cos\varphi_1$$

$$R_{y_2} = \frac{a}{2}\sin\varphi_1$$

- For joint 3

$$R_{x_3} = a\cos\varphi_1 + \frac{b}{2}\cos\varphi_2$$

$$R_{y_3} = a\sin\varphi_1 + \frac{b}{2}\sin\varphi_2$$

- For joint 4

$$R_{x_4} = \varphi_3$$

$$R_{y_4} = 0$$

(b) - Find the variation of the position vector in the direction of the applied load.

- For joint 2

$$\delta R_{y_2} = \frac{a}{2}\delta\varphi_1\cos\varphi_1$$

- For joint 3

$$\delta R_{y_3} = a\delta\varphi_1\cos\varphi_1 + \frac{b}{2}\delta\varphi_2\cos\varphi_2$$

- For joint 4

$$\delta R_{x_4} = \delta\varphi_3$$

(c) - Write the virtual work expression for each joint

- For joint 2

$$\delta u_2 = m_1 g \delta R_{y_2} = \left(m_1 g \frac{a}{2} \cos\varphi_1 \right) \delta\varphi_1$$

- For joint 3

$$\delta u_3 = m_2 g \delta R_{y_3} = (m_2 g a \cos\varphi_1)\delta\varphi_1 + \left(m_2 g \frac{b}{2} \cos\varphi_2 \right) \delta\varphi_2$$

- For joint 4

$$\delta u_4 = k\left(\varphi_{3_o} - \varphi_3\right)\delta\varphi_3$$

where φ_{3_o} is the value of φ_3 at which the spring is unloaded.

- if there is a moment M at joint 1 we add

$$\delta u_1 = M\delta\varphi_1$$

(d) - Define the C coefficients in Equation 1

$$C_1 = m_1 g \frac{a}{2} \cos\varphi_1 + m_2 g a \cos\varphi_1$$

$$C_2 = m_2 g \frac{b}{2} \cos\varphi_2$$

$$C_3 = k\left(\varphi_{3_o} - \varphi_3\right)$$

<u>Step(2): Write the constraints Equations:</u>

(a) One closed loop gives 2 equations for the x and y directions

$$f_1 = a\cos\varphi_1 + b\cos\varphi_2 - \varphi_3 = 0$$

$$f_1 = a\sin\varphi_1 + b\sin\varphi_2 = 0$$

(b) Write the Jacobian of the constraints equations

$$\frac{\partial f_1}{\partial\varphi_1} = -a\sin\varphi_1 = J_{11}$$

$$\frac{\partial f_1}{\partial\varphi_2} = -b\sin\varphi_2 = J_{12}$$

$$\frac{\partial f_1}{\partial\varphi_3} = -1 = J_{13}$$

$$\frac{\partial f_2}{\partial\varphi_1} = a\cos\varphi_1 = J_{21}$$

$$\frac{\partial f_2}{\partial\varphi_2} = b\cos\varphi_2 = J_{22}$$

$$\frac{\partial f_2}{\partial\varphi_3} = 0 = J_{23}$$

then

$$\begin{bmatrix} -a\sin\varphi_1 & -b\sin\varphi_2 & -1 \\ a\cos\varphi_1 & b\cos\varphi_2 & 0 \end{bmatrix} \begin{Bmatrix} \delta\varphi_1 \\ \delta\varphi_2 \\ \delta\varphi_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

and

$$[J_1] = \begin{bmatrix} -a\sin\varphi_1 & -b\sin\varphi_2 \\ a\cos\varphi_1 & b\cos\varphi_2 \end{bmatrix}$$

$$[J_2] = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

Step(3): Write the equilibrium configuration expression

The B matrix is

$$[B] = \begin{bmatrix} [J_1]^{-1} & [J_2] \\ & [I] \end{bmatrix}$$

$$[J_1]^{-1} = \frac{1}{ab\sin(\varphi_2 - \varphi_1)} \begin{bmatrix} b\cos\varphi_2 & b\sin\varphi_2 \\ -a\cos\varphi_1 & a\sin\varphi_1 \end{bmatrix}$$

$$[J_1]^{-1}[J_2] = \frac{1}{ab\sin(\varphi_2 - \varphi_1)} \begin{bmatrix} b\cos\varphi_2 \\ -a\cos\varphi_1 \end{bmatrix}$$

then

$$[B] = \begin{bmatrix} \dfrac{\cos\varphi_2}{a\sin(\varphi_2 - \varphi_1)} \\ \dfrac{-\cos\varphi_1}{b\sin(\varphi_2 - \varphi_1)} \\ 1 \end{bmatrix}$$

Finally, we write the expression

$[B]^T\{C\} = \{0\}$, or

$$\left(m_1 g\frac{a}{2} + m_2 g a\right)\cos\varphi_1 \frac{\cos\varphi_2}{a\sin(\varphi_2 - \varphi_1)} - m_2 g\frac{b}{2}\cos\varphi_2 \frac{\cos\varphi_1}{b\sin(\varphi_2 - \varphi_1)} + (\varphi_{3_o} - \varphi_3)k = 0$$

which can be reduced to

$$\left(\frac{m_1 g}{2} - \frac{m_2 g}{2}\right)\frac{\cos\varphi_1\cos\varphi_2}{\sin(\varphi_2 - \varphi_1)} + (\varphi_{3_o} - \varphi_3)k = 0$$

using this equation with the two constraints equations, we can solve for the equilibrium configuration of the mechanical system.

This problem was modeled and solved by the DAMES program. There is one primary coordinate associated with a spring. This coordinate is φ_3 . The primary freedom related to this primary coordinate is freedom 13. This freedom is in the direction of the axis of the spring. The load generated by this primary coordinate is

$$fpc(1) = k(\varphi_{3_o} - \varphi_3)$$

where k is the spring constant, and φ_{3_o} is the value of φ_3 when the spring is relaxed. This force is introduced to the program in the *#drive* section of the input data, and it is listed in Appendix B.

The input data is listed in Appendix A, with the following values for m_1, m_2, k, and g

$$m_1 = 0.510528 \text{ Kg}, m_1 = 0.510528 \text{ Kg},$$

$$k = 11.4 \text{ N/m}, g = 9.81 \text{ m/sec}^2$$

and the solution was

φ_1=0.510528 rad , $\varphi_2 = 5.95137$ rad , $\varphi_3 = 41.2320$ m

by substituting these values in the equilibrium configuration equations developed above, we can see that they are satisfied.

A plot of the mechanism at the initial guess configuration and the static equilibrium configuration was obtained by the DAMES program and is shown in Figures (40.A) and (40.B).

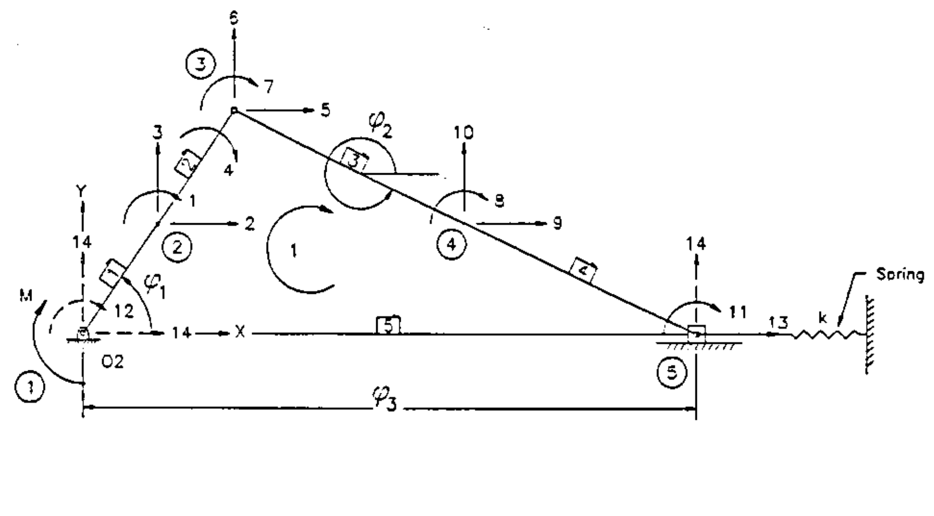

Figure (39) The Finite Element Model of the Slider Crank Mechanism.

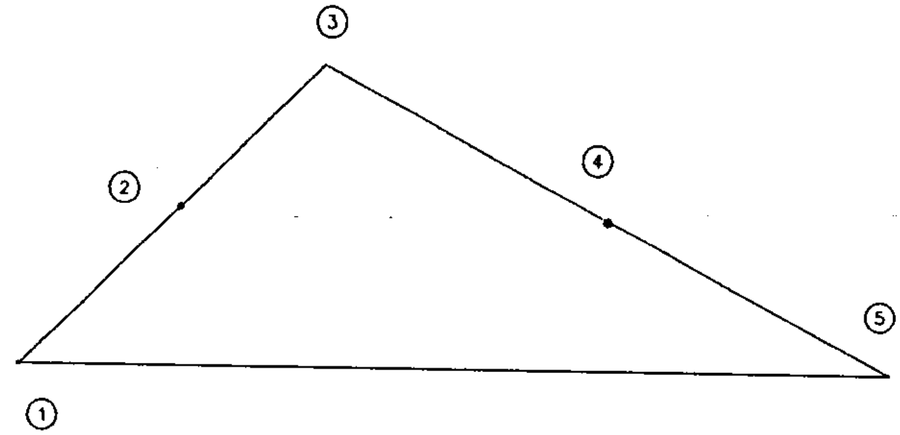

Figure (40.A) The Mechanism Configuration at the Initial Guess.

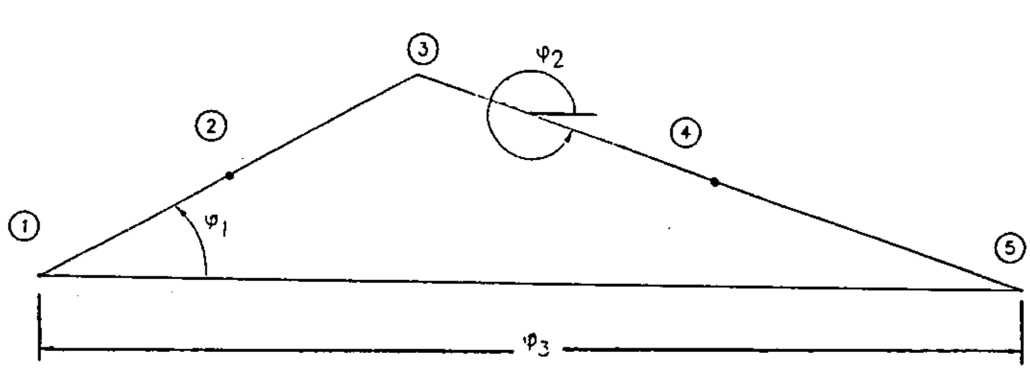

Figure (40.B) The Static Equilibrium Configuration of the Mechanism.

