DYNAMICS OF A GENERAL FLEXIBLE MULTIBODY SYSTEM

by

Zhicheng Zhao

A thesis
Presented to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF MANITOBA
Department of Mechanical and Industrial Engineering
Winnipeg, Manitoba, Canada

March 1992

ISBN 0-315-77952-7

DYNAMICS OF A GENERAL FLEXIBLE MULTIBODY SYSTEM

BY

ZHICHENG ZHAO

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

I hereby declare that I am the sole author of this thesis. I authorize the University of manitoba to lend this thesis to othor institutions or individuals for the purpose of scholarly research.

Zhicheng Zhao

I further authorize the University of Manitoba to reproduce this thesis by photocopying or by other means, in whole or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Zhicheng Zhao

Abstract

This thesis presents a dynamic model of a general, flexible multibody system by means of the lumped mass finite element approach formulated using Kane's equations. The system topology considered here is defined as an arbitrary combination of both rigid and flexible bodies, connected together by joints that permit translation and compliance, in a general tree configuration. An extension to handle closed loop kinematic chains is also indicated. Kane's theory of generalized speeds which is based on the *Lagrange-d'Alembert* principle, is used to derive the equations of motion, and this results in a very efficient computer oriented methodology for solving the dynamics of such large mechanical systems. To facilitate numerical computations, the dynamical equations are transformed into a system of first-order differential equations, for an explicit solution of the problem. The accuracy of the proposed formulation is assessed via three examples with known solutions. The results obtained indicate the method is accurate, efficient and versatile for the analysis of a general, flexible multibody system.

Acknowledgments

The author wishes to express his sincere thanks to Professor Ray P.S. Han for his supervision and financial support during the work on this thesis. The author would also like to thank his wife for her patience, encouragement and sacrifices.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Description

A multibody system may be defined as any finite number of bodies, rigid or otherwise, connected together in some arbitrary fashion by joints. Typical examples include spacecrafts with flexible appendages, large space structures, robot manipulators, machines and mechanisms and even the human body which can be modeled as a number of interconnected body parts. With the advent of computers, there has been considerable research in the field of rigid-multibody dynamics. It is only recently that the assumption of the bodies being *rigid* has been dropped to incorporate the effects of *flexibility*. Such effects are a major concern as they exert a strong influence on the dynamic characteristics of mechanical systems that not only operate at high speeds, but are frequently constructed of light-weight materials.

1.2 Previous Work

One of the first papers on the dynamics of multibodies which were modeled as rigid bodies, were described by Hooker and Margulies (1965), and by Roberson and Wittenburg (1966). A large number of papers in this area soon followed, with the three main beneficiaries for this type of research being, space structures, robotics and mechanisms. Like the rigid multibodies, interest in flexible multibody systems can also be grouped into these three distinct areas. Since this thesis is concerned with flexible multibodies, it will be appropriate for the literature review given here to address only such systems. Hence, the review will be organized into the three main areas of research: flexible space structures, compliant robotics and elastic mechanisms.

In the early 1970s, models were developed for the analysis of rigid bodies with elastic appendages [Likins (1970), Hooker (1975), Ho (1977)]. Roberson (1972) introduced relative translation between flexible bodies; Kulla (1972), Bodley and Park (1972), and Likins *et. al.* (1973) presented models for spinning elastic bodies; Ho and Herber (1985) and many others considered flexible spacecrafts. Hughes (1979), Huston (1981), and Singh, VanderVoort and Likins (1985) extended the analysis to handle general, flexible multibodies. Both Huston and Singh *et. al.* formulated their analysis based on a Lagrange's form of the d'Alembert's principle, as first exposited by Kane and Wang (1965). In a series of papers, Shabana (1986), and Agrawal and Shabana (1986) investigated the dynamic characteristics of inertia variant flexible multibody systems using the Lagrange's equation technique. In the dynamics of flexible multibodies that undergo large deformations the recursive formulation appears to be popular and is described in Changizi and Shabana (1988) and Kim and Haug (1988).

In the area of flexible robotics, Sunada and Dubowsky (1983) evaluated the kinetoelastic deformations in industrial manipulators; Book (1984) proposed a simulation model for the dynamics of spatial manipulators with revolute joints; Low (1987) derived the equations of motion for mechanical manipulators with elastic links using the Hamilton's principle; and Lee and Wang (1988) employed the Newton-Euler approach to present dynamic equations for flexible single and double link manipulators, for use in computer simulations.

There is a considerable amount of information in the dynamic analysis of elastic mechanisms and is elegantly summarized in Lowen and Chassapis (1986). Kohli et. al. (1977) investigated the dynamic behavior of an elastic slider-crank mechanism using the Lagrange's equation approach; Thompson and Barr (1976) employed a variational procedure to incorporate constraint equations into Hamilton's principle for a flexible slider-crank mechanism; Jandrasit and Lowen (1979) presented an analytical model of the elastic-dynamic behavior in a four-bar linkage, again using the Hamilton's principle; Dubowsky and Maatuk (1975) using a Lagrangian approach, investigated the vibratory characteristics of spatial elastic mechanisms and manipulators; Sadler and Sandor (1973) and Sadler (1975) studied the kineto-elastodynamic behavior of linkages using a lumped mass model, Kohli and Sandor (1975) applied the theory to analyse an RCCC linkage with three elastic links; and Midha et. al. (1978), Turcic and Midha (1984), and Cleghorn et. al. (1981) presented finite element equations of motion for elastic mechanisms.

1.3 Proposed Research

In this thesis, a general treatment is proposed to model the dynamics of flexible multibody systems connected by hinges which can accomodate both translations and rotations. Unlike the paper by Singh, VanderVoort and Likins (1985) which employed a set of modal coordinates to approximately represent the elastic deformations, the formulation here is based on a lumped mass finite element model. Only multibody systems with a general tree topology, as depicted in *Figure* 2.1 and exhibiting small deformations, are analysed here. An extension of the formulation to handle closed loop kinematic chains is also presented. Several methods of analytical dynamics are available for automatic generation of the equations of motion of these multibodies. For example, methods based on the Newton-Euler approach, the Lagrange's equations technique, Hamilton's principle, or some combination of these, are probably most popular with researchers in this area. However, an increasing number of researchers are formulating their dynamic analysis based on some generalization of the Lagrange's form of the d'Alembert's principle. This is because the resulting governing equations possess superior computational advantage in that, the non-working constraint forces are eliminated without introducing tedious differentiation as in the Lagrangian formulation. One of the most well known theory based on this approach is the Kane's method of generalized speeds (Kane and Wang, 1965), and some of its most ardent supporters have been Likins (1974, 1975), Huston (1978, 1979, 1980, 1981), Levinson (1977) and Kane himself (1965, 1968, 1980, 1983, 1987).

To facilitate numerical calculations, the equations of motion are recast into a system of first-order differential equations, resulting in an explicit formulation of the problem. Three examples with known solutions are solved, and the results obtained indicate the proposed method is accurate, efficient, and versatile for the analysis of a general, flexible multibody system.

Chapter 2

Kinematics of Flexible Multibody Systems

2.1 Introduction

In this chapter, the geometry and kinematics of a flexible multibody system is introduced. In the first part of this chapter, the mathematical tools required to describe the system motion are developed. They comprise the body connection array, the characterization of degrees of freedom and the transformation matrices.

In the second part, the concepts of partial velocity, partial angular velocity and generalized speeds are introduced and having done so, the angular and linear velocities, and angular and linear accelerations for the flexible multibody system are derived.

2.2 System Description

2.2.1 Body Connection Array

If a mechanical system consists of connected bodies, rigid or otherwise, such that adjacent bodies share at least one common point and no closed loops or circuits are formed, the system is called a "general-chain" (or "open-chain") system. *Figure* 2.1 depicts such a system. In order to get a unique kinematic description of the multibodies and their various deformed configurations in the dynamic analysis of a *flexible* multibody system, it is essential to develop a compact and efficient accounting procedure. The procedure adopted here is based on a direct path array used by Kane (1968), and Huston, Passerello and Harlow (1978) in their work on *rigid* multibody systems. The direct path array may be obtained by follows. Arbitrarily select one of the bodies as a reference body and call it B_1. Then number the other bodies of the system in ascending progression away from B_1 as shown in *Figure* 2.1. Let $L(k)$, $k = 1, 2, \ldots, N$, be an array of the adjoining lower numbered body of the kth body. For example, for the system shown in *Figure* 2.1, $L(k)$ is

$$L(k) = (0, 1, 2, 1, 4, 5, 6, 5, 8, 1) \tag{2.1}$$

where

$$(k) = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$$

and it is defined that

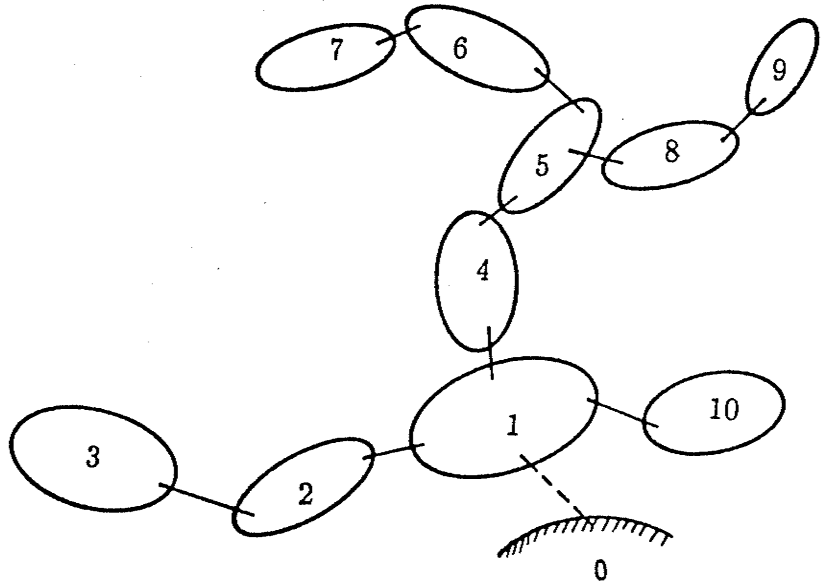


Figure 2.1: A general tree configuration

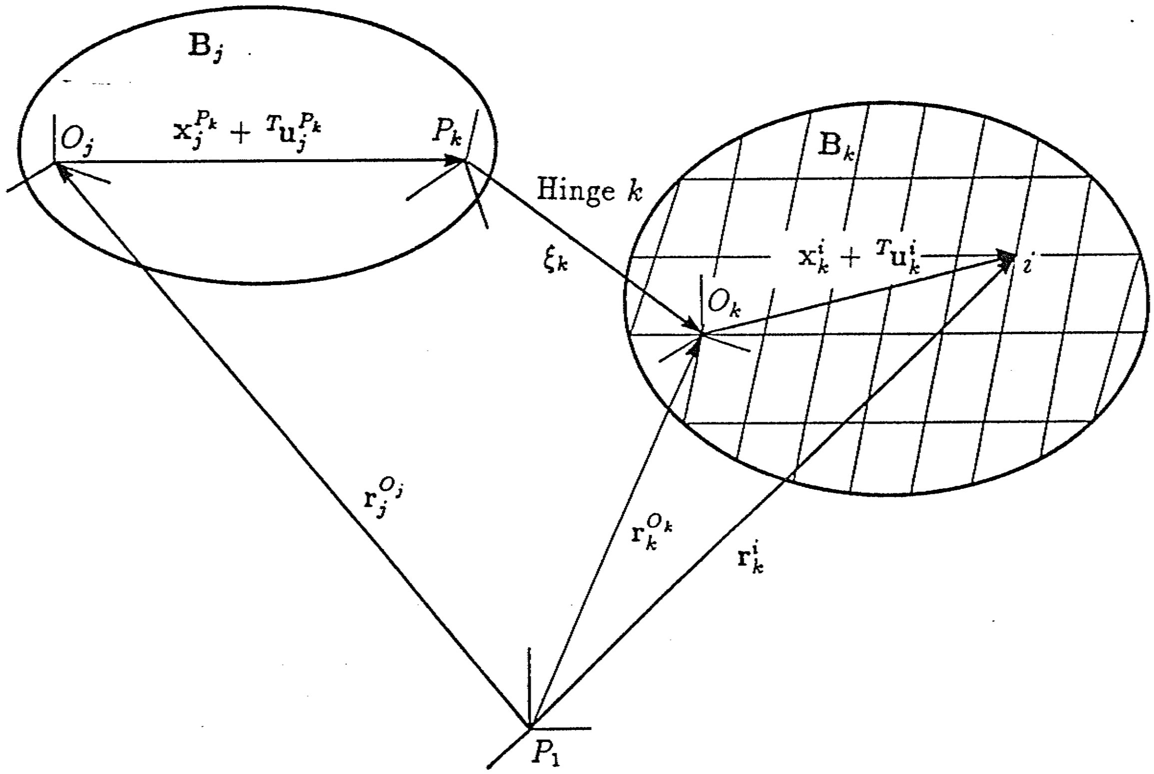
$$L^0(k) = k, \quad L^1(k) = L(k), \quad L^2(k) = L(L(k))$$

and so on.

For convenience, the reference body B_1 is assumed to be connected to an inertially fixed body numbered as 0. Now, a direct path from the inertia frame to any body of the system can be described by the body connection array $L(k)$.

2.2.2 Degrees of Freedom for Flexible Multibody Tree Systems

Consider a typical pair of adjacent bodies such as B_j and B_k shown in *Figure* 2.2. The following degrees of freedom are considered:



Inertial frame **R**

Figure 2.2: Two typical adjacent bodies with joint translation and compliance

1. *Rigid body degrees of freedom.* The kth hinge, which connects the kth body B_k to its lower numbered adjacent body B_j, is assumed, in general, to permit relative rotations and translations. Reference frames $\mathbf{n}_{ki}^{O_k}$ and $\mathbf{n}_{ji}^{P_k}$, (i =1, 2, 3) are fixed at hinge point O_k (a point of B_k) and P_k (a point of B_j) respectively, as shown in *Figure* 2.2. Let NR_k represents the relative rotation degrees of freedom, and NT_k the relative translation degrees of freedom for the kth hinge, then the rigid body degrees of freedom for B_k are given by

$$NB = \sum_{k=1}^{N} NR_k + NT_k \tag{2.2}$$

where N is the number of bodies in the tree topology.

2. *Deformational degrees of freedom.* Generate a finite element mesh for each flexible body, and choose every hinge point P_k as one of the nodes. Let NP_k represents the number of nodes, and NN_k represents the nodal degrees of freedom in B_k. Then the deformational degrees of freedom, ND_k for the kth body, are given by the product of the degrees of freedom per node and the number of nodes employed in the discretization of the body, namely:

$$ND_k = NP_k \times NN_k \tag{2.3}$$

The total number of degrees of freedom for the system can then be written as

$$NS = \sum_{k=1}^{N} NR_k + NT_k + ND_k \tag{2.4}$$

2.2.3 Transformation Matrices

Transformation matrices play an important role in the kinematic and dynamic analysis of multibody systems. The vectors of positions, velocities and accelerations derived in the local reference frame can be transformed into any other reference frames, and in particular, to the inertial reference frame.

As before, let B_k be a typical body of the system and B_j its adjacent lower numbered body, such as shown in *Figure* 2.2. The rigid orientation of B_k relative to B_j may be defined in terms of the relative orientation of the dextral orthogonal unit vector sets $\mathbf{n}_{ji}^{P_k}$ to $\mathbf{n}_{ki}^{O_k}$ $(i = 1, 2, 3)$. A 3×3 orthogonal transformation matrix, $\mathbf{TR}_j^k$ can be defined as

$$TR_{jim}^{k} = \mathbf{n}_{ji}^{P_k} \bullet \mathbf{n}_{km}^{O_k} \tag{2.5}$$

where the j and k in *Equation* (2.5) refer to bodies B_j and B_k. Then $\mathbf{n}_{ji}^{P_k}$ and $\mathbf{n}_{km}^{O_k}$ are related to each other as

$$\mathbf{n}_{ji}^{P_k} = \sum_{m=1}^{3} TR_{jim}^{k}\, \mathbf{n}_{km}^{O_k} \tag{2.6}$$

The orientation of the hinge point, P_k relative to local reference frame $\mathbf{n}_{ji}^{O_j}$, due to the elastic deformation, also can be defined through a 3×3 orthogonal transformation matrix, $\mathbf{TD}_j$ as

$$\mathbf{n}_{ji}^{O_j} = \sum_{m=1}^{3} TD_{jim}\, \mathbf{n}_{jm}^{P_k} \tag{2.7}$$

where the transformation matrix $\mathbf{TD}_j$ is given by

$$TD_{jim} = \mathbf{n}_{ji}^{O_j} \bullet \mathbf{n}_{jm}^{P_k} \tag{2.8}$$

The general orientation of body B_k relative to B_j now can be defined as the relative orientation of the reference frame, $\mathbf{n}_{ki}^{O_k}$ to the reference frame, $\mathbf{n}_{ji}^{O_j}$. It is not difficult to find that $\mathbf{n}_{ji}^{O_j}$ and $\mathbf{n}_{km}^{O_k}$ are related to each other as

$$\mathbf{n}_{ji}^{O_j} = \sum_{m=1}^{3} T_{jim}^{k}\ \mathbf{n}_{km}^{O_k} \tag{2.9}$$

where the general transformation matrix, $\mathbf{T}_j^k$ is given by

$$\mathbf{T}_j^k = \mathbf{TD}_j\ \mathbf{TR}_j^k \tag{2.10}$$

From *Equation* (2.9), it is easily seen that with three bodies, B_j, B_k and B_l, the transformation matrix obeys the following chain and identity rules

$$\mathbf{T}_j^l = \mathbf{T}_j^k\ \mathbf{T}_k^l \tag{2.11}$$

and

$$\mathbf{T}_j^j = \mathbf{I} = \mathbf{T}_j^k\ \mathbf{T}_k^j = \mathbf{T}_j^k\ (\ \mathbf{T}_j^k\ )^{-1} \tag{2.12}$$

where $\mathbf{I}$ is the identity matrix.

These equations allow for the transformation of components of vectors referred to one body of the system into components referred to any other body of the system, and in particular, to the inertial reference frame, R. For example, if a typical vector, $\mathbf{A}$ is expressed as

$$\mathbf{A} = \sum_{i=1}^{3} A_i^{(k)}\ \mathbf{n}_{ki}^{O_k} = \sum_{i=1}^{3} A_i^{(0)}\ \mathbf{n}_{0i} \tag{2.13}$$

then

$$A_i^{(0)} = \sum_{j=1}^{3} T_{0ij}^{k} \; A_j^{(k)} \tag{2.14}$$

where 0 refers to the inertial frame, R.

2.3 Kinematics

2.3.1 Concepts of Partial Velocity and Partial Angular Velocity

Consider a mechanical system S with n degrees of freedom. Let q_1, q_2, ..., and q_n be the generalized coordinates describing the system in reference frame A. Let the derivatives of generalized coordinates be represented by $\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n$, then the generalized speeds, $u_1, u_2, \ldots, u_n$, of the system S relative to the reference frame A are computed from

$$u_r = \sum_{s=1}^{n} Y_{rs}\dot{q}_s + Z_r \qquad (r = 1,\ 2,\ \ldots,\ n) \tag{2.15}$$

where Y_{rs} and Z_r are the functions of n generalized coordinates, $q_1, q_2, \ldots, q_n$ and time t, and are chosen such that all $\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n$ have unique solution in *Equation* (2.15). Note that ($\dot{}$) implies time differentiation in the local reference frame.

Their introduction enable one to take advantages of special features of a given physical system to bring equations of motion into a particularly simple form. Usually, the generalized speeds are chosen to be the quantities related to the movements of the system, such as components of the angular velocity of a body or the linear velocity of a particle, or simply $\dot{q}_r$ for all or part of r.

After determination of generalized speeds, the angular velocity ω, of any body B in system S relative to reference frame A, may be uniquely described by the linear combination of generalized speeds u_r. That is,

$$\omega \;=\; \sum_{r=1}^{n} \boldsymbol{\Omega}_{\mathbf{r}} \; u_r \;+\; \bar{\omega} \tag{2.16}$$

where $\boldsymbol{\Omega}_r$ and $\bar{\omega}$ are computable functions of $q_1,\ q_2,\ \ldots,\ q_n$ and t. The vector $\boldsymbol{\Omega}_r$ is called rth partial angular velocity of the body B relative to A [Kane (1983 a, b, c)].

Similarly, the linear velocity $\mathbf{v}$ of any particle P relative to A, may be given by

$$\mathbf{v} \;=\; \sum_{r=1}^{n} \mathbf{V}_r \; u_r \;+\; \bar{\mathbf{v}} \tag{2.17}$$

where $\mathbf{V}_r$ and $\bar{\mathbf{v}}$ are computable functions of $q_1,\ q_2,\ \ldots,\ q_n$ and t. The vector $\mathbf{V}_r$ is called rth partial velocity of the particle P relative to A.

2.3.2 Generalized Speeds

The following generalized vectors are defined:

1. $\varphi_k, (k = 1, 2, \ldots, N)$ represents the relative orientation vector of the kth body to the reference frame fixed at hinge point, P_k:

$$\varphi_k = \sum_{l=1}^{NR_k} \varphi_{kl} \; \mathbf{n}_{kl}^{P_k} \tag{2.18}$$

2. $\xi_k, (k = 1, 2, \ldots, N)$ represents the relative translational displacement vector of the kth body to the reference frame fixed at hinge point, P_k:

$$\xi_k = \sum_{l=1}^{NT_k} \xi_{kl}\ \mathbf{n}_{kl}^{P_k} \tag{2.19}$$

3. ${}^{T}\mathbf{u}_k^i,\ {}^{R}\mathbf{u}_k^i, (k = 1, 2, \ldots, N; i = 1, 2, \ldots, NP_k)$ represent the elastic nodal displacement vectors for translational and rotational degrees of freedom respectively, of the ith node in the kth body to the reference frame fixed at hinge point, O_k,:

$${}^{T}\mathbf{u}_k^i = \sum_{l=1}^{NN_k^T} {}^{T}u_{kl}^i\ \mathbf{n}_{kl}^{O_k} \tag{2.20}$$

$${}^{R}\mathbf{u}_k^i = \sum_{l=1}^{NN_k^R} {}^{R}u_{kl}^i\ \mathbf{n}_{kl}^{O_k} \tag{2.21}$$

Note that in general, $NN_k^T + NN_k^R = NN_k \leq 6$.

To compactly summarize the above quantities, a generalized coordinate q_m can be defined by ordering its components as follows,

$$q_m = \begin{cases} m = 1, 2, \ldots, \sum_{k=1}^{N} NR_k & \text{relative rigid orientations} \\ m = \sum_{k=1}^{N} NR_k + 1, \ \ldots, \sum_{k=1}^{N} NR_k + NT_k & \text{relative rigid translations} \\ m = \sum_{k=1}^{N} NR_k + NT_k + 1, \ldots, NS & \text{elastic deformations} \end{cases} \tag{2.22}$$

For convenience, let the generalized speeds of the system be denoted by

$$y_m = \dot{q}_m \qquad (m = 1, 2, \ldots, NS) \tag{2.23}$$

2.3.3 Angular Velocity

The absolute angular velocity of the kth body can be represented by the addition formula as

$$\begin{aligned} \omega_k &= \hat{\omega}_1 + \hat{\omega}_2 + \cdots + \hat{\omega}_{L(k)} + \hat{\omega}_k \\ &\quad + {}^{R}\dot{\mathbf{u}}_1^{P_2} + \cdots + {}^{R}\dot{\mathbf{u}}_{L^2(k)}^{P_{L(k)}} + {}^{R}\dot{\mathbf{u}}_{L(k)}^{P_k} \end{aligned} \tag{2.24}$$

where the relative angular velocities, $\hat{\omega}_i$ are each measured with respect to the respective adjacent lower numbered bodies and is summed over the bodies of the chain from B_1 outward to B_k. In the case of flexible bodies, the deformation of each body, including the rotational rate of each node, ${}^{R}\dot{\mathbf{u}}_{L(k)}^{P_k}$ due to elastic deformation, should be taken into account. So $\hat{\omega}_k$ is exactly the relative angular velocity of the local reference frame, $\mathbf{n}_{ki}^{O_k}$ to the reference frame, $\mathbf{n}_{ji}^{P_k}$ due to rigid body rotation.

The body connection array $L(k)$ is very useful in computing this sum. Considering the system in *Figure* 2.1, for example. The angular velocity of B_7 is given by

$$\begin{aligned} \omega_7 &= \hat{\omega}_1 + \hat{\omega}_4 + \hat{\omega}_5 + \hat{\omega}_6 + \hat{\omega}_7 \\ &\quad + {}^{R}\dot{\mathbf{u}}_1^{P_4} + {}^{R}\dot{\mathbf{u}}_4^{P_5} + {}^{R}\dot{\mathbf{u}}_5^{P_6} + {}^{R}\dot{\mathbf{u}}_6^{P_7} \end{aligned} \tag{2.25}$$

Using connection array, ω_7 may be written as

$$\omega_7 = \sum_{\gamma=0}^{4} \left(\hat{\omega}_s + {}^{R}\dot{\mathbf{u}}_{L(s)}^{P_s} \right) \tag{2.26}$$

where $s = L^{\gamma}(7)$, i.e. $\gamma = 0$, $s = L^0(7) = 7$; $\gamma = 1$, $s = L^1(7) = 6$; $\gamma = 2$, $s = L^2(7) = 5$, $\ldots$, $\gamma = 4$, $s = L^4(7) = 1$.