Example 7

A 3R Robot under Gravity Effect

In this example, the robot shown on Figure (41), is allowed to fall under the action of gravity. The following values were given for the masses at the joints,

$m_3 = 0.01$, $m_4 = 0.001$, $m_5 = 0.001$, $m_7 = 0.001$

The input data for this example is taken directly from Figure(41). It is listed in Appendix A. A plot of the robot at the initial configuration and the static equilibrium configuration of the mechanism is shown in Figures (42.A) and (42.B) as obtained by the DAMES program.

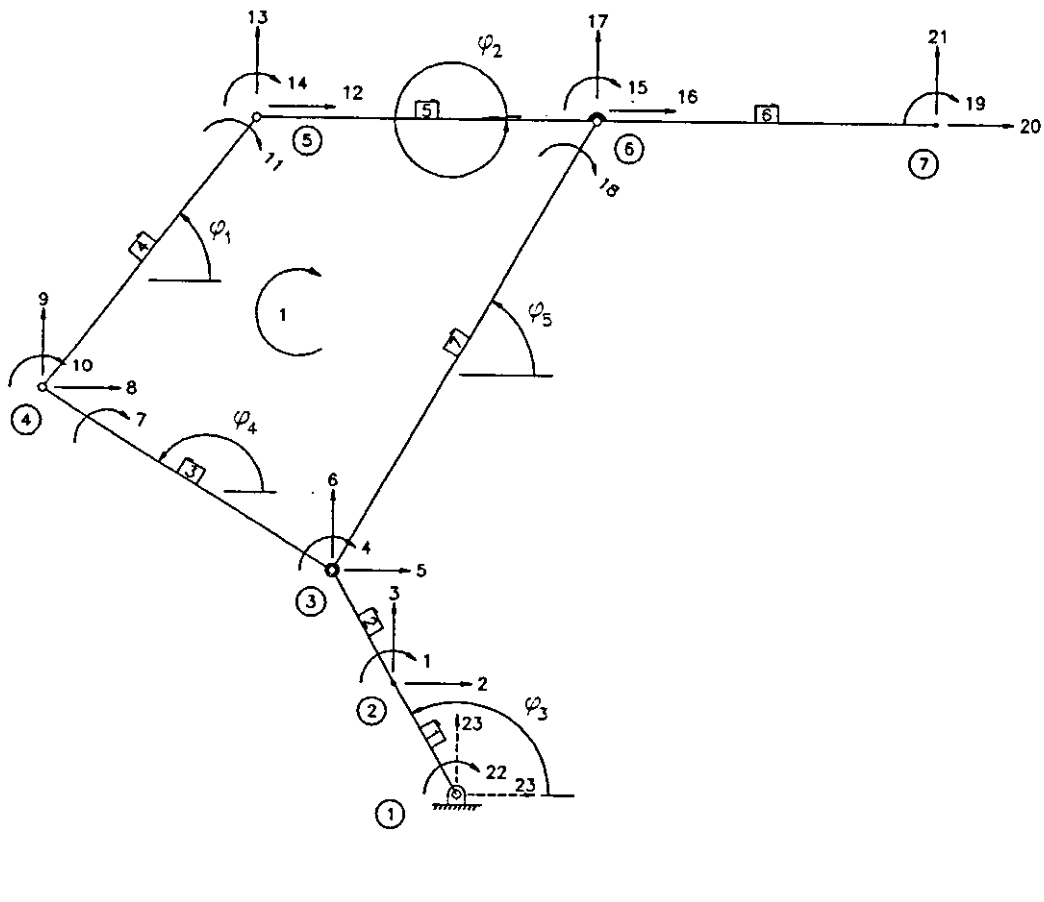

Figure (41) A 3R Robot Under Gravity Effect.

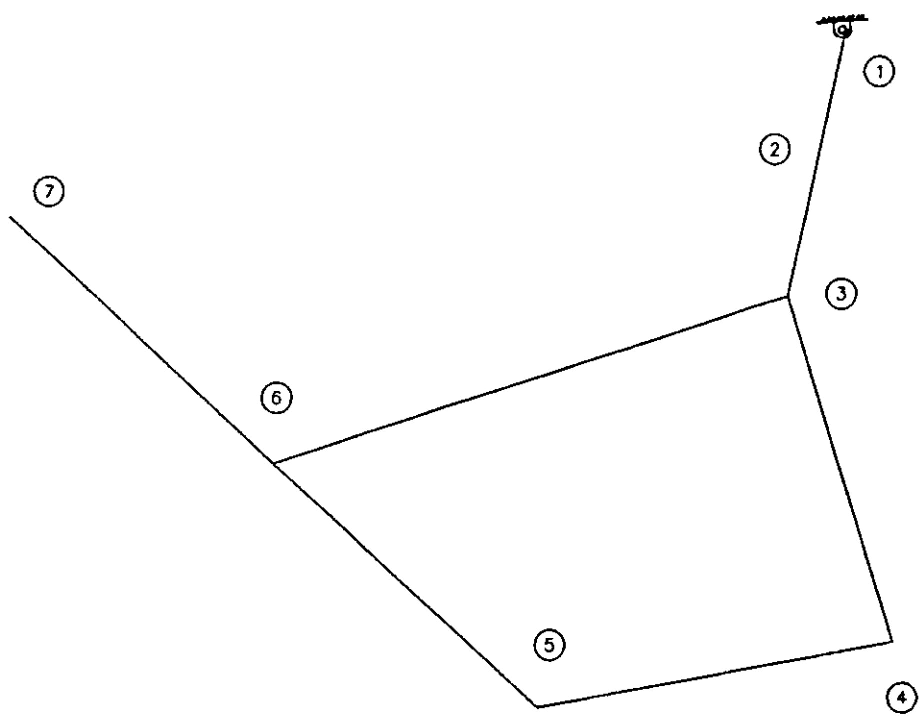

Figure (42.A) The Mechanism Configuration at the Initial Guess.

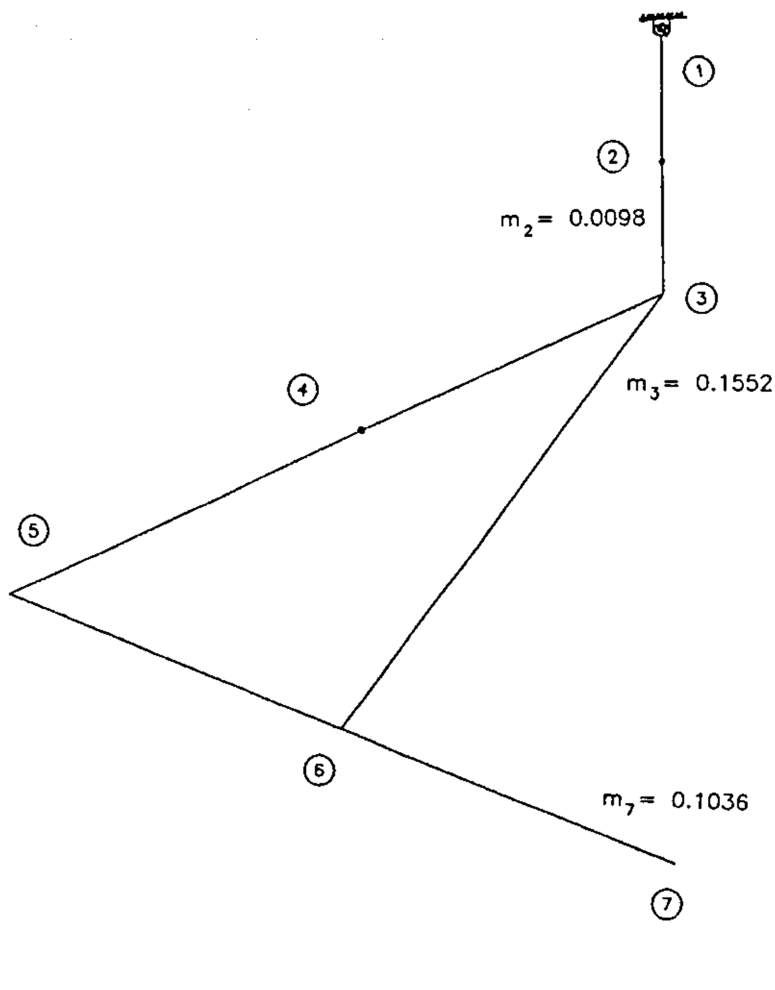

Figure (42.B) The Static Equilibrium Configuration of the Mechanism.

Example 8

An Application to the Human Spine

In this example, the lumbar region of the human spine is modeled in the sagittal plane (vertical section in the direction from back to face). The lumbar region consists of the bottom five vertebrae of the spine. The facet joint, where contact occurs between two vertebrae on the back side, is modeled by a slider moving on an inclined straight path. The two dimensional model is shown in Figure (43). The dimensions of the model were taken from a real spine model [1]. The spine was subjected to two sets of loading conditions. These sets are given in table 5.1. The elastic properties of the bones are not well defined. For this reason, the modulus of elasticity and the cross-sectional properties of the elements were given as unity and the magnitudes of the elastic deflections are neglected. The system has five primary coordinates. The primary forces associated with the primary coordinates are considered as spring forces, each with a spring constant k. Because of the lack of information about this subject, a value is assumed for the spring constant.

The input data for this example is taken directly from Figure (43). It is listed in Appendix A. The primary forces are introduced in the *#drive* section of the input data, and it is listed in Appendix B. The configuration of the spine at the initial guess is shown in Figure (44). The static equilibrium configuration of the mechanism is shown in Figures (45.A) and (45.B) for both loading sets as obtained by the DAMES program.