Now, in general, the absolute angular velocity of B_k may be written as

$$\omega_k = \sum_{\gamma=0}^{u} \left(\hat{\omega}_s + {}^{R}\dot{\mathbf{u}}_{L(s)}^{P_s} \right) \tag{2.27}$$

where $s = L^{\gamma}(k)$ and the index u is defined such that $L^u(k) = 1$.

The angular velocity of ith node in the kth body can be represented as

$$\omega_k^i = \omega_k + {}^{R}\dot{\mathbf{u}}_k^i \tag{2.28}$$

where ω_k is the absolute angular velocity of B_k given by *Equation* (2.27) and ${}^{R}\dot{\mathbf{u}}_k^i$ is the nodal rotational rate of the ith node in body B_k.

The absolute angular velocity, ω_k in *Equation* (2.27) and the nodal rotational rate, ${}^{R}\dot{\mathbf{u}}_k^i$ of *Equation* (2.28) can be expressed in terms of the generalized speeds, y_m as

$$\omega_k = \sum_{m=1}^{NS} \mathbf{\Omega}_{km}\, y_m \tag{2.29}$$

$${}^{R}\dot{\mathbf{u}}_k^i = \sum_{m=1}^{NS} {}^{R}\mathbf{U}_{km}^i\, y_m \tag{2.30}$$

The term, $\mathbf{\Omega}_{km}$ is the mth partial angular velocity of B_k and ${}^{R}\mathbf{U}_{km}^i$ is the mth partial rotational rate of the ith node in B_k. Note that $\bar{\omega}_k$ and ${}^{R}\bar{\dot{\mathbf{u}}}_k^i$ are zero in this case. In a similar manner, the absolute angular velocity of the ith node given by *Equation* (2.28) is also expressible in terms of the generalized speeds, y_m. That is,

$$\omega_k^i = \sum_{m=1}^{NS} \left(\mathbf{\Omega}_{km} + {}^{R}\mathbf{U}_{km}^i \right) y_m \tag{2.31}$$

2.3.4 Linear Velocity

The linear velocity of each node of the kth body may be obtained as follows. First, define $\mathrm{r}_k^{O_k}$ be the absolute position vector of the hinge point O_k in body B_k; x_k^i be the relative position vector from the original point of the reference frame fixed at hinge point O_k to the ith node in B_k; and ${}^{T}\mathrm{u}_k^i$ represents the elastic translational displacement of ith node in B_k, as sketched in *Figure* 2.2. Then, the position vector of the ith node in B_k relative to a fixed point O in the inertia frame R may be written as

$$\mathrm{r}_k^i = \mathrm{r}_k^{O_k} + \mathrm{x}_k^i + {}^{T}\mathrm{u}_k^i \tag{2.32}$$

where $\mathrm{r}_k^{O_k}$ represents the summation of all position vectors, starting from the original point of the inertia frame outward through the reference body B_1 and the branch containing the kth body to B_k, and is given by,

$$\mathrm{r}_k^{O_k} = \mathrm{r}_1^{O_1} + \sum_{\gamma=0}^{u-1} \left(\mathrm{x}_{L(s)}^{P_s} + {}^{T}\mathrm{u}_{L(s)}^{P_s} + \xi_s \right) \tag{2.33}$$

where $s = L^{\gamma}(k)$, the index $(u-1)$ is defined such that $L^{u}(k) = 1$ and $\mathrm{r}_1^{O_1}$ represents the absolute position vector of the hinge point O_1 in B_1; $\mathrm{x}_{L(s)}^{P_s}$ and

${}^{T}\mathbf{u}_{L(s)}^{P_s}$ refer, respectively, to the position vector and the elastic translation of hinge point P_s in body $B_{L(s)}$; and ξ_s is the relative translational displacement of the sth body to the $L(s)$th body.

Differentiating *Equation* (2.32) with respect to time yields the linear velocity of the ith node in body B_k as

$$\mathbf{v}_k^i = \mathbf{v}_k^{O_k} + \omega_k \times \left(\mathbf{x}_k^i + {}^{T}\mathbf{u}_k^i\right) + {}^{T}\dot{\mathbf{u}}_k^i \tag{2.34}$$

As in the case of angular velocity, the concept of partial velocity is also introduced here:

$$\mathbf{v}_k^{O_k} = \sum_{m=1}^{NS} \mathbf{V}_{km}^{O_k}\, y_m \tag{2.35}$$

$$\mathbf{v}_k^i = \sum_{m=1}^{NS} \mathbf{V}_{km}^i\, y_m \tag{2.36}$$

$${}^{T}\dot{\mathbf{u}}_k^i = \sum_{m=1}^{NS} {}^{T}\mathbf{U}_{km}^i\, y_m \tag{2.37}$$

where $\mathbf{V}_{km}^{O_k}$ and $\mathbf{V}_{km}^i$ represent the mth partial velocities of paticales O_k and i in body B_k, respectively; ${}^{T}\mathbf{U}_{km}^i$ represents the partial elastic translational velocity of ith node in body B_k, and y_m the generalized speeds.

The velocity vector, $\mathbf{v}_k^i$ can then be compactly expressed in terms of generalized speeds, y_m as

$$\mathbf{v}_k^i = \sum_{m=1}^{NS} \left[\mathbf{V}_{km}^{O_k} + \mathbf{\Omega}_{km} \times \left(\mathbf{x}_k^i + {}^{T}\mathbf{u}_k^i\right) + {}^{T}\mathbf{U}_{km}^i\right] y_m \tag{2.38}$$

2.3.5 Angular Acceleration

The angular acceleration of the ith node in body B_k, in terms of generalized speeds and accelerations, y_m and $\dot{y}_m$, may be given by differentiating *Equation* (2.31)

$$\alpha_k^i = \sum_{m=1}^{NS} \left(\dot{\mathbf{\Omega}}_{km} + {}^R\dot{\mathbf{U}}_{km}^i\right) y_m + \sum_{m=1}^{NS} \left(\mathbf{\Omega}_{km} + {}^R\mathbf{U}_{km}^i\right) \dot{y}_m \qquad (2.39)$$

2.3.6 Linear Acceleration

Differentiating *Equation* (2.38) yields the linear acceleration of the i th node in body B_k, in terms of generalized speeds and accelerations, y_m and $\dot{y}_m$:

$$\begin{aligned} \mathbf{a}_k^i \;=\; & \sum_{m=1}^{NS} \left[\dot{\mathbf{V}}_{km}^{O_k} y_m + \mathbf{V}_{km}^{O_k} \dot{y}_m + \left(\dot{\mathbf{\Omega}}_{km} y_m + \mathbf{\Omega}_{km} \dot{y}_m\right) \times \left(\mathbf{x}_k^i + {}^T\mathbf{u}_k^i\right)\right] \\ & + \left(\sum_{m=1}^{NS} \mathbf{\Omega}_{km} y_m\right) \times \left(\sum_{m=1}^{NS} \mathbf{\Omega}_{km} y_m\right) \times \left(\mathbf{x}_k^i + {}^T\mathbf{u}_k^i\right) \\ & + 2\left(\sum_{m=1}^{NS} \mathbf{\Omega}_{km} y_m\right) \times \left(\sum_{m=1}^{NS} {}^T\mathbf{U}_{km}^i y_m\right) \\ & + \left(\sum_{m=1}^{NS} {}^T\mathbf{U}_{km}^i \dot{y}_m\right) \end{aligned} \qquad (2.40)$$

Equation (2.40) can also be put into a familiar form by an appropriate grouping of the various terms, so that the acceleration components become more recognizable. That is,

$$\mathbf{a}_k^i = \mathbf{a}_k^{O_k} + \alpha_k \times \left(\mathbf{x}_k^i + {}^T\mathbf{u}_k^i\right) + \omega_k \times \omega_k \times \left(\mathbf{x}_k^i + {}^T\mathbf{u}_k^i\right) + 2\omega_k \times {}^T\dot{\mathbf{u}}_k^i + {}^T\ddot{\mathbf{u}}_k^i \qquad (2.41)$$

in which the first term represents the acceleration of the reference frame at hinge point O_k , the second term is the tangential acceleration, the third term is the centrifugal acceleration, the fourth term represents the Coriolis acceleration and the last term is due to the elastic deformation. Obviously this last term, along with the other elastic terms, vanish for the case of rigid body dynamics. Note that *Equation* (2.41) can also be derived by directly differentiating *Equation* (2.34).

Chapter 3

Differential Equations of Motion

3.1 Introduction

Dynamical equations of motion for multibody systems can be derived by Newton-Euler method or by Lagrangian method. The equations obtained from the Newton-Euler formulation include the constraint forces acting between adjacent bodies. Thus, additional arithmetic operations are required to eliminate these terms and obtain explicit relations between the joint torques and the resultant motion in terms of joint displacements. In the Lagrangian formulation the system's dynamic behaviour is described in terms of work and energy using generalized coordinates. The method adopted here is called Kane's method of generalized speed which combines the computational advantages of both Newton-Euler and Lagrangian formulations in that the non-working constraint forces and torques are automatically eliminated and tedious differentiation of the scalar energy functions is avoided. The dimension of the resulting equations is thus minimum.

In order to apply Kane's method to flexible multibody systems, a flexible body is discretized into a system of particles, using a lumped mass finite element approach.

A brief review of Kane's method is given in the second section. The third section presents the derivation of equations of motion for open loop flexible multibody system. The basic parts needed for the construction of Kane's dynamical equations, namely generalized inertia forces and generalized active forces, are derived in two subsections, respectively. An extension to apply Kane's equation to the closed loop multibody system is given in the fourth section.

3.2 On Kane's Equations

Given a system S possessing n degrees of freedom in a Newtonian reference frame R, and composed of N particles, $P_1,\ P_2,\ \ldots,\ P_N$ having masses $m_1,\ m_2,\ \ldots,\ m_N$, respectively. Consider one of the particles, P_i and let $\mathbf{a}_i$ be the acceleration of P_i in inertia reference frame R, $\mathbf{R}_i$ be the resultant of all contact and body forces acting on P_i. Then in accordance with *d'Alembert's* principle we have

$$\mathbf{R}_i + \mathbf{R}_i^* = 0 \qquad (i = 1,\ 2,\ \ldots,\ N) \tag{3.1}$$

where $\mathbf{R}_i^*$ is the inertia force for P_i in R, which is defined as

$$\mathbf{R}_i^* = -m_i\,\mathbf{a}_i \qquad (i = 1,\ 2,\ \ldots,\ N) \tag{3.2}$$

Dot-multiplication of *Eqution* (3.1) with $\mathbf{V}_m^{P_i}$, the mth partial velocity of P_i in R, yields

$$\mathbf{V}_m^{P_i} \bullet \mathbf{R}_i + \mathbf{V}_m^{P_i} \bullet \mathbf{R}_i^* = 0 \tag{3.3}$$

where $m = 1, 2, \ldots, n; \quad i = 1, 2, \ldots, N$. Summing over all particles of S, it can be written as

$$\sum_{i=1}^{N} \mathbf{V}_m^{P_i} \bullet \mathbf{R}_i + \sum_{i=1}^{N} \mathbf{V}_m^{P_i} \bullet \mathbf{R}_i^* = 0 \tag{3.4}$$

where $m = 1, 2, \ldots, n$. Define

$$F_m = \sum_{i=1}^{N} \mathbf{V}_m^{P_i} \bullet \mathbf{R}_i \qquad (m = 1,\ 2,\ \ldots,\ n) \tag{3.5}$$

and

$$F_m^* = \sum_{i=1}^{N} \mathbf{V}_m^{P_i} \bullet \mathbf{R}_i^* \qquad (m = 1,\ 2,\ \ldots,\ n) \tag{3.6}$$

where F_r and F_r^* are called generalized active forces and generalized inertia forces for S in R, respectively. Then *Equation* (3.4) becomes

$$F_m + F_m^* = 0 \qquad (m = 1,\ 2,\ \ldots,\ n) \tag{3.7}$$

These equations are called *Lagrange's form of d'Alembert's principle* or *Kane's dynamical equations.*

3.3 Open Loop Multibody System

3.3.1 Generalized Active Forces

For the discretized multibody system associated with the proposed lumped finite element model, the generalized active forces may expressed as

$$F_m = \sum_{k=1}^{N} \sum_{i=1}^{NP_k} \mathbf{V}^i_{km} \bullet \mathbf{F}^i_k \tag{3.8}$$

where $\mathbf{F}^i_m$ is the resultant force acting on ith node, and $m = 1,\ 2,\ ...,\ NS$.

Substituting partial velocity $\mathbf{V}^i_{km}$ given by *Equation* (2.38) into *Equation* (3.8) yields

$$F_m = \sum_{k=1}^{N} \sum_{i=1}^{NP_k} [\mathbf{V}^{O_k}_{km} + \boldsymbol{\Omega}_{km} \times (\mathbf{x}^i_k + {}^T\mathbf{u}^i_k) + {}^T\mathbf{U}^i_{km}] \bullet \mathbf{F}^i_k \tag{3.9}$$

Define $\mathbf{F}_k$ be the resultant working forces applied at the hinge point O_k and $\mathbf{M}^{O_k}_k$ be the resultant moment of working forces about O_k in B_k. The following relationship are valid:

$$\sum_{k=1}^{N} \sum_{i=1}^{NP_k} \mathbf{V}^{O_k}_{km} \bullet \mathbf{F}^i_k = \sum_{k=1}^{N} \mathbf{V}^{O_k}_{km} \bullet \mathbf{F}_k \tag{3.10}$$

$$\sum_{k=1}^{N} \sum_{i=1}^{NP_k} \boldsymbol{\Omega}_{km} \times (\mathbf{x}^i_k + {}^T\mathbf{u}^i_k) \bullet \mathbf{F}^i_k = \sum_{k=1}^{N} \boldsymbol{\Omega}_{km} \bullet \mathbf{M}^{O_k}_k \tag{3.11}$$

Then divide the resultant force at ith node, $\mathbf{F}^i_k$ into two parts: external force $\mathbf{P}^{i(e)}_k$ and internal elastic force $\mathbf{P}^{i(i)}_k$ at the ith node for the discretized bodies. That is

$$\mathbf{F}^i_k = \mathbf{P}^{i(e)}_k + \mathbf{P}^{i(i)}_k \tag{3.12}$$

where the internal elastic force $\mathbf{P}_k^{i(i)}$ may be obtained by the product of the generalized stiffness matrix and the vector of the generalized coordinates.

Finally, the active forces of the system, F_m may be written as

$$F_m = \sum_{k=1}^{N} [\mathbf{V}_{km}^{O_k} \bullet \mathbf{F}_k + \mathbf{\Omega}_{km} \bullet \mathbf{M}_k^{O_k} + \sum_{k=1}^{NP_k} {}^T\mathbf{U}_{km}^i \bullet (\mathbf{P}_k^{i(e)} + \mathbf{P}_k^{i(i)})] \quad (3.13)$$

where the first two terms represent rigid body dynamics, while the last term is due to the elastic deformations of the flexible bodies.

3.3.2 Generalized Inertia Forces

The generalized inertia forces, F_m^* for the discretized system, may be given by

$$F_m^* = \sum_{k=1}^{N} \sum_{i=1}^{NP_k} \mathbf{V}_{km}^i \bullet m_k^i \, \mathbf{a}_k^i \quad (3.14)$$

Substituting the acceleration and partial velocity terms given by *Equations* (2.41) and (2.38) respectively,into *Equation* (3.14) yields,

$$\begin{aligned} F_m^* = & \sum_{k=1}^{N} \sum_{i=1}^{NP_k} m_k^i \, [\mathbf{a}_k^{O_k} + \alpha_k \times (\mathbf{x}_k^i + {}^T\mathbf{u}_k^i) \\ & + \omega_k \times \omega_k \times (\mathbf{x}_k^i + {}^T\mathbf{u}_k^i) + 2\omega_k \times {}^T\dot{\mathbf{u}}_k^i + {}^T\ddot{\mathbf{u}}_k^i] \\ & \bullet [\mathbf{V}_{km}^{O_k} + \mathbf{\Omega}_{km} \times (\mathbf{x}_k^i + {}^T\mathbf{u}_k^i) + {}^T\mathbf{U}_{km}^i] \end{aligned} \quad (3.15)$$

To facilitate numerical computations, the equations of motion given in *Equation* (3.7) are rearranged into a system of first-order differential equations for an explicit formulation of the problem, namely:

$$\begin{aligned} \sum_{p=1}^{NS} \mathrm{b}_{mp} \, \dot{y}_p &= f_m, \qquad m = 1, 2, \ldots, NS \\ \dot{q}_p &= y_p, \qquad p = 1, 2, \ldots, NS \end{aligned} \quad (3.16)$$

where y_p is a vector of generalized speeds, $\mathbf{b}_{mp}$ is the generalized mass matrix, and f_m is the generalized load vector. Complete expression for these two terms are given in Appendix B.

These differential equations were solved using DGEAR routine available in the IMSL (International Mathematical and Statistical Library) package.

3.4 Closed Loop Multibody System

So far, the formulation is applicable only to the analysis of a flexible multibody system with an open loop chain configuration. To extend the technique to handle a closed kinematic chain, the equations of motion given by *Equation* (3.7) must be suitably modified. Singh and Likins (1985), Wampler, Buffinton and Shu-hui (1985), Wang and Huston (1987) and Amirouche and Huston (1988) have presented alternative schemes to construct these constrained multibody system equations for Kane's theory. In another work on constrained systems, Shabana (1985) discussed the effects of consistent, lumped and hybrid mass modelling of inertia properties of flexible components that exhibit large angular rotations. Bakr and Shabana (1986) analysed the dynamics of flexible constrained multibody systems that undergo large deformations. In both studies, the equations of motion are formulated using the Lagrange's equation and the dependent coordinates are eliminated through the use of constraint equations. Several techniques are available for introducing these constraint equations into the differential equations of motion. They include for example, the Lagrange's multipliers method, the penalty function method, and in the approach given here, a direct substitution method is employed. This procedure appears to be conveniently suited for use in Kane's theory, and it involves cutting open a closed loop system at suitable hinges, into one or more open loop structures. The equations of motion for these open loop chains are derived and constraint equations are then directly substituted into them, yielding the equations of motion for the original closed loop system. An outline of this modification is summarized next.

Assume the number of additional independent constraints being introduced into the open loop system is NC, then the degrees of freedom for the closed loop system, NS' is given by

$$NS' = NS - NC \tag{3.17}$$

where NS is the degrees of freedom for the corresponding open loop system.

Constraint equations can be obtained through the use of loop closure equations and when differentiated in time, yield expressions involving generalized speeds. Wampler *et al.* have shown that if the constraint equations after time differentiated, are assumed linear, then the following relationship is also valid:

$$\dot{q}_m = \sum_{n=1}^{NS'} \mathbf{A}_{mn}\, \dot{q}'_n + B_m \tag{3.18}$$

where $\dot{q}_m$ represents the generalized speeds of the open loop system, $\dot{q}'_n$ represents the generalized speeds of the closed loop system and $\mathbf{A}_{mn}$, B_m are functions of generalized coordinates and time. Recognizing that the square-bracket term on the right-hand side of *Equation* (2.38) is the partial velocity term, namely,

$$\mathbf{v}_k^i = \sum_{m=1}^{NS} \mathbf{V}_{km}^i \, \dot{q}_m \tag{3.19}$$

we have after substituting from *Equation* (3.18),

$$\mathbf{v}_k^i = \sum_{m=1}^{NS} \mathbf{V}_{km}^i \left(\sum_{n=1}^{NS'} \mathbf{A}_{mn} \, \dot{q}_n' + B_m \right) \tag{3.20}$$

Observe in *Equation* (3.20) that the partial velocity for the closed loop system is now expressed in terms of product the partial velocity for the open loop system and the matrix, $\mathbf{A}_{mn}$. That is,

$$\mathbf{V}_{kn}'^i = \sum_{m=1}^{NS} \mathbf{V}_{km}^i \, \mathbf{A}_{mn} \tag{3.21}$$

The generalized active and inertia forces for the closed loop system are now given by the dot product of their respective force components and the partial velocity for the closed loop system:

$$F_n' = \sum_{k=1}^{N} \sum_{i=1}^{NP_k} \left(\mathbf{F}_k^i + \mathbf{F}_k'^i \right) \bullet \mathbf{V}_{kn}'^i \tag{3.22}$$

$$F_n'^* = \sum_{k=1}^{N} \sum_{i=1}^{NP_k} m_k^i \, \mathbf{a}_k^i \bullet \mathbf{V}_{kn}'^i \tag{3.23}$$

where $\mathbf{F}_k'^i$ represents the additional forces due to constraints. The equations of motion for a closed loop system are thus given by,

$$F_n' + F_n'^* = 0, \qquad n = 1, 2, \ldots, NS' \tag{3.24}$$

A more useful form of *Equation* (3.24) is to relate it to the terms given in open loop equations. In this way, it can be immediately seen that the equations of motion for a closed loop system are simply a recombination of the open loop equations and the constraint equations, and the computer program can then be adjusted with minimum alterations. Hence, we have,

$$\sum_{m=1}^{NS} \mathbf{A}_{mn} \left(F_m + F_m^* + \sum_{k=1}^{N} \sum_{i=1}^{NP_k} \mathbf{F}_k'^i \bullet \mathbf{V}_{km}^i \right) = 0 \tag{3.25}$$

Chapter 4

Results and Discussion

4.1 Introduction

In order to illustrate the accuracy and versatility of the proposed technique and the program DAFMS (Dynamic Analysis of Flexible Multibody Systems), three flexible multibody examples with analytical or published solutions are solved and presented for comparison. Both rigid body and flexible body motions are shown. Very good agreement with the known solutions are obtained.

4.2 An Elastic Simple Pendulum

Figure 4.1 shows a two-degree-of-freedom system comprised of a weightless, linear spring and a particle, with the spring being free to rotate about a horizontal axis. This is the simplest example of the flexible multibody systems ($N = 1,\ NS = 2$), and its exact solution is available.

The equations of motion for this two-degree-of-freedom system as given

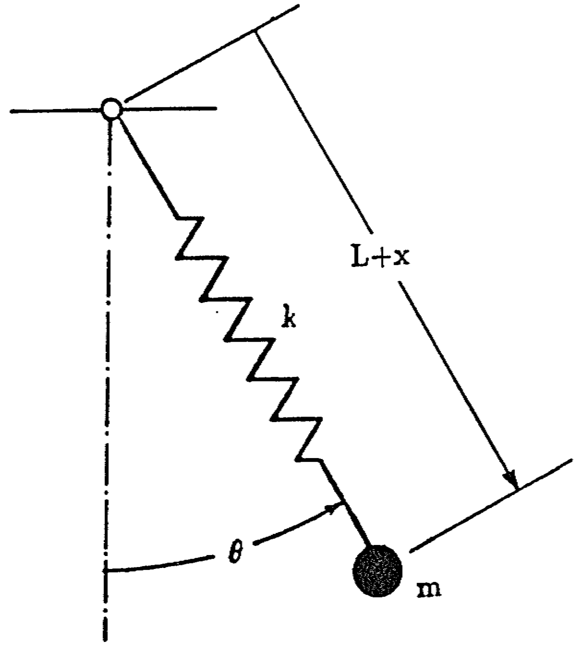


Figure 4.1: An elastic simple pendulum

in Kane and Kahn (1968) are

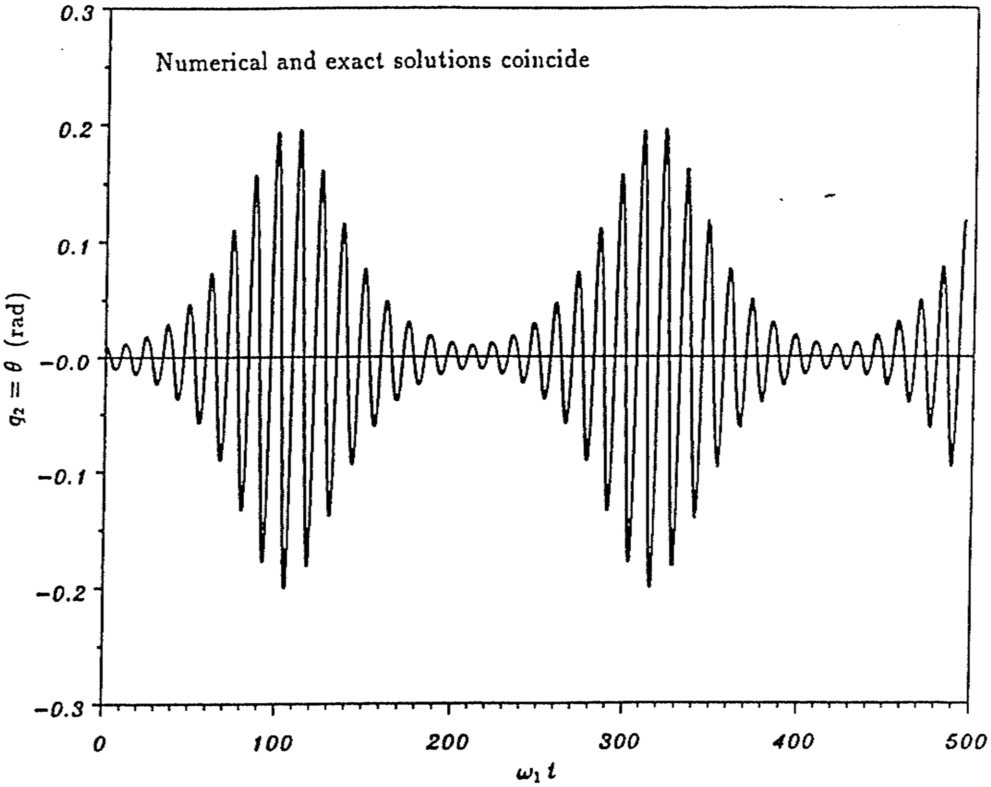
$$
\begin{aligned}
\ddot{q}_1 + \omega_1^2\, q_1 - (1+q_1)\dot{q}_2^2 + \omega_2^2(1-\cos q_2) &= 0 \\
\ddot{q}_2 + \frac{2}{1+q_1}\,\dot{q}_1\,\dot{q}_2 + \omega_2^2\,\frac{1}{1+q_1}\,\sin q_2 &= 0
\end{aligned}
\tag{4.1}
$$

where $\omega_1^2 = k/m$, and $\omega_2^2 = g/L$, g is the gravitational constant, $q_1 = x/L$ and $q_2 = \theta$; k is the spring constant, L the length of spring in static equilibrium, x the spring displacement and m is the mass of the particle.