Table 5.1

The Loading Sets of the Spine

SET No.1		SET No.2	
Freedom No.	Load	Freedom No.	Load
36	10 N	35	5 Nm
		37	-5 N
		39	10 N
		40	-7 N

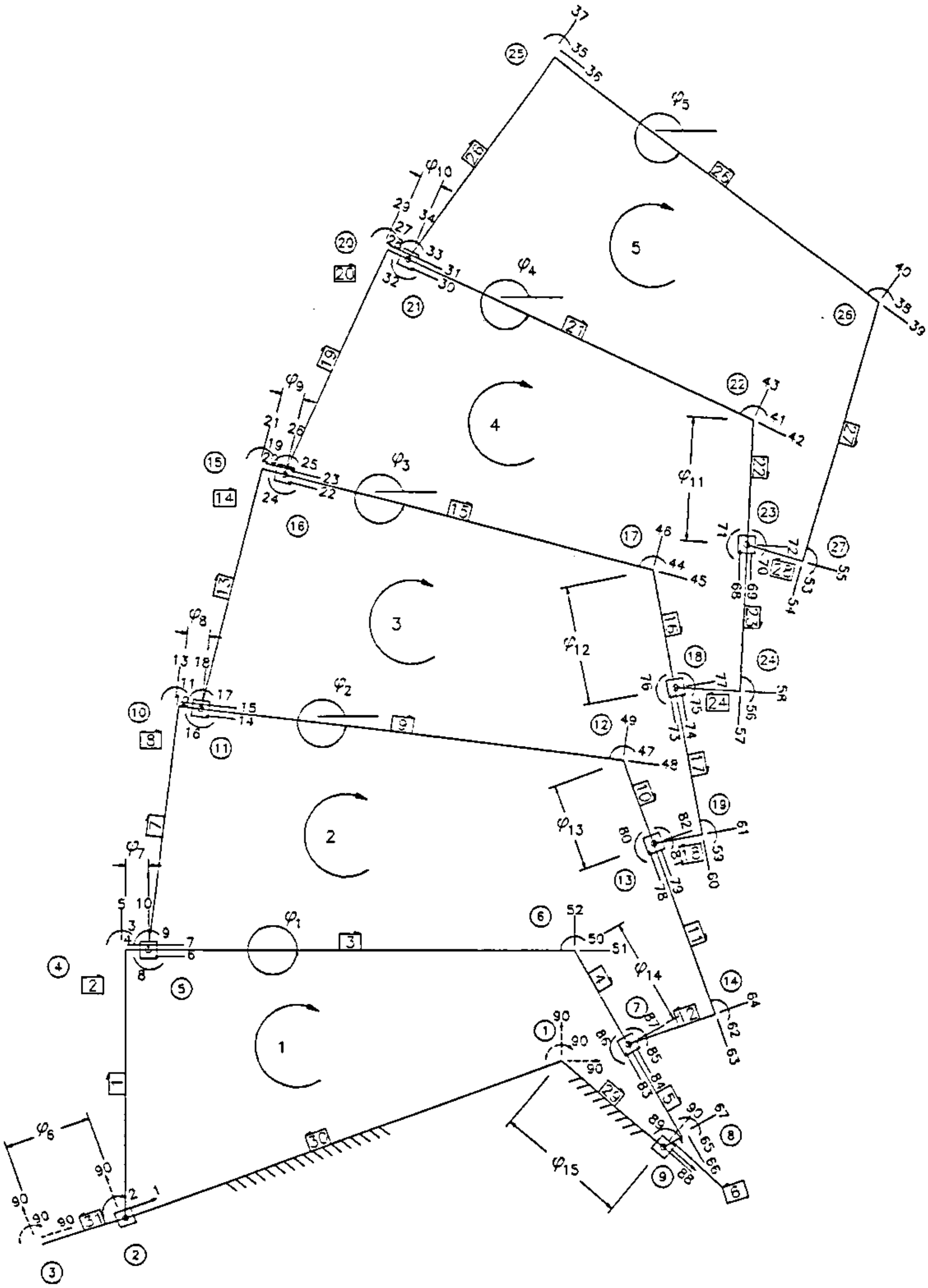

Figure (43) The Finite Element Model of the Lumbar Region of the Human Spine. [1]

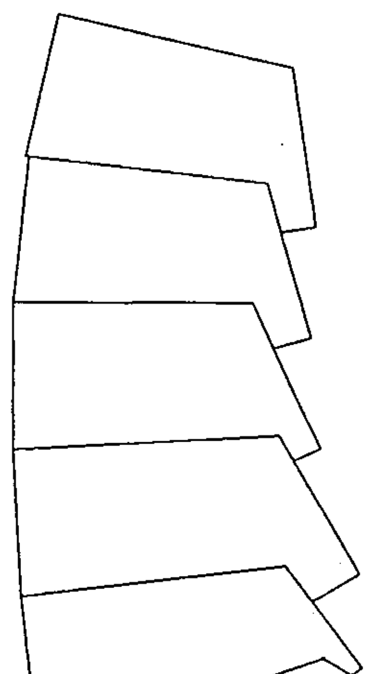

Figure (44) The configuration of the Spine at the Initial Guess.

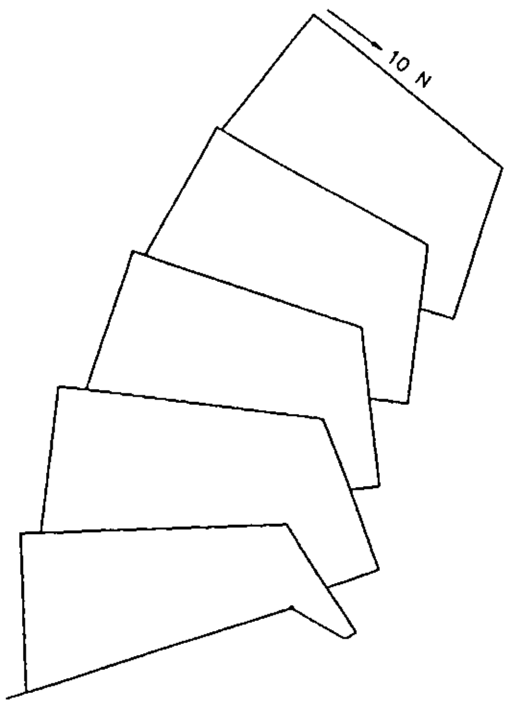

Figure (45.A) The Static Equilibrium Configuration of the Spine for Loading Set No.1.

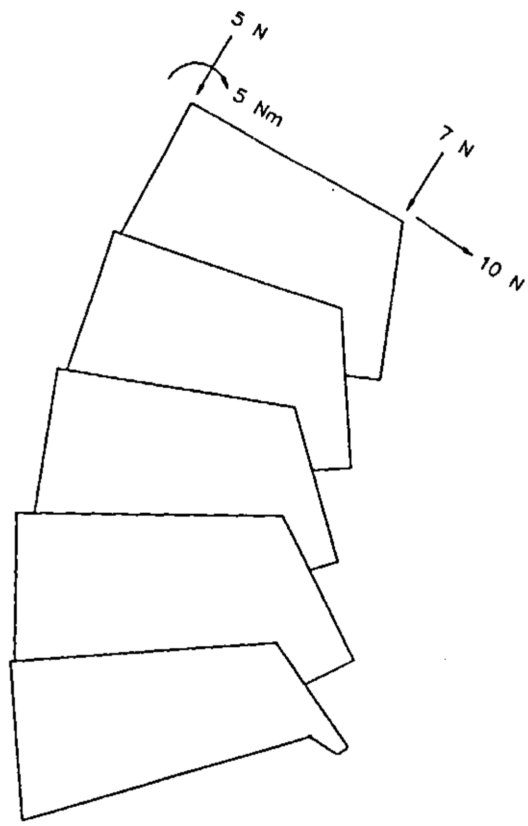

Figure (45.B) The Static Equilibrium Configuration of the Spine for Loading Set No.2.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

Conclusions:

A generalized computer program for dynamic and static analysis of planar mechanisms was developed and presented in this thesis. This computer program is an extension to a previous work done by Dr. Mohammad Dado [1]. This program is a result of the application of theories in four important areas in mechanical design:

1. Theories of kinematic analysis of planar mechanisms.
2. Theories of static and dynamic analysis.
3. Finite element utilization in kineto-elasto-static analysis.
4. Theories of numerical analysis for solving linear and nonlinear simultaneous equations.

This program performs four types of analysis:

1. Kinematic analysis, where the positions, velocities and accelerations of the links and any point on the mechanism are determined.
2. Static analysis of structures, where the deflections at the joints and the element internal forces and moments are determined.
3. Two types of dynamic analysis are performed:
   a. Forward dynamic analysis, where the driving forces and torques are known functions of time and/or coordinates positions and velocities.
   b. Inverse dynamic analysis, where the driving forces and torques are unknown but the motion they generate is known.
4. Static equilibrium position analysis, where the static equilibrium configuration of a mechanism is determined for a given loading condition.

This program features a general modeling technique for planar mechanisms, which is capable of accommodating arbitrary constraints and motion generators.

The program provides the results in plotted or tabulated forms. It also provides a simulation of the motion of the mechanism.

During the development of this program, several examples where solved, either for the purpose of checking the program or for the purpose of illustration.

This program represents a significant design tool for the mechanism designers. They can perform dynamic and static analysis of their designed mechanisms while they are in the design stage. Thus, reducing the time and cost needed to complete their design.

Recommendations:

An important extension of the work presented here is to include analysis of three-dimensional mechanical systems, which would increase the capability and significance of the computer program. Using such a program, a large class of problems can be analyzed in the area of robotics, spatial mechanisms, and dynamics of 3-D suspension systems. This extension involves the development of a 3-D finite line element. And it would be expected to increase the time to enter the data describing a given mechanism. Hence, the development of automatic data generation-schemes are highly recommended. The development of such a program will be a very important addition to the limited number of programs available in the area of three-dimensional analysis.

References:

1. Dado, M. F., Dynamic Analysis of Mechanical Systems, Report, Oklahoma State University, Mechanical and Aero-space Engineering Department, Stillwater, OK., 1984, PP. 1-85.
2. Chace, M. A. and Sheth, P. N., "Adaptation of Computer Techniques to the Design of Mechanical Dynamic Machinery", ASME Paper No. 73-DET-58.
3. Sheth, P. N., and Uicker, J. J., Jr., "IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkage", J. Engr. Idus. Trans., ASME, 94, 1972, PP. 454-646.
4. Amin, A. U., "Automatic Formulation and Solution Techniques in Dynamics of Machinery", Ph.D., Dissertation. University of Pennsylvania, 1979.
5. Bagci, C., and Abounassif, Jamal A.-N., "Computer-Aided Dynamic Force, Stress and Gross-Motion Response Analysis of Planar Mechanisms Using Finite Line Element Technique",ASME Paper No. 82-DET-11, presented at the 17th Mechanisms Conference, Washington, D.C., Sept. 12-15, 1982.
6. Dado, M. F., and Soni, A. H., "A Generalized Approach for Forward and Inverse Dynamics of Elastic Manipulators", IEEE International Conference on Robotics and Automation, April, 1986.
7. Dado, M. F., and Soni, A. H., "Complete Dynamic Analysis of Elastic Linkages", Journal of Mechanisms, Transmission and Automation in Design, ASME paper No. 86-DET-52.
8. D'Souza, A. Frank, and Garg, Vijay K. , Advanced Dynamics - Modeling and Analysis, 1st Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1984., PP. 124-126.
9. Beer, F. P., and Johnston, E. R., Jr., Vector Mechanics for Engineers - Dynamics, Fourth Edition, McGraw-Hill Book Company, Singapore, 1984, PP. 854-856.
10. Thomson, William T., Theory of Vibration with Applications, 2nd Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1981. PP. 163-164.