Assume $\omega_1/\omega_2 = 0.5$, and initial condition to be

$$
\begin{aligned}
q_1(0) &= 0.1, & q_2(0) &= 0.01 \\
\dot{q}_1(0) &= 0, & \dot{q}_2(0) &= 0
\end{aligned}
$$

The numerical integration results are displayed in *Figure* 4.2. Identical agree-



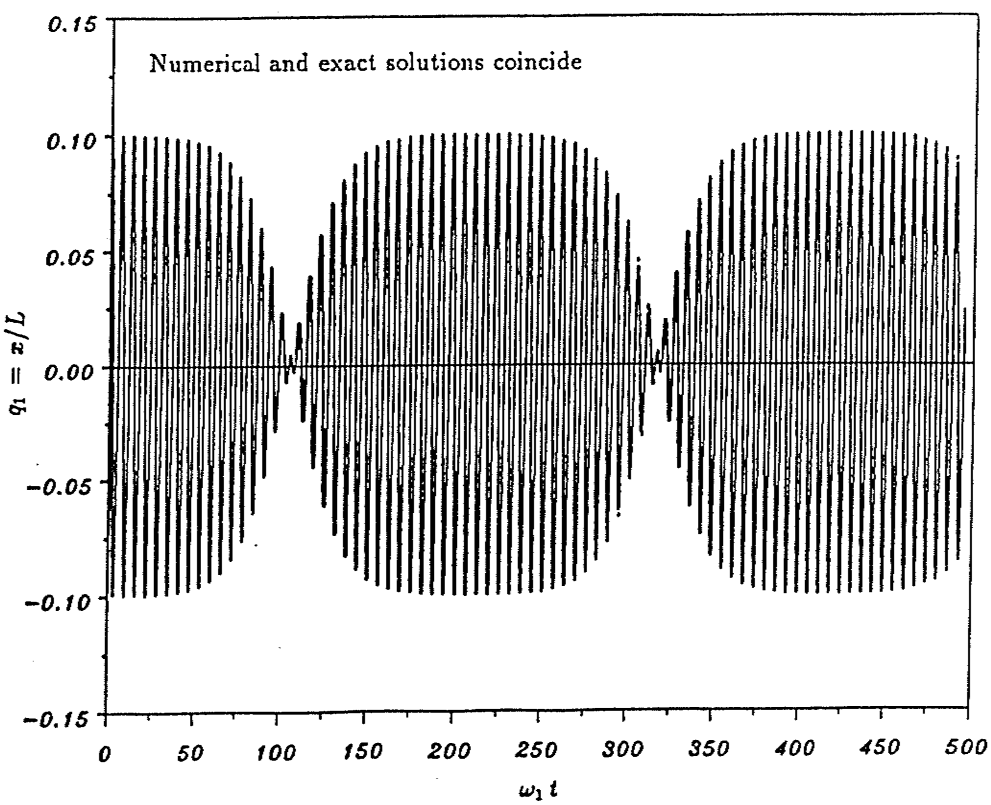


Figure 4.2: System responses (numerical and analytical solutions coincide)

Table 4.1: Comparison of maximum q_1 and q_2

	(q_1)max	(q_2)max
DAFMS Solution	0.0999	0.1986
Exact	0.0999	0.1986
Kane and Kahn(1968)	0.1000	0.2002

ment between the solution given by the program DAFMS and the exact solutions of *Equation* (4.1) is observed. The maximum values of q_1 and q_2 are listed in Table 4.1.

Figure 4.2 shows a very interesting phenomenon. The maximum value of θ grows with successive oscillations; the motion eventually becomes almost entirely pendulum like; the θ-oscillations then decrease; and this process repeats itself periodically. This phenomenon is called non-linear resonance, which should occur under the selected input values.

4.3 A Rotating Rigid Shaft-Flexible Beam Multibody System

In mechanical systems, because of increasing operating speeds and reduced weights, the oscillatory elastic motion becomes a problem in operation. The effect of flexibility on the dynamic behavior of mechanical systems has warranted a good deal of attention. This example investigates the flexural motion of a beam firmly attached to a radially rotating rigid body. This model rep-

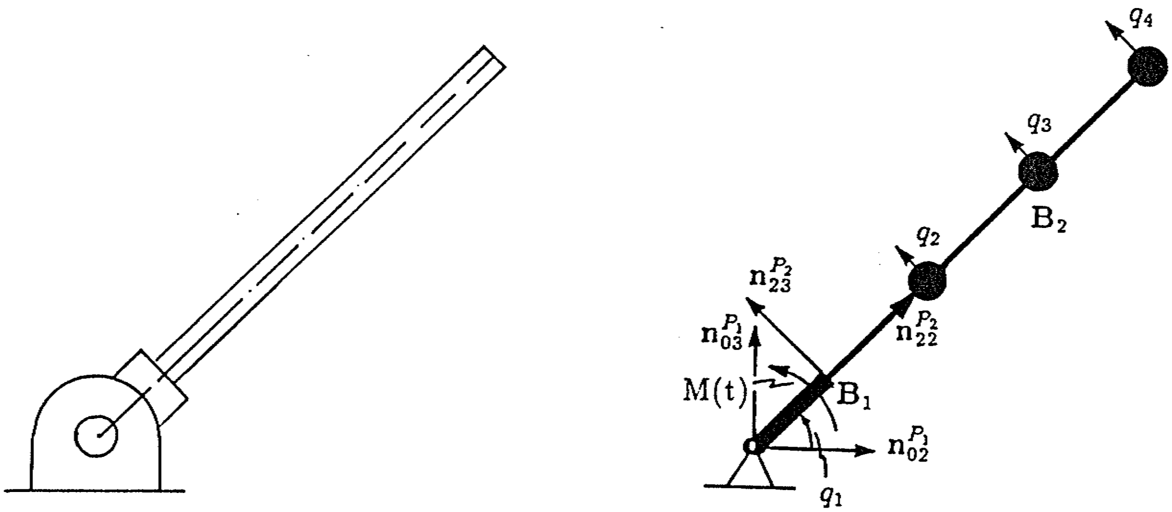


Figure 4.3: A rotating rigid shaft-flexible beam multibody system

resents a variety of technological problems, such as a high speed-low weight manipulator, a turbine blade, a helicopter rotor, and a space satellite with flexible appendages, etc.

Consider in *Figure* 4.3, a slender flexible beam firmly attached to a rotating rigid shaft which is subjected to an applied torque, $M(t)$. Because the rotating shaft is driven by the applied torque, its rigid body motion is not known *apriori*, and the problem therefore consists of solving for both rigid body motion and the flexible beam vibration. It is assumed that the elastic motions have no influence on the rigid body motion, but the vice-versa, the effects of the rigid body motion on the elastic body motions are not neglected. This would result in an uncoupled set of equations, in *Equation* (3.16), for solving the rigid body motions.

This rotating beam problem has been investigated by several researchers, but for our purpose, comparison with the published results of Yigit, Scott and Ulsoy (1988) will be made here. In their example, they considered a torque pulse loading as shown in *Figure* 4.4. The input parameters used were:

$$EI = 5.50 \text{ N/m}^2, \qquad \rho = 0.0858 \text{ kg/m}^3, \qquad L = 0.5 \text{ m}$$

$$a = 0.05 \text{ m}, \quad M = \pm 1 \text{ Nm}$$

$$t_1 = 0.05 \text{ s}, \quad t_2 = 0.1\text{s}, \quad t_3 = 0.15 \text{ s}$$

and the moment of inertia ratio,

$$I_{\text{r}}/I_{\text{f}} = 0.5,$$

where the subscripts r and f refer to the rigid attachment and flexible beam respectively, and the other symbols are defined in *Figure* 4.3.

The results of the beam tip vibration response and the rigid shaft motion are presented in *Figures* 4.5, 4.6, and 4.7, respectively. The agreement with the analytical solution of Yigit *et al.*, for the elastic vibration is very good. However, it should be pointed out that in their formulation, they employed only a one mode approximation in the analysis. Comparison for the rigid shaft motion is not possible, since their paper did not present results for this response that is computed from the uncoupled set of equations. The rigid

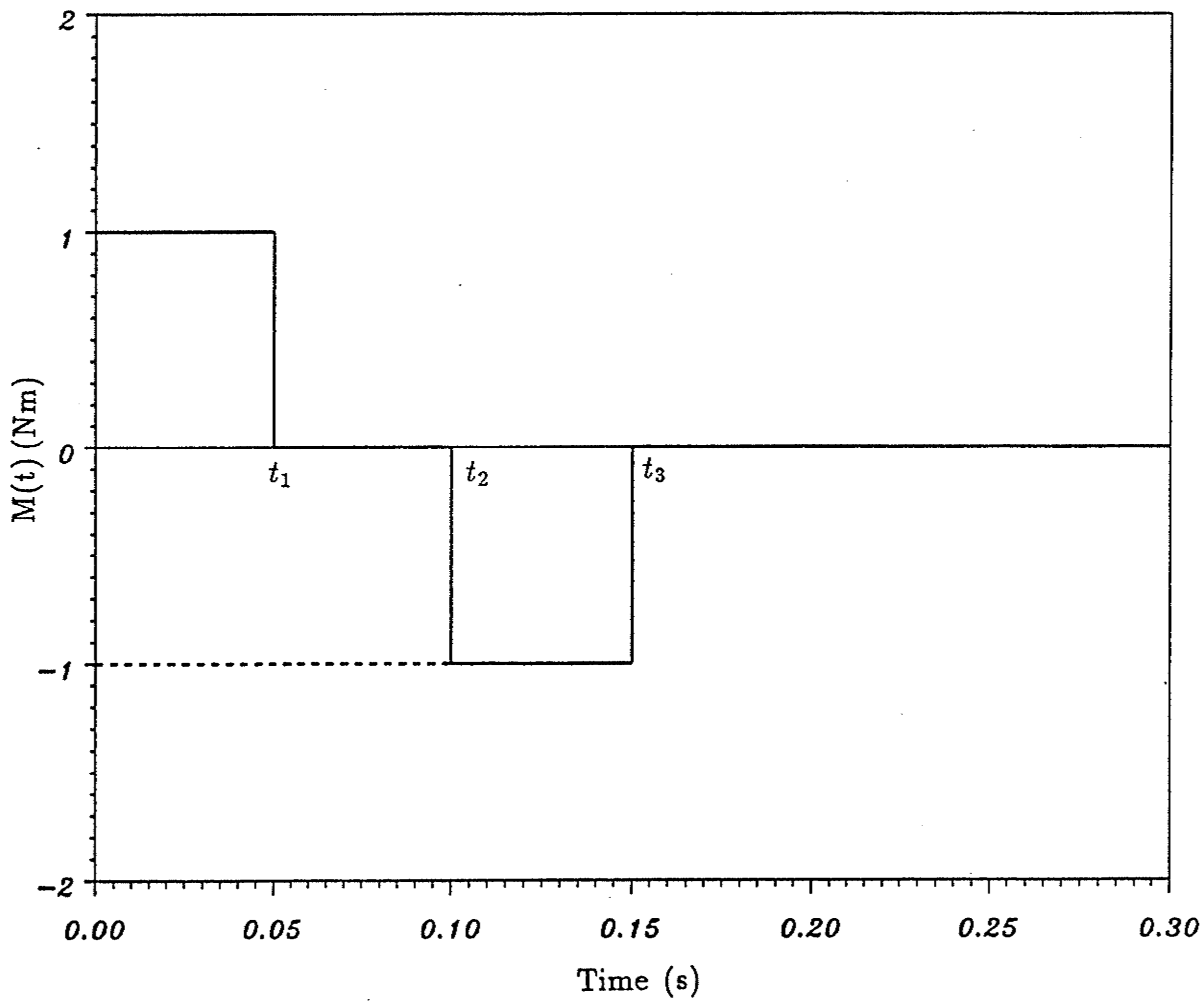


Figure 4.4: A prescribed torque pulse loading (Yigit *et al.*, 1988)