11. Mabie, H.H., and Ocvirk, F.W., Mechanisms and Dynamics of Machinery, Third Edition, John Wiley & Sons, USA, 1978, PP. 62-67.

Appendix A

The Listing of the Input Model Data for the Eight Examples

MODEL DATA FOR EXAMPLE 1: A SIMPLE PENDULUM.

The characteristic data.

number of elements	=	1
number of loops	=	0
number of lagrangian coordinates	=	1
number of joints	=	2
time increment	=	0.0050
total time for analysis	=	2.5000
analysis index	=	1
total number of freedoms	=	3
gravity value	=	9.8100
modulus of elasticity	=	207000000.000000
number of variable masses	=	0
number of primary coordinates	=	1

The elements data. Each line with the following order:
element#,length,variable coord#,orientation,variable coord#,
initial joint#,terminal joint#,global freedom nos.(N1..N6)
orientation of initial joint freedoms,variable coord#,
orientation of terminal joint freedoms,variable coord#,
cross-sectional area,corss-sectional area moment of inertia

1 1.0000000e+00 0 0.0000000e+00 1 1 2 4 5 5 1 2 3
0.0000000e+00 0 0.0000000e+00 0 1.0000000e+00 1.0000000e+00

Loops data. Each line with following order:
loop #, no. of elements in the loop,
elements numbers in the loop when traced in the CW direction
number of loops = 0 ---> no loop data

The joints data. Each line with the following order:
joint#,mass lumped at the joint,mass moment of inertia,# of elements in the path
elements numbers included in the path
1 0.000000e+00 0.000000e+00 0
2 1.000000e+00 0.000000e+00 1 1

The variable length data. Each line with the following order:
joint# at which the var. mass is lumped,element# at one side,
element# at the other side,linear density of the link material
number of var. masses = 0 -> no data

The primary coordinates data. Each line with the following order:
primary coord#, initial position, initial velocity, global freedom# associated
with the primary coord., number of elements associated with the primary coord.
the elements' numbers.
1 -1.400000e+00 0.000000e+00 4 1 1

Initial conditions for the lagrangian coordinates:
(ti(k) : lagrangian coordinate number k)

MODEL DATA FOR EXAMPLE 2: A DOUBLE PENDULUM.

The characteristic data.

number of elements	=	2
number of loops	=	0
number of lagrangian coordinates	=	2
number of joints	=	3
time increment	=	0.0050
total time for analysis	=	3.0000
analysis index	=	1
total number of freedoms	=	6
gravity value	=	9.8000
modulus of elasticity	=	1.000000
number of variable masses	=	0
number of primary coordinates	=	2

The elements data. Each line with the following order:
element#,length,variable coord#,orientation,variable coord#,
initial joint#,terminal joint#,global freedom nos.(N1..N6)
orientation of initial joint freedoms,variable coord#,
orientation of terminal joint freedoms,variable coord#,
cross-sectional area,corss-sectional area moment of inertia

```
1  1.0000000e+00  0  0.0000000e+00  1  1  2  8  9  9  1  2  3
   0.0000000e+00  0  0.0000000e+00  0  1.0000000e+00  1.0000000e+00
2  1.0000000e+00  0  0.0000000e+00  2  2  3  7  2  3  4  5  6
   0.0000000e+00  0  0.0000000e+00  0  1.0000000e+00  1.0000000e+00
```

Loops data. Each line with following order:
loop #, no. of elements in the loop,
elements numbers in the loop when traced in the CW direction
number of loops = 0 ---> no loop data

The joints data. Each line with the following order:
joint#,mass lumped at the joint,mass moment of inertia,# of elements in the path
elements numbers included in the path

```
1  0.000000e+00  0.000000e+00  0
2  1.000000e+00  0.000000e+00  1  1
3  1.000000e+00  0.000000e+00  2  1  2
```

The variable length data. Each line with the following order:
joint# at which the var. mass is lumped,element# at one side,
element# at the other side,linear density of the link material
number of var. masses = 0 -> no data

The primary coordinates data. Each line with the following order:
primary coord#, initial position, initial velocity, global freedom# associated with the primary coord., number of elements associated with the primary coord. the elements' numbers.

```
1  -1.000000e+00  0.000000e+00  8  1  1
2  -8.000000e-01  0.000000e+00  7  1  2
```

Initial conditions for the lagrangian coordinates:
(ti(k) : lagrangian coordinate number k)

MODEL DATA FOR EXAMPLE 3: AUTOMOBILE ON A ROUGH ROAD.

The characteristic data.

number of elements	=	8
number of loops	=	1
number of lagrangian coordinates	=	7
number of joints	=	8
time increment	=	0.0050
total time for analysis	=	2.0000
analysis index	=	1
total number of freedoms	=	11
gravity value	=	386.0000
modulus of elasticity	=	1.000000
number of variable masses	=	0
number of primary coordinates	=	2

The elements data. Each line with the following order:
element#,length,variable coord#,orientation,variable coord#,
initial joint#,terminal joint#,global freedom nos.(N1..N6)
orientation of initial joint freedoms,variable coord#,
orientation of terminal joint freedoms,variable coord#,
cross-sectional area,corss-sectional area moment of inertia

```
1 0.0000000e+00 1 0.0000000e+00 0 1 2 0 0 0 0 0 0
  0.0000000e+00 0 0.0000000e+00 0 0.0000000e+00 0.0000000e+00
2 0.0000000e+00 2 9.0000000e+01 0 2 3 0 0 0 0 0 0
  0.0000000e+00 0 0.0000000e+00 0 0.0000000e+00 0.0000000e+00
3 0.0000000e+00 6 9.0000000e+01 3 3 4 1 12 12 2 3 7
  0.0000000e+00 0 0.0000000e+00 3 1.0000000e+00 1.0000000e+00
4 5.4000000e+01 0 0.0000000e+00 3 4 5 2 3 7 9 10 11
  0.0000000e+00 3 0.0000000e+00 0 1.0000000e+00 1.0000000e+00
5 5.4000000e+01 0 0.0000000e+00 3 5 6 9 10 11 5 6 8
  0.0000000e+00 0 0.0000000e+00 3 1.0000000e+00 1.0000000e+00
6 0.0000000e+00 4 0.0000000e+00 0 2 7 0 0 0 0 0 0
  0.0000000e+00 0 0.0000000e+00 0 0.0000000e+00 0.0000000e+00
7 0.0000000e+00 5 9.0000000e+01 0 7 8 0 0 0 0 0 0
  0.0000000e+00 0 0.0000000e+00 0 0.0000000e+00 0.0000000e+00
8 0.0000000e+00 7 9.0000000e+01 3 8 6 4 12 12 5 6 8
  0.0000000e+00 0 0.0000000e+00 3 1.0000000e+00 1.0000000e+00
```

Loops data. Each line with following order:
loop #, no. of elements in the loop,
elements numbers in the loop when traced in the CW direction

```
1 7 2 3 4 5 -8 -7 -6
```

The joints data. Each line with the following order:
joint#,mass lumped at the joint,mass moment of inertia,# of elements in the path
elements numbers included in the path

```
1 0.000000e+00 0.000000e+00 0
2 0.000000e+00 0.000000e+00 1 1
3 0.000000e+00 0.000000e+00 2 1 2
4 0.000000e+00 0.000000e+00 3 1 2 3
5 6.476700e+02 1.200000e+04 4 1 2 3 4
6 0.000000e+00 0.000000e+00 4 1 6 7 8
7 0.000000e+00 0.000000e+00 2 1 6
8 0.000000e+00 0.000000e+00 3 1 6 7
```

MODEL DATA FOR EXAMPLE 3 (Continued)

The variable length data. Each line with the following order:
joint# at which the var. mass is lumped,element# at one side,
element# at the other side,linear density of the link material
number of var. masses = 0 -> no data

The primary coordinates data. Each line with the following order:
primary coord#, initial position, initial velocity, global freedom# associated
with the primary coord., number of elements associated with the primary coord.
the elements' numbers.

6	1.615380e+01	0.000000e+00	7	1	4
7	1.615380e+01	0.000000e+00	8	1	5

Initial conditions for the lagrangian coordinates:
(ti(k) : lagrangian coordinate number k)
ti(1)= -1.080000e+02
ti(2)= 0.000000e+00
ti(3)= 0.000000e+00
ti(4)= 1.080000e+02
ti(5)= 0.000000e+00

MODEL DATA FOR EXAMPLE 4: DISK CAM WITH RADIAL FLAT-FACED FOLLOWER

The characteristic data.

number of elements	=	2
number of loops	=	0
number of lagrangian coordinates	=	2
number of joints	=	3
time increment	=	0.0050
total time for analysis	=	1.0000
analysis index	=	1
total number of freedoms	=	1
gravity value	=	9.8100
modulus of elasticity	=	207000000.000000
number of variable masses	=	0
number of primary coordinates	=	0

The elements data. Each line with the following order:
element#,length,variable coord#,orientation,variable coord#,
initial joint#,terminal joint#,global freedom nos.(N1..N6)
orientation of initial joint freedoms,variable coord#,
orientation of terminal joint freedoms,variable coord#,
cross-sectional area,corss-sectional area moment of inertia

```
1  0.0000000e+00  1  0.0000000e+00  0  1  2  2  2  2  2  1  2
   0.0000000e+00  0  0.0000000e+00  0  9.9999998e-03  9.9999998e-03
2  0.0000000e+00  2  0.0000000e+00  0  2  3  2  1  2  2  2  2
   0.0000000e+00  0  0.0000000e+00  0  9.9999998e-03  9.9999998e-03
```

Loops data. Each line with following order:
loop #, no. of elements in the loop,
elements numbers in the loop when traced in the CW direction
number of loops = 0 ---> no loop data

The joints data. Each line with the following order:
joint#,mass lumped at the joint,mass moment of inertia,# of elements in the path
elements numbers included in the path

```
1  0.000000e+00  0.000000e+00  0
2  1.000000e+00  0.000000e+00  1  1
3  0.000000e+00  0.000000e+00  2  1  2
```

The variable length data. Each line with the following order:
joint# at which the var. mass is lumped,element# at one side,
element# at the other side,linear density of the link material
number of var. masses = 0 -> no data