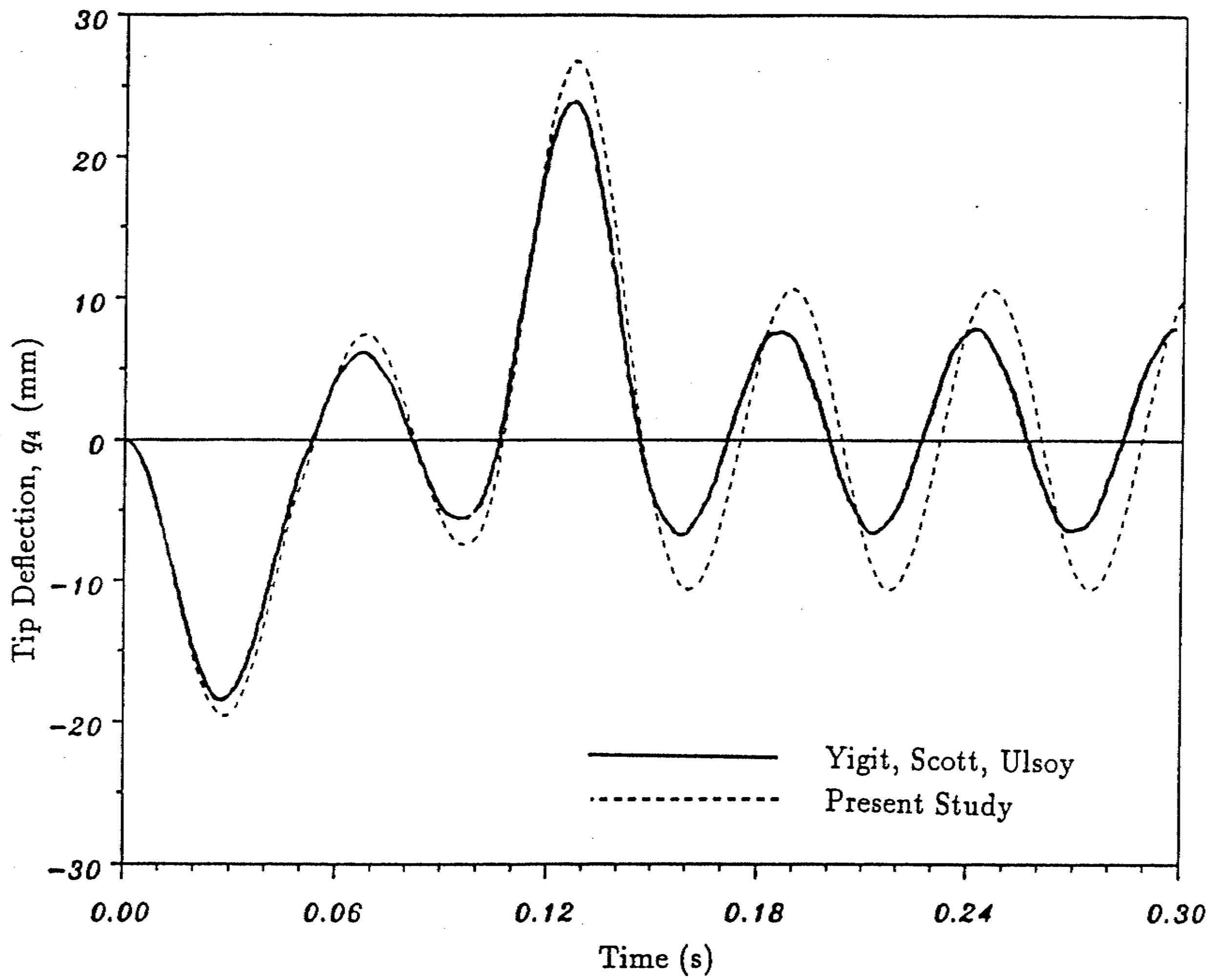


Figure 4.5: Beam tip vibration response

body motions corresponding to angular displacement and angular velocity are shown in *Figures* 4.6 and 4.7. Except for some very small oscillatory residue motion, the shaft comes to a stop at the conclusion of the torque loading.

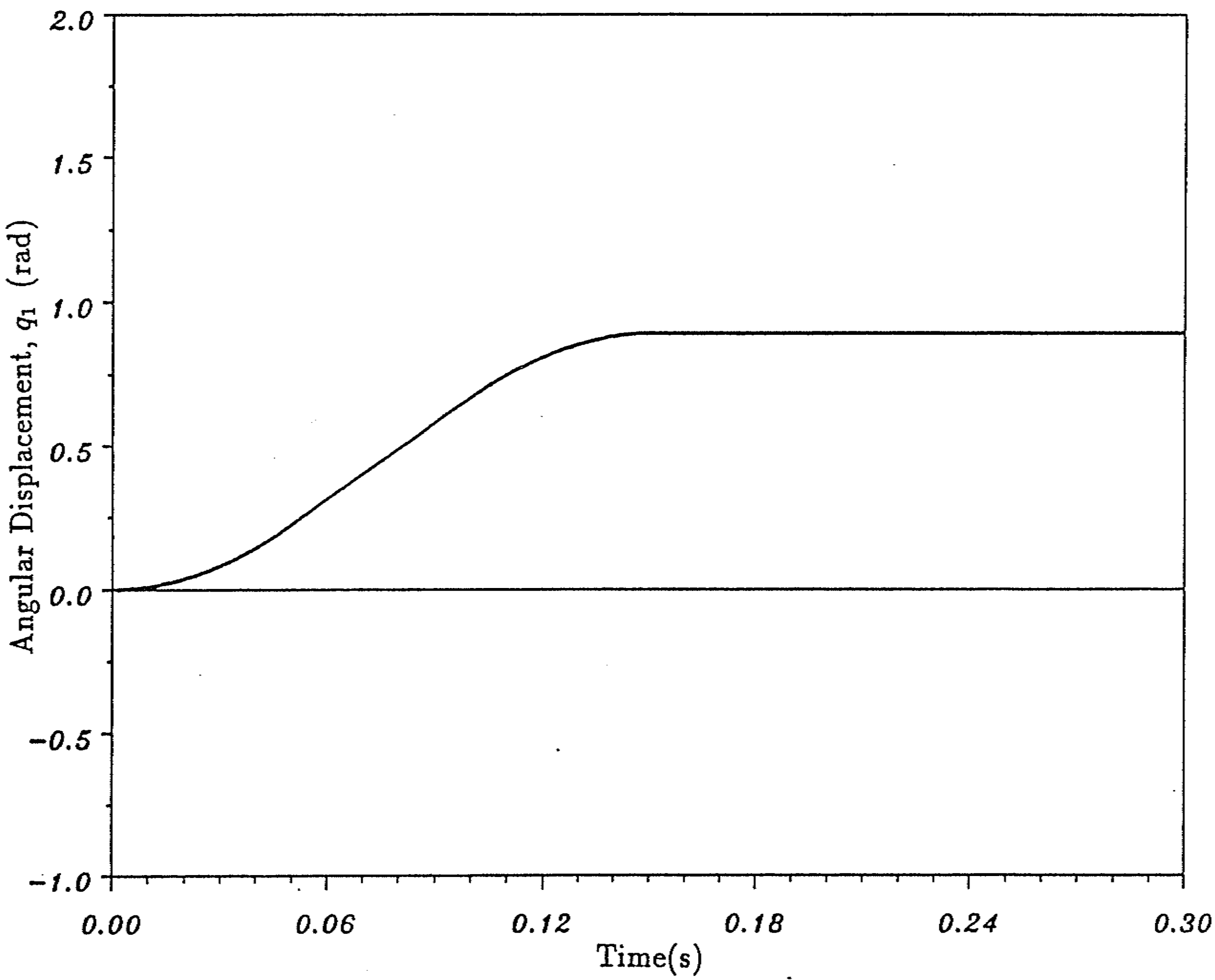


Figure 4.6: Rigid shaft motion: angular displacement

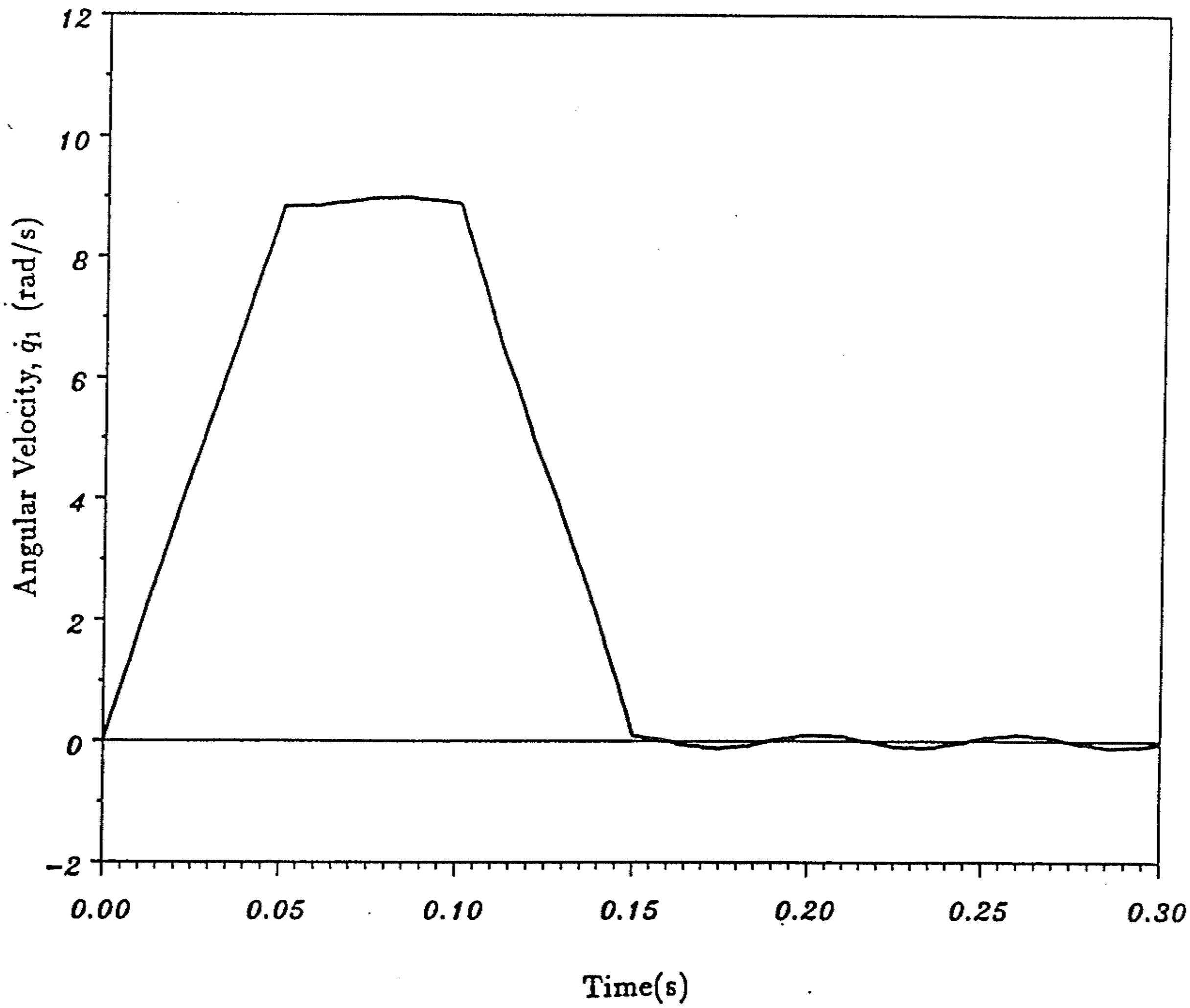


Figure 4.7: Rigid shaft motion: angular velocity

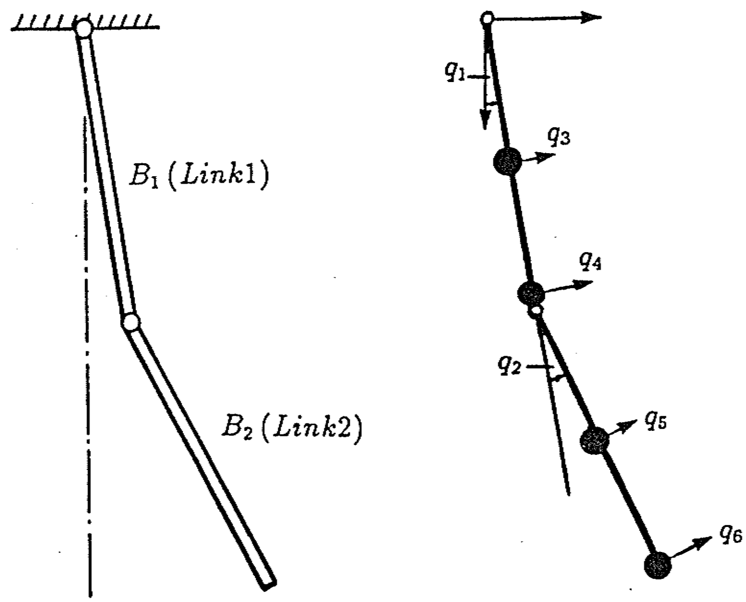


Figure 4.8: A flexible two-link manipulator

4.4 A Flexible Two-link Manipulator

In terms of speed, weight, power consumption and maneuverability, flexible manipulators are more desirable compared to their rigid-arm counterparts. However, their performance can be severely curtailed by oscillatory vibrations. Clearly, an accurate dynamic analysis of such flexible manipulator systems is required for a proper development of efficient controls to attenuate these undesirable motions.

Figure 4.8 shows a flexible two-link manipulator, in which both links are hanging freely under gravity load, with no torques applied at the joints. This example was taken from Usoro, Nadira and Mahil (1986), who used a Lagrangian finite element approach in their formulation. They presented solutions for the following initial conditions:

$$\theta_1(0) = 0^0,\ \theta_2(0) = 5^0.$$

The model parameters used were:

$$L_1 = L_2 = 1 \text{ m}$$

$$I_1 = I_2 = 5 \times 10^{-9} \text{ m}^4$$

$$m_1 = m_2 = 5 \text{ kg/m}$$

and

$$E_1 = E_2 = 2.0 \times 10^{11} \text{ N/m}^2$$

The rigid body motions corresponding to θ_1, θ_2 vibrations are depicted in *Figure* 4.9.

They show the system responses arising from an initial angular perturbation of 5^0 in θ_2. They agree very well with the results given in Usoro *et al.* The elastic body motions corresponding to flexural responses at the midpoint and at the tip of each link are also summarized in this figure. As also shown in Usoro *et al.*, two modes of vibrations are observed in each graph. A fast mode which corresponds to the high frequency vibrations, and, superimposed on these vibrations, a relatively slower oscillatory mode. The elastic motions computed here, however, do not quite agree with the results of Usoro et. al. For example, the slower oscillatory central and tip vibrations of the first link

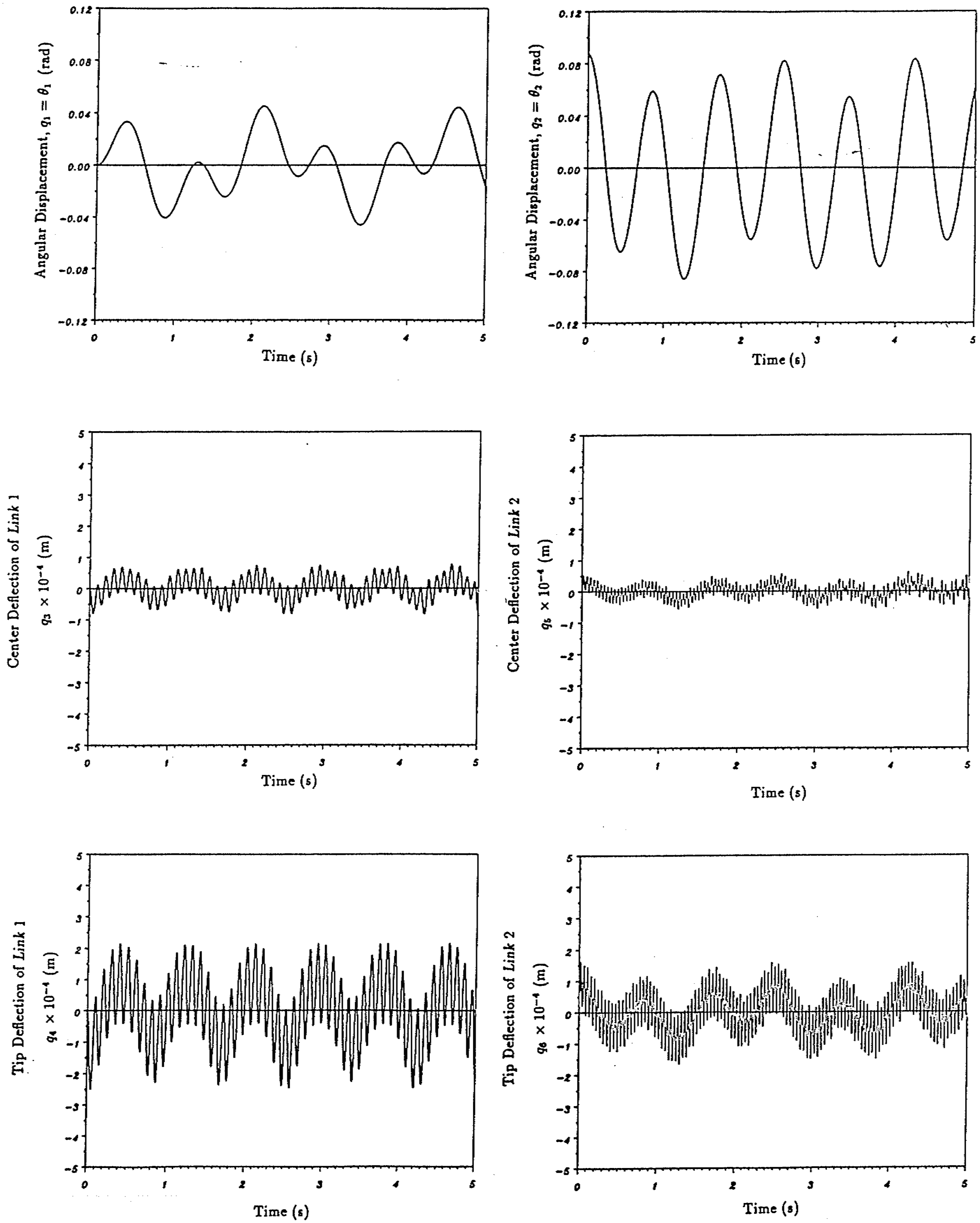


Figure 4.9: Simulation results: rigid body and flexible body motions

and second link are out-of-phase with each other, as would be expected. But in Usoro et. al.'s solutions, they are in-phase. It is felt that the results presented here are more accurate since this slower oscillatory mode is heavily influenced by the rigid body motions in the two links which, as shown here and as well as in Usoro et. al., are also out-of-phase with each other. Other discrepancies, attributable to differences in modeling, include the amplitude of the slow mode and the frequencies of the fast mode.

These simulation results indicate that a flexible manipulator exhibits high undesirable oscillatory motions, and an accurate dynamic analysis is required for a proper development of efficient controls to attenuate these vibrations.

Chapter 5

Summary and Conclusions

The dynamics of a general, flexible multibody system using a lumped mass finite element model is presented. The system topology considered here consists of an arbitrary combination of both rigid and flexible bodies, linked together by joints that permit translations and rotations, in a general tree configuration. An extension of the formulation to handle closed loop kinematic chains is also suggested. The equations of motion are derived using Kane's theory of generalized speeds, resulting in a very efficient computer oriented methodology for solving the dynamics of such large mechanical systems. For an explicit formulation of the problem, these dynamical equations are recast into a system of first-order differential equations. Three examples with known solutions were solved for comparison. The first example is an elastic simple pendulum in which analytical results are available. An exact agreement is obtained for this simple example. The second example which also possess analytical solutions, is a flexible beam firmly attached to a rotating rigid shaft. The agreement obtained here was very good. The third example is a flexible two-link manipulator, for which numerical solutions from a Lagrangian finite element formulation are available. Except for the elastic vibrations, very good agreement is obtained for the rigid body motions. It is felt that the elastic solutions computed here are more accurate than those given by the solution of Usoro et. al. As expected, a flexible manipulator exhibits highly undesirable oscillatory motions, and an accurate dynamic analysis is required for a proper development of efficient controls to attenuate these vibrations. In summary, the results from these three examples indicate the method is accurate, efficient and versatile for the analysis of a general, flexible multibody system.

Bibliography

Agrawal, O.P. and A. Shabana, A. (1986), 'Application of Deformable-Body Mean Axis to Flexible Multibody System Dynamics', *Comp. Meth. Appl. Mech. Eng.*, **56**, 217-245.

Amirouche, F.M.L. and Huston, R.L.(1988), 'Dynamics of Large Constrained Flexible Structures', *ASME J. Dynamic Systems, Measurement & Control*, **110**, 78-83.

Bakr, E.M. and Shabana, A. (1986), 'Geometrically Nonlinear Analysis of Multibody Systems', *Computers & Structures*, **23**, 739-751.

Bodley, C.S. and Park, A.C. (1972), 'The Influence of Structural Flexibility on the Dynamic Response of Spinning Spacecraft', *AIAA Paper 72-348.*

Book, W.J. (1984), 'Recursive Lagrangian Dynamics of Flexible Manipulator Arms', *Int. J. Robotics Res.*, **3**, 87-101.

Changizi and Shabana, A. (1988), 'A Recursive Formulation for the Dynamic Analysis of Open Loop Deformable Multibody Systems', *ASME J. Appl. Mech.*, **55**, 687-693.

Cleghorn, W.L., Fenton R.G. and Tabarrok, B. (1981), 'Finite Element Analysis of High-Speed Flexible Mechanisms', *J. Mech. Mach. Theory*, **16**, 406-424.

Dubowsky, S. and Maatuk, J. (1975), 'The Dynamics Analysis of Spatial Mechanisms', *Proc. IV World Congr. Mach. Mech.*, University of Newcastle upon Tyne, England, 927-932.

Gupta, V.K. (1972), 'Dynamics Analysis of Multi-Rigid-Body Systems', *ASME J. Engineering for Industry*, **96**, 886-892.

Hablani, H.B. (1982), 'Constrained and Unconstrained Modes; Some Modeling Aspects of Flexible Spacecrafts', *J. Guidance, Control & Dynamics*, **5**, 164-173.

Han, R.P.S. and Zhao, Z.C. (1989), 'Vibration of Multibody System', *Proceedings of International Conference of Sound and Vibration*, Singapore.

Han, R.P.S. and Zhao, Z.C. (1990), 'Dynamics of General Flexible Multibody Systems', *Int. J. Numerical Methods in Engineering*, **30**, 77-97.

Han, R.P.S. and Zhao, Z.C. (1990), 'General Flexible Multibody Dynamics: Open and Closed Loop Systems', *Proceedings of International Conference of Vibration*, China.

Ho, J.Y.L. (1977), 'Direct Path Method for Flexible Multibody Spacecraft Dynamics', *J. Spacecraft & Rockets*, **14**, 102-110.

Ho, J.Y.L. and Herber, D.R. (1985), 'Development of Dynamics and Control Simulation of Large Flexible Space Systems', *J. Guidance, Control & Dynamics*, **8**, 374-383.

Hooker, W.W. and Margulies, G. (1965), 'Dynamical Attitude Equations for an n-Body Satellite', *J. Astronautical Sciences*, **12**, 123-128.

Hooker, W.W. (1975), 'Equations of Motion for Interconnected Rigid and Elastic Bodies', *Celestial Mechanics*, **11**, 337-359.

Hughes, P.C. (1979), 'Dynamics of a Chain of Flexible Bodies', *J. Astronautical Sciences*, **XXVII**, 359-380.

Huston, R.L., Passerello, C.E. and Harlow, M.W. (1978), 'Dynamics of Multirigid-Body Systems', *ASME J. Appl. Mech.*, **45**, 889-894 (1978).

Huston, R.L. and Passerello, C. (1979), 'On Multi-Rigid-Body System Dynamics', *Computers & Structures*, **10**, 439-446.

Huston, R.L. and Passerello, C. (1980), 'Multibody Structured Dynamics Including Translation Between the Bodies', *Computers & Structures*, **12**, 713-720.

Huston, R.L. (1981), 'Multi-Body Dynamics Including the Effects of Flexibility and Compliance, *Computers & Structures*, **14**, 443-451.

Jandrasits, W.G. and Lowen, G.G. (1979), 'The Elastic-Dynamic Behavior of a Counterweighted Rocker Link with an Overhanging Endmass in a Four-Bar Linkage, Part I: Theory', *ASME J. Mech. Design*, **101**, 77-88.