The primary coordinates data. Each line with the following order:
primary coord#, initial position, initial velocity, global freedom# associated
with the primary coord., number of elements associated with the primary coord.
the elements' numbers.
no. of primary coordinates = 0 ---> no data

Initial conditions for the lagrangian coordinates:
(ti(k) : lagrangian coordinate number k)
ti(1)= 3.200000e-02
ti(2)= 1.080000e-01

MODEL DATA FOR EXAMPLE 5: A 3-R MIXED-LOOP PLANAR ROBOT

The characteristic data.

number of elements	=	9
number of loops	=	2
number of lagrangian coordinates	=	6
number of joints	=	8
time increment	=	0.0500
total time for analysis	=	4.0000
analysis index	=	1
total number of freedoms	=	21
gravity value	=	386.0000
modulus of elasticity	=	30000000.000000
number of variable masses	=	0
number of primary coordinates	=	0

The elements data. Each line with the following order:
element#,length,variable coord#,orientation,variable coord#,
initial joint#,terminal joint#,global freedom nos.(N1..N6)
orientation of initial joint freedoms,variable coord#,
orientation of terminal joint freedoms,variable coord#,
cross-sectional area,corss-sectional area moment of inertia

```
1  9.0000000e+00  0  0.0000000e+00  1  1  2  22  22  22  1  2  3
   0.0000000e+00  0  0.0000000e+00  0  7.5000000e-01  1.4060000e-01
2  9.0000000e+00  0  0.0000000e+00  1  2  3  1  2  3  4  5  6
   0.0000000e+00  0  0.0000000e+00  0  7.5000000e-01  1.4060000e-01
3  2.4000000e+01  0  0.0000000e+00  2  3  4  4  5  6  7  8  9
   0.0000000e+00  0  0.0000000e+00  0  3.1999999e-01  2.1299999e-02
4  3.4000000e+01  0  0.0000000e+00  5  4  5  10  8  9  11  12  13
   0.0000000e+00  0  0.0000000e+00  0  3.1999999e-01  2.1299999e-02
5  2.4000000e+01  0  0.0000000e+00  4  5  6  14  12  13  15  16  17
   0.0000000e+00  0  0.0000000e+00  0  7.5000000e-01  1.4060000e-01
6  2.4000000e+01  0  0.0000000e+00  4  6  7  15  16  17  19  20  21
   0.0000000e+00  0  0.0000000e+00  0  7.5000000e-01  1.4060000e-01
7  3.6000000e+01  0  0.0000000e+00  6  3  6  4  5  6  18  16  17
   0.0000000e+00  0  0.0000000e+00  0  3.1999999e-01  2.1299999e-02
8  4.8000000e+01  0  4.5000000e+01  0  1  8  0  0  0  0  0  0
   0.0000000e+00  0  0.0000000e+00  0  0.0000000e+00  0.0000000e+00
9  1.2000000e+01  0  0.0000000e+00  3  8  7  0  0  0  0  0  0
   0.0000000e+00  0  0.0000000e+00  0  0.0000000e+00  0.0000000e+00
```

Loops data. Each line with following order:
loop #, no. of elements in the loop,
elements numbers in the loop when traced in the CW direction

```
1  6  1  2  7  6  -9  -8
2  4  3  4  5  -7
```

The joints data. Each line with the following order:
joint#,mass lumped at the joint,mass moment of inertia,# of elements in the path
elements numbers included in the path

```
1  0.000000e+00  0.000000e+00  0
2  9.800000e-03  2.640000e-01  1  1
3  1.552000e-01  2.800000e+00  2  1  2
4  0.000000e+00  0.000000e+00  3  1  2  3
5  0.000000e+00  0.000000e+00  4  1  2  3  4
```

MODEL DATA FOR EXAMPLE 5 (Continued)

```
6  0.000000e+00  0.000000e+00  5  1  2  3  4  5
7  1.036000e-01  0.000000e+00  4  1  2  7  6
8  0.000000e+00  0.000000e+00  1  8
```

The variable length data. Each line with the following order:
joint# at which the var. mass is lumped,element# at one side,
element# at the other side,linear density of the link material
number of var. masses = 0 -> no data

The primary coordinates data. Each line with the following order:
primary coord#, initial position, initial velocity, global freedom# associated
with the primary coord., number of elements associated with the primary coord.
the elements' numbers.

no. of primary coordinates = 0 ---> no data

Initial conditions for the lagrangian coordinates:
(ti(k) : lagrangian coordinate number k)
ti(1)= 2.095000e+00
ti(2)= 3.141600e+00
ti(3)= 1.570800e+00
ti(4)= 0.000000e+00
ti(5)= 1.047200e+00
ti(6)= 1.047200e+00

MODEL DATA FOR EXAMPLE 6: A SLIDER CRANK MECHANISM

The characteristic data.

number of elements	=	5
number of loops	=	1
number of lagrangian coordinates	=	3
number of joints	=	5
time increment	=	0.0000
total time for analysis	=	0.0000
analysis index	=	3
total number of freedoms	=	12
gravity value	=	9.8100
modulus of elasticity	=	207000000.000000
number of variable masses	=	0
number of primary coordinates	=	1

The elements data. Each line with the following order:
element#,length,variable coord#,orientation,variable coord#,
initial joint#,terminal joint#,global freedom nos.(N1..N6)
orientation of initial joint freedoms,variable coord#,
orientation of terminal joint freedoms,variable coord#,
cross-sectional area,corss-sectional area moment of inertia

```
1  9.0000000e+00  0  0.0000000e+00  1  1  2  13  13  13  1  2  3
   0.0000000e+00  0  0.0000000e+00  0  9.9999998e-03  9.9999998e-03
2  9.0000000e+00  0  0.0000000e+00  1  2  3  1  2  3  4  5  6
   0.0000000e+00  0  0.0000000e+00  0  9.9999998e-03  9.9999998e-03
3  1.3500000e+01  0  0.0000000e+00  2  3  4  7  5  6  8  9  10
   0.0000000e+00  0  0.0000000e+00  0  9.9999998e-03  9.9999998e-03
4  1.3500000e+01  0  0.0000000e+00  2  4  5  8  9  10  11  12  13
   0.0000000e+00  0  0.0000000e+00  0  9.9999998e-03  9.9999998e-03
5  0.0000000e+00  3  0.0000000e+00  0  1  5  0  0  0  0  0  0
   0.0000000e+00  0  0.0000000e+00  0  0.0000000e+00  0.0000000e+00
```

Loops data. Each line with following order:
loop #, no. of elements in the loop,
elements numbers in the loop when traced in the CW direction

```
1  5  1  2  3  4  -5
```

The joints data. Each line with the following order:
joint#,mass lumped at the joint,mass moment of inertia,# of elements in the path
elements numbers included in the path

```
1  0.000000e+00  0.000000e+00  0
2  5.000000e+00  3.000000e+01  1  1
3  0.000000e+00  0.000000e+00  2  1  2
4  6.000000e+00  4.500000e+01  3  1  2  3
5  0.000000e+00  0.000000e+00  1  5
```

The variable length data. Each line with the following order:
joint# at which the var. mass is lumped,element# at one side,
element# at the other side,linear density of the link material
number of var. masses = 0 -> no data

The primary coordinates data. Each line with the following order:
primary coord#, initial position, initial velocity, global freedom# associated with the primary coord., number of elements associated with the primary coord. the elements' numbers.

```
3  3.653970e+01  0.000000e+00  12  1  4
```

MODEL DATA FOR EXAMPLE 6 (Continued)

Initial conditions for the lagrangian coordinates:
(ti(k) : lagrangian coordinate number k)
ti(1)= 7.854000e-01
ti(2)= 5.794500e+00

MODEL DATA FOR EXAMPLE 7: A 3R ROBOT UNDER GRAVITY EFFECT

The characteristic data.

number of elements	=	7
number of loops	=	1
number of lagrangian coordinates	=	5
number of joints	=	7
time increment	=	0.0500
total time for analysis	=	4.0000
analysis index	=	3
total number of freedoms	=	22
gravity value	=	386.0000
modulus of elasticity	=	30000000.000000
number of variable masses	=	0
number of primary coordinates	=	3

The elements data. Each line with the following order:
element#,length,variable coord#,orientation,variable coord#,
initial joint#,terminal joint#,global freedom nos.(N1..N6)
orientation of initial joint freedoms,variable coord#,
orientation of terminal joint freedoms,variable coord#,
cross-sectional area,corss-sectional area moment of inertia

```
 1 9.0000000e+00  0 0.0000000e+00  3  1  2 22 23 23  1  2  3
   0.0000000e+00  0 0.0000000e+00  0 7.5000000e-01 1.4060000e-01
 2 9.0000000e+00  0 0.0000000e+00  3  2  3  1  2  3  4  5  6
   0.0000000e+00  0 0.0000000e+00  0 7.5000000e-01 1.4060000e-01
 3 2.4000000e+01  0 0.0000000e+00  4  3  4  4  5  6  7  8  9
   0.0000000e+00  0 0.0000000e+00  0 3.1999999e-01 2.1299999e-02
 4 2.4000000e+01  0 0.0000000e+00  1  4  5 10  8  9 11 12 13
   0.0000000e+00  0 0.0000000e+00  0 3.1999999e-01 2.1299999e-02
 5 2.4000000e+01  0 0.0000000e+00  2  5  6 14 12 13 15 16 17
   0.0000000e+00  0 0.0000000e+00  0 7.5000000e-01 1.4060000e-01
 6 2.4000000e+01  0 0.0000000e+00  2  6  7 15 16 17 19 20 21
   0.0000000e+00  0 0.0000000e+00  0 7.5000000e-01 1.4060000e-01
 7 3.6000000e+01  0 0.0000000e+00  5  3  6  4  5  6 18 16 17
   0.0000000e+00  0 0.0000000e+00  0 3.1999999e-01 2.1299999e-02
```

Loops data. Each line with following order:
loop #, no. of elements in the loop,
elements numbers in the loop when traced in the CW direction

```
 1  4  3  4  5 -7
```

The joints data. Each line with the following order:
joint#,mass lumped at the joint,mass moment of inertia,# of elements in the path
elements numbers included in the path

```
 1  0.000000e+00  0.000000e+00  0
 2  9.800000e-03  2.640000e-01  1  1
 3  1.552000e-01  2.800000e+00  2  1  2
 4  0.000000e+00  0.000000e+00  3  1  2  3
 5  0.000000e+00  0.000000e+00  4  1  2  3  4
 6  0.000000e+00  0.000000e+00  5  1  2  3  4  5
 7  1.036000e-01  0.000000e+00  4  1  2  7  6
```

The variable length data. Each line with the following order:
joint# at which the var. mass is lumped,element# at one side,
element# at the other side,linear density of the link material
number of var. masses = 0 -> no data

MODEL DATA FOR EXAMPLE 7 (Continued)

The primary coordinates data. Each line with the following order:
primary coord#, initial position, initial velocity, global freedom# associated with the primary coord., number of elements associated with the primary coord. the elements' numbers.

```
3  4.512400e+00  0.000000e+00  0  2  1  2
4  5.017400e+00  0.000000e+00  4  1  3
5  3.464600e+00  0.000000e+00  4  1  7
```

Initial conditions for the lagrangian coordinates:
(ti(k) : lagrangian coordinate number k)
ti(1)= 3.517400e+00
ti(2)= 2.617400e+00

MODEL DATA FOR EXAMPLE 8: AN APPLICATION TO THE HUMAN SPINE

The characteristic data.

number of elements	=	31
number of loops	=	5
number of lagrangian coordinates	=	15
number of joints	=	27
time increment	=	1.0000
total time for analysis	=	50.0000
analysis index	=	3
total number of freedoms	=	90
gravity value	=	386.0000
modulus of elasticity	=	30000000.000000
number of variable masses	=	0
number of primary coordinates	=	5

The elements data. Each line with the following order:
element#,length,variable coord#,orientation,variable coord#,
initial joint#,terminal joint#,global freedom nos.(N1..N6)
orientation of initial joint freedoms,variable coord#,
orientation of terminal joint freedoms,variable coord#,
cross-sectional area,corss-sectional area moment of inertia

```
1  1.2000000e+00  0 9.0000000e+01  1  2  4  2  1 90  3  4  5
   1.9194401e+01  0 0.0000000e+00  1 1.0000000e+00 1.0000000e+00
2  0.0000000e+00  7 0.0000000e+00  1  4  5  3  4  5  8  6 10
   0.0000000e+00  1 0.0000000e+00  1 1.0000000e+00 1.0000000e+00
3  2.0000000e+00 -7 0.0000000e+00  1  5  6  8  6 10 50 51 52
   0.0000000e+00  1 0.0000000e+00  1 1.0000000e+00 1.0000000e+00
4  0.0000000e+00 14 -6.0000000e+01  1  6  7 50 51 52 86 83 87
   0.0000000e+00  1 -6.0000000e+01  1 1.0000000e+00 1.0000000e+00
5  9.5999998e-01 -14 -6.0000000e+01  1  7  8 86 83 87 65 66 67
   -6.0000000e+01  1 -6.0000000e+01  1 1.0000000e+00 1.0000000e+00
6  9.4999999e-02  0 -1.5000000e+02  1  8  9 65 66 67 89 88 90
   -6.0000000e+01  1 -3.0000000e+01  0 1.0000000e+00 1.0000000e+00
7  1.1000000e+00  0 9.0000000e+01  2  5 10  9  7 10 11 12 13
   0.0000000e+00  1 0.0000000e+00  2 1.0000000e+00 1.0000000e+00
8  0.0000000e+00  8 0.0000000e+00  2 10 11 11 12 13 16 14 18
   0.0000000e+00  2 0.0000000e+00  2 1.0000000e+00 1.0000000e+00
9  2.0000000e+00 -8 0.0000000e+00  2 11 12 16 14 18 47 48 49
   0.0000000e+00  2 0.0000000e+00  2 1.0000000e+00 1.0000000e+00
10 0.0000000e+00 13 2.9700500e+02  2 12 13 47 48 49 80 78 82
   0.0000000e+00  2 2.9700500e+02  2 1.0000000e+00 1.0000000e+00
11 1.2000000e+00 -13 2.9700500e+02  2 13 14 80 78 82 62 63 64
   2.9700500e+02  2 2.9700500e+02  2 1.0000000e+00 1.0000000e+00
12 4.1000000e-01  0 2.0700500e+02  2 14  7 62 63 64 85 84 87
   2.9700500e+02  2 -6.0000000e+01  1 1.0000000e+00 1.0000000e+00
13 1.1000000e+00  0 9.0000000e+01  3 11 15 17 15 18 19 20 21
   0.0000000e+00  2 0.0000000e+00  3 1.0000000e+00 1.0000000e+00
14 0.0000000e+00  9 0.0000000e+00  3 15 16 19 20 21 24 22 26
   0.0000000e+00  3 0.0000000e+00  3 1.0000000e+00 1.0000000e+00
15 1.8000000e+00 -9 0.0000000e+00  3 16 17 24 22 26 44 45 46
   0.0000000e+00  3 0.0000000e+00  3 1.0000000e+00 1.0000000e+00
16 0.0000000e+00 12 2.9500000e+02  3 17 18 44 45 46 76 73 77
   0.0000000e+00  3 2.9500000e+02  3 1.0000000e+00 1.0000000e+00
17 1.2000000e+00 -12 2.9500000e+02  3 18 19 76 73 77 59 60 61
   2.9500000e+02  3 2.9500000e+02  3 1.0000000e+00 1.0000000e+00
```

MODEL DATA FOR EXAMPLE 8 (Continued)

```
18  2.2000000e-01  0 2.0500000e+02  3 19 13 59 60 61 81 79 82
    2.9500000e+02  3 2.9700500e+02  2 1.0000000e+00 1.0000000e+00
19  1.1000000e+00  0 9.0000000e+01  4 16 20 25 23 26 27 28 29
    0.0000000e+00  3 0.0000000e+00  4 1.0000000e+00 1.0000000e+00
20  0.0000000e+00 10 0.0000000e+00  4 20 21 27 28 29 32 30 34
    0.0000000e+00  4 0.0000000e+00  4 1.0000000e+00 1.0000000e+00
21  1.8000000e+00 -10 0.0000000e+00  4 21 22 32 30 34 41 42 43
    0.0000000e+00  4 0.0000000e+00  4 1.0000000e+00 1.0000000e+00
22  0.0000000e+00 11 2.9200000e+02  4 22 23 41 42 43 71 68 72
    0.0000000e+00  4 2.8048001e+02  4 1.0000000e+00 1.0000000e+00
23  1.2000000e+00 -11 2.9200000e+02  4 23 24 71 68 72 56 57 58
    2.8048001e+02  4 2.8048001e+02  4 1.0000000e+00 1.0000000e+00
24  2.8999999e-01  0 2.0200000e+02  4 24 18 56 57 58 75 74 77
    2.8048001e+02  4 2.9500000e+02  3 1.0000000e+00 1.0000000e+00
25  1.1000000e+00  0 9.0000000e+01  5 21 25 33 31 34 35 36 37
    0.0000000e+00  4 0.0000000e+00  5 1.0000000e+00 1.0000000e+00
26  1.8000000e+00  0 0.0000000e+00  5 25 26 35 36 37 38 39 40
    0.0000000e+00  5 0.0000000e+00  5 1.0000000e+00 1.0000000e+00
27  1.2000000e+00  0 2.9100000e+02  5 26 27 38 39 40 53 54 55
    0.0000000e+00  5 2.6512000e+02  5 1.0000000e+00 1.0000000e+00
28  2.5999999e-01  0 2.0100000e+02  5 27 23 53 54 55 70 69 72
    2.6500000e+02  5 2.8048001e+02  4 1.0000000e+00 1.0000000e+00
29  0.0000000e+00 15 3.3000000e+02  0  1  9 90 90 90 90 88 90
    0.0000000e+00  0 -3.0000000e+01 0 0.0000000e+00 0.0000000e+00
30  2.2483001e+00 -6 1.9919440e+02  0  1  2 90 90 90 90  1 90
    0.0000000e+00  0 1.9194401e+01  0 0.0000000e+00 0.0000000e+00
31  0.0000000e+00  6 1.9919440e+02  0  2  3 90  1 90 90 90 90
    1.9194401e+01  0 1.9194401e+01  0 0.0000000e+00 0.0000000e+00
```

Loops data. Each line with following order:
loop #, no. of elements in the loop,
elements numbers in the loop when traced in the CW direction

```
1  8 30  1  2  3  4  5  6 -29
2  8  7  8  9 10 11 12 -4  -3
3  8 13 14 15 16 17 18 -10 -9
4  8 19 20 21 22 23 24 -16 -15
5  6 25 26 27 28 -22 -21
```

The joints data. Each line with the following order:
joint#,mass lumped at the joint,mass moment of inertia,# of elements in the path
elements numbers included in the path

```
 1  0.000000e+00  0.000000e+00  0
 2  0.000000e+00  0.000000e+00  1 30
 3  0.000000e+00  0.000000e+00  2 30 31
 4  0.000000e+00  0.000000e+00  2 30  1
 5  0.000000e+00  0.000000e+00  3 30  1  2
 6  0.000000e+00  0.000000e+00  4 29 -6 -5 -4
 7  0.000000e+00  0.000000e+00  3 29 -6 -5
 8  0.000000e+00  0.000000e+00  2 29 -6
 9  0.000000e+00  0.000000e+00  1 29
10  0.000000e+00  0.000000e+00  4 30  1  2  7
11  0.000000e+00  0.000000e+00  5 30  1  2  7  8
12  0.000000e+00  0.000000e+00  6 30  1  2  7  8  9
```

MODEL DATA FOR EXAMPLE 8 (Continued)

```
13  0.000000e+00  0.000000e+00  5 29 -6 -5 -12 -11
14  0.000000e+00  0.000000e+00  4 29 -6 -5 -12
15  0.000000e+00  0.000000e+00  6 30  1  2  7  8 13
16  0.000000e+00  0.000000e+00  7 30  1  2  7  8 13 14
17  0.000000e+00  0.000000e+00  8 30  1  2  7  8 13 14 15
18  0.000000e+00  0.000000e+00  7 29 -6 -5 -12 -11 -18 -17
19  0.000000e+00  0.000000e+00  6 29 -6 -5 -12 -11 -18
20  0.000000e+00  0.000000e+00  8 30  1  2  7  8 13 14 19
21  0.000000e+00  0.000000e+00  9 30  1  2  7  8 13 14 19 20
22  0.000000e+00  0.000000e+00 10 30  1  2  7  8 13 14 19 20 21
23  0.000000e+00  0.000000e+00  9 29 -6 -5 -12 -11 -18 -17 -24 -23
24  0.000000e+00  0.000000e+00  8 29 -6 -5 -12 -11 -18 -17 -24
25  0.000000e+00  0.000000e+00 10 30  1  2  7  8 13 14 19 20 25
26  0.000000e+00  0.000000e+00 11 30  1  2  7  8 13 14 19 20 25 26
27  0.000000e+00  0.000000e+00 12 30  1  2  7  8 13 14 19 20 25 26 27
```

The variable length data. Each line with the following order:
joint# at which the var. mass is lumped,element# at one side,
element# at the other side,linear density of the link material
number of var. masses = 0 -> no data

The primary coordinates data. Each line with the following order:
primary coord#, initial position, initial velocity, global freedom# associated
with the primary coord., number of elements associated with the primary coord.
the elements' numbers.

```
11  3.700000e-01  0.000000e+00  69  1  28
12  3.700000e-01  0.000000e+00  74  1  24
13  2.200000e-01  0.000000e+00  79  1  18
14  3.320000e-01  0.000000e+00  84  1  12
15  2.500000e-01  0.000000e+00  88  1  6
```

Initial conditions for the lagrangian coordinates:
(ti(k) : lagrangian coordinate number k)

```
ti(1)=  1.236000e-01
ti(2)=  6.270000e-02
ti(3)=  7.700000e-03
ti(4)=  6.182700e+00
ti(5)=  6.066100e+00
ti(6)=  0.000000e+00
ti(7)=  0.000000e+00
ti(8)=  0.000000e+00
ti(9)=  0.000000e+00
ti(10)=  0.000000e+00
```

Appendix B
The Listing of the Input Data for the Constraints, External Loads, and Driving Forces & Torques for the Eight Examples

INPUT DATA FOR EXAMPLE 1: A SIMPLE PENDULUM

1. THE SIMPLE PENDULUM WITH FREE OSCILLATIONS

```
#uscons
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=0
nsc=0
;   tc(i)=Time dependent motion generator.
;   ptc(i,j)=It's partial derivative w.r.t. coord. j
;   pttc(i)=It's partial derivative w.r.t. time
;   dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;   dpttc(i)=The derivative of pttc(i) w.r.t. time
;   sc(i)=Path & Length constraint
;   psc(i,j)=It's partial derivative w.r.t coord. j
;   dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;   The value of nst is passed from the Analysis module to
;   form the acceleration equations
if(nst=2)
goto 10
endif
;   enter code here, for tc,ptc,pttc,sc & psc
;   & with conditions
goto 20
10:
;   enter code here for dptc &/OR dpttc
20:
return
#end
#usload
;   enter code here for
;   usl(i)=Externally applied load in direction of
;          freedom no. i
return
#end
#drive
;   enter code here for
;   fpc(i)=Driving force or torque
;      i=index corresponds the order by which the input data  for a
;        given driving load is given
; adding a damping force with C = 0 ---> free oscillations
fpc(1)=0
return
#end
```

2. THE SIMPLE PENDULUM WITH DAMPED OSCILLATIONS (C = 0.5 Cc)

```
#uscons
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=0
nsc=0
;   tc(i)=Time dependent motion generator.
;   ptc(i,j)=It's partial derivative w.r.t. coord. j
;   pttc(i)=It's partial derivative w.r.t. time
;   dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;   dpttc(i)=The derivative of pttc(i) w.r.t. time
```

```
;    sc(i)=Path & Length constraint
;    psc(i,j)=It's partial derivative w.r.t coord. j
;    dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;    The value of nst is passed from the Analysis module to
;    form the acceleration equations
if(nst=2)
goto 10
endif
;    enter code here, for tc,ptc,pttc,sc & psc
;    & with conditions
goto 20
10:
;   enter code here for dptc &/OR dpttc
20:
return
#end
#usload
;    enter code here for
;    usl(i)=Externally applied load in direction of
;          freedom no. i
return
#end
#drive
;    enter code here for
;    fpc(i)=Driving force or torque
;      i=index corresponds the order by which the input data  for a
;        given driving load is given
; adding a damping force with C = 0.5 Cc = 0.5*6.264184
fpc(1)=0.5*6.264184*tiv(1)
return
#end
```

3. THE SIMPLE PENDULUM WITH DAMPED OSCILLATIONS (C = Cc)

```
#uscons
;    ntc=Number of time dependent constraints.
;    nsc=Number of spatial constraints.
ntc=0
nsc=0
;    tc(i)=Time dependent motion generator.
;    ptc(i,j)=It's partial derivative w.r.t. coord. j
;    pttc(i)=It's partial derivative w.r.t. time
;    dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;    dpttc(i)=The derivative of pttc(i) w.r.t. time
;    sc(i)=Path & Length constraint
;    psc(i,j)=It's partial derivative w.r.t coord. j
;    dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;    The value of nst is passed from the Analysis module to
;    form the acceleration equations
if(nst=2)
goto 10
endif
;    enter code here, for tc,ptc,pttc,sc & psc
;    & with conditions
goto 20
10:
;   enter code here for dptc &/OR dpttc
20:
```

```
return
#end
#usload
;    enter code here for
;    usl(i)=Externally applied load in direction of
;           freedom no. i
return
#end
#drive
;    enter code here for
;    fpc(i)=Driving force or torque
;        i=index corresponds the order by which the input data  for a
;          given driving load is given
; adding a damping force with C = Cc = 6.264184
fpc(1)=6.264184*tiv(1)
return
#end
```

INPUT DATA FOR EXAMPLE 2: A DOUBLE PENDULUM.

```
#uscons
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=0
nsc=0
;   tc(i)=Time dependent motion generator.
;   ptc(i,j)=It's partial derivative w.r.t. coord. j
;   pttc(i)=It's partial derivative w.r.t. time
;   dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;   dpttc(i)=The derivative of pttc(i) w.r.t. time
;   sc(i)=Path & Length constraint
;   psc(i,j)=It's partial derivative w.r.t coord. j
;   dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;   The value of nst is passed from the Analysis module to
;   form the acceleration equations
if(nst=2)
goto 10
endif
;   enter code here, for tc,ptc,pttc,sc & psc
;   & with conditions
goto 20
10:
;  enter code here for dptc &/OR dpttc
20:
return
#end
#usload
;   enter code here for
;   usl(i)=Externally applied load in direction of
;          freedom no. i
return
#end
#drive
;   enter code here for
;   fpc(i)=Driving force or torque
;      i=index corresponds the order by which the input data  for a
;        given driving load is given
fpc(1)=0
fpc(2)=0
return
#end
```

INPUT DATA FOR EXAMPLE 3: AUTOMOBILE ON A ROUGH ROAD.

```
#uscons
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=3
nsc=0
;   tc(i)=Time dependent motion generator.
;   ptc(i,j)=It's partial derivative w.r.t. coord. j
;   pttc(i)=It's partial derivative w.r.t. time
;   dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;   dpttc(i)=The derivative of pttc(i) w.r.t. time
;   sc(i)=Path & Length constraint
;   psc(i,j)=It's partial derivative w.r.t coord. j
;   dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;   The value of nst is passed from the Analysis module to
;   form the acceleration equations
if(nst=2)
goto 10
endif
;   enter code here, for tc,ptc,pttc,sc & psc
;   & with conditions
tc(1)=550.*tm-108.-ti(1)
ptc(1,1)=-1.
pttc(1)=550.
if(ti(1)<0)
tc(2)=-ti(2)
ptc(2,2)=-1.
else
if(ti(1)>=0)
tc(2)=6.*(1-cos(2*pi/360*ti(1)))-ti(2)
ptc(2,1)=6.*2*pi/360*sin(2*pi/360*ti(1))
ptc(2,2)=-1.
endif
tc(3)=6.*(1.-cos(2*pi/360*(ti(1)+ti(4))))-ti(5)
ptc(3,1)=6.*2*pi/360*sin(2*pi/360*(ti(1)+ti(4)))
ptc(3,4)=6.*2*pi/360*sin(2*pi/360*(ti(1)+ti(4)))
ptc(3,5)=-1.
10:
;  enter code here for dptc &/OR dpttc
if(ti(1)<0)
dptc(2,2)=0.
else
if(ti(1)>=0)
dptc(2,1)=6*power(2*pi/360,2)*tiv(1)*cos(2*pi/360*ti(1))
dptc(2,2)=0.
endif
dptc(3,1)=6*power(2*pi/360,2)*(tiv(1)+tiv(4))*cos(2*pi/360*(ti(1)+ti(4)))
dptc(3,4)=6*power(2*pi/360,2)*(tiv(1)+tiv(4))*cos(2*pi/360*(ti(1)+ti(4)))
dptc(3,5)=0.
return
#end
#usload
;   enter code here for
;   usl(i)=Externally applied load in direction of
;          freedom no. i
return
#end
```

```
#drive
;    enter code here for
;    fpc(i)=Driving force or torque
;        i=index corresponds the order by which the input data  for a
;          given driving load is given
fpc(1)=32500.*(20.-ti(6))-1000.*tiv(6)
fpc(2)=32500.*(20.-ti(7))-1000.*tiv(7)
return
#end
```

INPUT DATA FOR EXAMPLE 4: DISK CAM WITH RADIAL FLAT-FACED FOLLOWER

```
#uscons
; Disk Cam with Radial Flat-Faced Follower
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=2
nsc=0
;   tc(i)=Time dependent motion generator.
;   ptc(i,j)=It's partial derivative w.r.t. coord. j
;   pttc(i)=It's partial derivative w.r.t. time
;   dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;   dpttc(i)=The derivative of pttc(i) w.r.t. time
;   sc(i)=Path & Length constraint
;   psc(i,j)=It's partial derivative w.r.t coord. j
;   dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;   The value of nst is passed from the Analysis module to
;   form the acceleration equations
if(nst=2)
goto 10
endif
;   enter code here, for tc,ptc,pttc,sc & psc
;   & with conditions
if((tm>=0) & (tm<=.25))
tc(1)=.032-ti(1)
ptc(1,1)=-1.0
pttc(1)=0.0
else
if((tm>.25)&(tm<.5))
tc(1)=.0762/pi*(tm*2*pi*1.0-pi/2)-.0762/(4*pi)*sin(4*tm*2*pi*1-2*pi)-ti(1)+.032
ptc(1,1)=-1.0
pttc(1)=.0762/pi*2*pi*1.0-
.0762/(4*pi)*(cos(4*tm*2*pi*1)*4*2*pi*1.0*cos(2*pi)+sin(2*pi)*cos(4*tm*2*pi*1)*4*2*pi*1)
else
if((tm>=.5) & (tm<=.75))
tc(1)=.0381-ti(1)+.032
ptc(1,1)=-1.0
pttc(1)=0.0
else
if((tm>.75) & (tm<=1.0))
tc(1)=.0381*(1.0-(tm*2*pi-3*pi/2)/(pi/2)+1/(2*pi)*sin(4*2*pi*tm-6*pi))-ti(1)+.032
ptc(1,1)=-1.0
pttc(1)=.0381*(2*pi+1/(2*pi)*(cos(4*2*pi*tm)*4*2*pi*cos(6*pi)+sin(6*pi)*cos(4*2*pi*tm)*4*2*pi))
endif
tc(2)=.140-ti(1)-ti(2)
ptc(2,1)=-1
ptc(2,2)=-1
pttc(2)=0
goto 20
10:
;   enter code here for dptc &/OR dpttc
if((tm>=0) & (tm<=.25))
dpttc(1)=0
endif
if((tm>.25) & (tm<.5))
dpttc(1)=.0762/(4*pi)*(-1*sin(4*tm*2*pi*1)*4*2*pi*1.0*4*2*pi*cos(2*pi)-
sin(2*pi)*sin(4*tm*2*pi*1)*4*2*pi*1*4*2*pi*1)
endif
```

```
if((tm>=.5) & (tm<=.75))
dpttc(1)=0.0
endif
if((tm>.75) & (tm<=1.0))
dpttc(1)=.0381*1/(2*pi)*(-1*sin(4*2*pi*tm)*4*2*pi*4*2*pi*cos(6*pi)-
sin(6*pi)*sin(4*2*pi*tm)*4*2*pi*4*2*pi)
endif
dpttc(2)=0
20:
return
#end
#usload
;   enter code here for
;   usl(i)=Externally applied load in direction of
;          freedom no. i
usl(1)=-500*(ti(2)-.108)
return
#end
#drive
;   enter code here for
;   fpc(i)=Driving force or torque
;      i=index corresponds the order by which the input data  for a
;        given driving load is given
return
#end
```

INPUT DATA FOR EXAMPLE 5: A 3-R MIXED-LOOP PLANAR ROBOT

```
#uscons
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=1
nsc=1
;   tc(i)=Time dependent motion generator.
;   ptc(i,j)=It's partial derivative w.r.t. coord. j
;   pttc(i)=It's partial derivative w.r.t. time
;   dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;   dpttc(i)=The derivative of pttc(i) w.r.t. time
;   sc(i)=Path & Length constraint
;   psc(i,j)=It's partial derivative w.r.t coord. j
;   dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;   The value of nst is passed from the Analysis module to
;   form the acceleration equations
if(nst=2)
goto 10
endif
;   enter code here, for tc,ptc,pttc,sc & psc
;   & with conditions
tc(1)=2*pi*sin(.3927*tm)+1.5708-ti(3)
ptc(1,3)=-1.0
pttc(1)=2.4674*cos(.3927*tm)
sc(1)=-ti(4)
psc(1,4)=-1
goto 20
10:
;  enter code here for dptc &/OR dpttc
dpttc(1)=-.9689*sin(.3927*tm)
20:
return
#end
#usload
;   enter code here for
;   usl(i)=Externally applied load in direction of
;          freedom no. i
return
#end
#drive
;   enter code here for
;   fpc(i)=Driving force or torque
;      i=index corresponds the order by which the input data  for a
;        given driving load is given
return
#end
```

INPUT DATA FOR EXAMPLE 6: A SLIDER CRANK MECHANISM

```
#uscons
;    ntc=Number of time dependent constraints.
;    nsc=Number of spatial constraints.
ntc=0
nsc=0
if(ti(3)>45)
ti(3)=45.0
endif
;    tc(i)=Time dependent motion generator.
;    ptc(i,j)=It's partial derivative w.r.t. coord. j
;    pttc(i)=It's partial derivative w.r.t. time
;    dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;    dpttc(i)=The derivative of pttc(i) w.r.t. time
;    sc(i)=Path & Length constraint
;    psc(i,j)=It's partial derivative w.r.t coord. j
;    dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;    The value of nst is passed from the Analysis module to
;    form the acceleration equations
if(nst=2)
goto 10
endif
;    enter code here, for tc,ptc,pttc,sc & psc
;    & with conditions
goto 20
10:
;   enter code here for dptc &/OR dpttc
20:
return
#end
#usload
;    enter code here for
;    usl(i)=Externally applied load in direction of
;          freedom no. i
return
#end
#drive
;    enter code here for
;    fpc(i)=Driving force or torque
;       i=index corresponds the order by which the input data  for a
;         given driving load is given
fpc(1)=1.*(36.-ti(3))*11.4
return
#end
```

INPUT DATA FOR EXAMPLE 7: A 3R ROBOT UNDER GRAVITY EFFECT

```
#uscons
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=0
nsc=0
;   tc(i)=Time dependent motion generator.
;   ptc(i,j)=It's partial derivative w.r.t. coord. j
;   pttc(i)=It's partial derivative w.r.t. time
;   dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;   dpttc(i)=The derivative of pttc(i) w.r.t. time
;   sc(i)=Path & Length constraint
;   psc(i,j)=It's partial derivative w.r.t coord. j
;   dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;   The value of nst is passed from the Analysis module to
;   form the acceleration equations
if(nst=2)
goto 10
endif
;   enter code here, for tc,ptc,pttc,sc & psc
;   & with conditions
goto 20
10:
;  enter code here for dptc &/OR dpttc
20:
return
#end
#usload
;   enter code here for
;   usl(i)=Externally applied load in direction of
;          freedom no. i
return
#end
#drive
;   enter code here for
;   fpc(i)=Driving force or torque
;      i=index corresponds the order by which the input data  for a
;        given driving load is given
return
#end
```

INPUT DATA FOR EXAMPLE 8: AN APPLICATION TO THE HUMAN SPINE

1. THE INPUT DATA FOR LOADING SET #1

```
#uscons
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=0
nsc=0
;   tc(i)=Time dependent motion generator.
;   ptc(i,j)=It's partial derivative w.r.t. coord. j
;   pttc(i)=It's partial derivative w.r.t. time
;   dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;   dpttc(i)=The derivative of pttc(i) w.r.t. time
;   sc(i)=Path & Length constraint
;   psc(i,j)=It's partial derivative w.r.t coord. j
;   dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;   The value of nst is passed from the Analysis module to
;   form the acceleration equations
if(nst=2)
goto 10
endif
;   enter code here, for tc,ptc,pttc,sc & psc
;   & with conditions
goto 20
10:
;  enter code here for dptc &/OR dpttc
20:
return
#end
#usload
;   enter code here for
;   usl(i)=Externally applied load in direction of
;          freedom no. i
usl(36)=10
return
#end
#drive
;   enter code here for
;   fpc(i)=Driving force or torque
;      i=index corresponds the order by which the input data  for a
;       given driving load is given
fpc(1)=(0.370-ti(11))*100.0
fpc(2)=(0.370-ti(12))*100.0
fpc(3)=(0.370-ti(13))*100.0
fpc(4)=(0.332-ti(14))*100.0
fpc(5)=(0.250-ti(15))*100.0
return
#end
```

1. THE INPUT DATA FOR LOADING SET #2

```
#uscons
;   ntc=Number of time dependent constraints.
;   nsc=Number of spatial constraints.
ntc=0
nsc=0
```

```
;    tc(i)=Time dependent motion generator.
;    ptc(i,j)=It's partial derivative w.r.t. coord. j
;    pttc(i)=It's partial derivative w.r.t. time
;    dptc(i,j)=The derivative of ptc(i,j) w.r.t time
;    dpttc(i)=The derivative of pttc(i) w.r.t. time
;    sc(i)=Path & Length constraint
;    psc(i,j)=It's partial derivative w.r.t coord. j
;    dpsc(i,j)=The derivative of psc(i,j) w.r.t. time
;    The value of nst is passed from the Analysis module to
;    form the acceleration equations
if(nst=2)
goto 10
endif
;    enter code here, for tc,ptc,pttc,sc & psc
;    & with conditions
goto 20
10:
;   enter code here for dptc &/OR dpttc
20:
return
#end
#usload
;    enter code here for
;    usl(i)=Externally applied load in direction of
;           freedom no. i
usl(35)=5
usl(37)=-5
usl(39)=10
usl(40)=-7
return
#end
#drive
;    enter code here for
;    fpc(i)=Driving force or torque
;       i=index corresponds the order by which the input data  for a
;         given driving load is given
fpc(1)=(0.370-ti(11))*100.0
fpc(2)=(0.370-ti(12))*100.0
fpc(3)=(0.370-ti(13))*100.0
fpc(4)=(0.332-ti(14))*100.0
fpc(5)=(0.250-ti(15))*100.0
return
#end
```

ملخص

التحليلات الديناميكية للأنظمة الميكانيكية باستخدام الحاسوب

إعداد: يوسف الكيري خانكيري الشاشاني

إشراف: د. "محمد حمزة" فتح الله الددو

تم في هذه الرسالة تطوير نظام محوسب، يقوم بإجراء تحليلات ديناميكية واستاتيكية للأنظمة الميكانيكية المستوية. يستخدم هذا النظام طريقةً لصياغة المعادلات الحركية والديناميكية تم تطويرها في عمل سابق للدكتور محمد الددو [١]. كما يعتبر هذا العمل إكمالاً للنظام الذي طوره.

يستخدم هذا النظام اسلوباً لبناء نموذج قادر على وصف أي نظام ميكانيكي مع أي تقييد أو مولدات للحركة.

هذا النظام قادر على إجراء تحليلات حركية و ديناميكية و استاتيكية و ايجاد الوضع الثابت المتزن للأنظمة الميكانيكية.

التحليلات الحركية تشمل حساب الموضع والسرعة والتسارع لكل وصلة أو ذراع في النظام الميكانيكي.

يوجد نوعان من التحليلات الديناميكية: أمامية و عكسية. في التحليلات الديناميكية الأمامية تكون القوى المحركة معروفة، ولكن الحركة الناتجة عنها غير معروفة. و في التحليلات الديناميكية العكسية يتم إيجاد أوضاع النظام الميكانيكي عن طريق حل المعادلات الحركية مباشرة.

التحليلات الاستاتيكية تشمل إيجاد الإنحرافات و ردود الفعل على جميع نقاط التوصيل للمنشآت الثابتة.

أما في النوع الأخير من التحليلات فيتم إيجاد الوضع الثابت المتزن للنظام الميكانيكي تحت تأثير قوى ثابتة. يتم حل هذا النوع من المسائل باستخدام مبدأ الشغل الإفتراضي.

أثناء تطوير هذا النظام تمت مراعاة أن يكون سهل الأستعمال سواءً في طريقة إدخال المعلومات التي تصف النظام الميكانيكي أو في طريقة عرض النتائج. فبعد إدخال المعلومات و إجراء التحليلات يستطيع المستعمِل أن يرى النتائج مباشرة على شكل رسومات أو جداول بيانية. كما يستطيع المستعمِل أن يرى محاكاةً لحركة النظام الميكانيكي.