Jerkovsky, W. (1978), 'The Structure of Multibody Dynamics Equations', *J. Guidance & Control*, **1**, 173-182.

Kane, T.R. and Wang, C.F. (1965), 'On the Derivation of *Equations* of Motion', *SIAM J. Applied Mathematics*, **13**.

Kane, T.R. (1968) 'Dynamics', Holt, Rinehart & Winston, New York.

Kane, T.R. and Kahn, M.E. (1968), 'On a Class of Two-Degree-of-Freedom Oscillations', *ASME J. Appl. Mech.*, **35**, 547-552.

Kane, T.R. and Levinson, D.A. (1980), 'Formulation of Equations of Motion for Complex Spacecraft', *J. Guidance & Control*, **3**, 99-112.

Kane, T.R., Likins, P.W. and Levinson, D.A. (1983a), *Spacecraft Dynamics*, New York, McGraw-Hill.

Kane, T.R. and Levinson, D.A. (1983b), 'Multibody Dynamics', *ASME J. Appl. Mech.*, **50**, 1071-1078.

Kane, T.R. and Levinson, D.A. (1983c), 'The Use of Kane's Dynamical Equations in Robotics', *Int. J. of Robotics Research*, **3**, 3-21.

Kane, T.R. and Ryan, R.R. (1987), 'Dynamics of a Cantiliever Beam Attached to a Moving Base', *J. Guidance, Control & Dynamics*, **10**, 139-151.

Kim, S.S. and Vanderploeg, M.J. (1986), 'A General and Efficient Method for Dynamic Analysis of Mechanical Systems using Velocity Transformations', *ASME J. Mech. Trans. & Auto. Des.*, **108**, 176-184.

Kim, S.S. and Haug, E.J. (1988), 'A Recursive Formulation for Flexible Multibody Dynamics, Part I: Open-Loop Systems', *Comp. Meth. Appl. Mech. Eng.*, **71**, 293-314.

Kohli, D. and Sandor, G.N. (1975), 'Lumped Parameter Approach for Kineto-Elastodynamic Analysis of Elastic Spatial Mechanisms', *Proc. IV World Congr. Mach. Mech.*, University of Newcastle upon Tyne, England.

Kohli, D., Hunter, D. and Sandor, G.N. (1977), 'Elastodynamic Analysis of a Completely Elastic Systems', *ASME J. Eng. Ind.*, **99**, 604-609.

Kulla, P. (1972), Dynamics of Spinning Bodies Containing Elastic Rods', *J. Spacecraft & Rockets*, **9**, 246-253.

Lee, J.D. and Wang, B. (1988), 'Dynamic Equations for a Two-Link Flexible Robot Arm', *Computers & Structures*, **29**, 469-477.

Levinson, D.A. (1977), ' Equations of Motion for Multiple-Rigid Body Systems via Symbolic Manipulation', *J. Spacecraft & Rockets*, **14**, 479-487.

Likins, P.W. (1970), 'Dynamic Analysis of a System of Hinge-Connected Rigid Bodies with Non-Rigid Appendages', *Int. J. Solids & Structures*, **9**, 1473-1487.

Likins, P.W., Barbera, F.J. and Baddeley, V. (1973), 'Mathematical Modeling of Spinning Elastic Bodies for Modal Analysis', *AIAA J.*, **11**, 1251-1258.

Likins, P.W. (1974), 'Analytical Dynamics and Non-Rigid Spacecraft Simulation', JPL TR 32-1593, California Institute of Technology.

Likins, P.W. (1975), 'Point-Connected Rigid Bodies in a Topological Tree', *Celestial Mechanics*, **11**, 301-317.

Low, K.H. (1987), 'A Systematic Formulation of Dynamic Equations for Robot Manipulators with Elastic Links', *J. Robotic Systems*, **4**, 435-456.

Lowen, G.G. and Chassapis, C. (1986), 'The Elastic Behavior of Linkages: an Update', *Mechanism and Machine Theory*, **21**, 33-42.

Magnus, K. (1978), 'Dynamics of Multibody Systems', Springer-Verlag, Berlin.

Midha, A., Erdman, A.G. and Frohrib, D.A. (1978), 'Finite Element Approach to Mathematical Modeling of High Speed Elastic Linkages', *J. Mech. Mach. Theory*, **13**, 603-618.

Roberson, R.E. and Wittenburg, J. (1966), 'A Dynamical Formalism for an Arbitrary Number of Interconnected Rigid Bodies, with Reference to the Problem of Satellite Attitude Control', *Proc.Third Congress IFAC*, Butterworth, London, 46D.1-46D.8.

Roberson, R.E. (1972), 'A Form of the Translational Dynamical Equations for Relative Motion in Systems of Many Non-Rigid Bodies', *Acta Mechanica*, **14**, 297-308.

Sadler, J.P. and Sandor, G.N. (1973), 'A lumped Parameter Approach to Vibration and Stress Analysis of Elastic Linkages', *ASME J. Eng. Ind.*, **95**, 549-557.

Sadler, J.P. (1975), 'On the Analytical lumped-Mass Model of an Elastic Four-Bar Mechanism', *ASME J. Eng. Ind.*, **97**, 561-565.

Shabana, A.A. (1985), 'Automated Analysis of Constrained Systems of Rigid and Flexible Bodies', *ASME J. Vibr. Acou. Str. & Rel. Des.*, **107**, 431-439.

Shabana, A. (1986), 'Dynamics of Inertia-Variant Flexible Systems using Experimentally Identified Parameters', *ASME J. Mech. Trans. & Auto. Des.*, **108**, 358-36.

Singh, R.P., VanderVoort, R.J. and Likins, P.W. (1985), 'Dynamics of Flexible Bodies in Tree Topology - A Computer-Oriented Approach', *AIAA J. Guidance*, **8**, 584-590.

Singh, R.P. and Likins, P.W. (1985), 'Singular Value Decomposition for Constrained Dynamical Systems', *ASME J. Appl. Mech.*, **52**, 943-948.

Sunada, W.H. and Dubowsky, S. (1983), 'On the Dynamic Analysis and Behavior of Industrial Robotic Manipulator with Elastic Members', *ASME J. Mech. Trans. & Auto. Des.*, **105**, 42-51.

Thompson, B.S. and Barr, A.D.S. (1976), 'A Variational Principle for the Elastodynamic Motion of Planar Linkages', *ASME J. Eng. Ind.*, **98**, 541-548.

Turcic, D.A. and Midha, A. (1984), 'Generalized Equations of Motion for the Dynamic Analysis of Elastic Mechanism Systems', *J. Dyn. Sys. Meas. Cont.*, **106**, 243-248.

Usoro, P.B., Nadira, R and Mahil, S.S. (1986), 'A Finite Element/Lagrange Approach to Modelling Lightweight Flexible Manipulators', *ASME J. Dyn. Sys. Meas. Cont.*, **108**, 198-205.

Yigit, A., Scott, R.A. and Ulsoy, A.G. (1988), 'Flexural Motion of a Radially Rotating Beam Attached to a Rigid Body', *J. Sound & Vibrations*, **121**, 201-210.

Wampler, C., Buffinton, K. and Shu-hui, J. (1985), 'Formulation of Equations of Motion for Systems Subject to Constraints', *ASME J. Appl. Mech.*, **52**, 465-470.

Wang, J.T. and Huston, R.L. (1987), 'Kane's Equations with Undetermined Multipliers - Application to Constrained Multibody Systems', *ASME J. Appl. Mech.*, **54**, 424-429 (1987).

Wittenburg, J. (1977), 'Dynamics of Systems of Rigid Bodies', *Teubner, Leitfäden der Angewandten Math. u. Mech.*, **33**, 221-222.

Appendix A

Nomenclature

$\mathbf{a}_k^i$	Acceleration vector of node i in body B_k
$\mathbf{A}_{mn}$, B_m	Functions of generalized coordinates and time
$\mathbf{b}_{mp}$	Generalized mass matrix
F_m, F_m^*	Generalized active and inertia forces for unconstrained systems
F'_m, F'^*_m	Generalized active and inertia forces for constrained systems
$\mathbf{F}_k$	Total working forces in body B_k
f_m	Generalized load vector
$\mathcal{I}_k^{O_k}$	Inertia dyadic of rigid body
$L(k)$	Body connection array
m_k, m_k^i	Mass of body B_k and node i in body B_k respectively
$M_k^{O_k}$	Moment of working forces about hinge point, O_k in body B_k
$\mathcal{M}_k$, $\mathcal{N}_k$	Inertia dyadics of flexible body
N	Number of bodies in the system

NC	Number of independent constraints
ND_k	Number of deformational degrees of freedom in body B_k
NN_k	Nodal degrees of freedom in body B_k
NN_k^T, NN_k^R	Nodal translational and rotational degrees of freedom in body B_k
NP_k	Number of nodes in body B_k
NR_k	Number of rotational degrees of freedom in body B_k
NS	Total number of degrees of freedom for unconstrained systems
NS'	Total number of degrees of freedom for constrained systems
NT_k	Number of translational degrees of freedom in body B_k
P_k^i, Q_k^i	External and internal forces at node i in body B_k
q_m, $\dot{q}_m$, $\ddot{q}_m$	Generalized coordinates, speeds and accelerations
$\mathbf{r}_k^i$	Absolute position vector of node i in body B_k after deformation
$\mathbf{r}_k^{O_k}$	Absolute position vector of hinge point O_k in body B_k
$\mathbf{x}_k^i$	Relative position vector from the frame at hinge point O^k to the i^{th} node
${}^T\mathbf{u}_k^i$, ${}^R\mathbf{u}_k^i$	Elastic translational and rotational nodal displacement vectors
${}^T\mathbf{U}_{km}^i$, ${}^R\mathbf{U}_{km}^i$	Partial elastic translational and rotational vectors of node i in body B_k
$\mathbf{v}_k^i$	Linear velocity of node i in body B_k
$\mathbf{v}_k^{O_k}$	Linear velocity of hinge point O_k in body B_k

$\mathbf{V}_{km}^{O_k}$, $\mathbf{V}_{km}^{i}$	Partial velocity in body B_k for unconstrained systems
$\mathbf{V}_{km}^{\prime i}$	Partial velocity in body B_k for constrained systems
y_m	Vector of generalized speeds
α_k^i	Angular acceleration vector of node i in body B_k
ω_k^i	Angular velocity vector of node i in body B_k
$\mathbf{\Omega}_{km}$	Partial angular velocity in body B_k
φ_k	Relative orientation vector of the k^{th} body
ξ_k	Relative translational displacement vector of the k^{th} body

Appendix B

Explicit Expressions for the Generalized Mass Matrix and Generalized Load Vector

The generalized mass matrix is obtained by substituting *Equation* (18) into *Equation* (25) and simplifying yields,

$$\begin{aligned}
\mathrm{b}_{mp} \;=\; & \sum_{k=1}^{N} \left[m_k \mathbf{V}_{kp}^{O_k} \bullet \mathbf{V}_{km}^{O_k} + \left(\mathcal{I}_k^{O_k} + \mathcal{N}_k + \mathcal{M}_k \right) \bullet \Omega_{kp} \bullet \Omega_{km} \right] \\
& + \sum_{k=1}^{N} \sum_{i=1}^{NP_k} m_k^i \left[\left(\mathbf{x}_k^i + {}^T\mathbf{u}_k^i \right) \times \left(\mathbf{V}_{kp}^{O_k} + {}^T\mathbf{U}_{kp}^i \right) \bullet \Omega_{km} \right. \\
& + \left(\mathbf{x}_k^i + {}^T\mathbf{u}_k^i \right) \times \left(\mathbf{V}_{km}^{O_k} + {}^T\mathbf{U}_{km}^i \right) \bullet \Omega_{kp} \\
& + \mathbf{V}_{km}^{O_k} \bullet {}^T\mathbf{U}_{kp}^i + \mathbf{V}_{kp}^{O_k} \bullet {}^T\mathbf{U}_{km}^i \\
& \left. + {}^T\mathbf{u}_k^i \times \left(\Omega_{kp} \times {}^T\mathbf{u}_k^i \right) \bullet \Omega_{km} + {}^T\mathbf{U}_{km}^i \bullet {}^T\mathbf{U}_{kp}^i \right]
\end{aligned} \tag{B.1}$$

where $\mathcal{I}_k^{O_k}$ is the inertia dyadic of the rigid body, and $\mathcal{N}_k$, $\mathcal{M}_k$ are the inertia dyadics of the flexible body. These dyadic quantities are given by,

$$
\begin{aligned}
\mathcal{I}_k^{O_k} &= \sum_{i=1}^{NP_k} m_k^i \left(\mathbf{x}_k^i \bullet \mathbf{x}_k^i \, \mathbf{U} - \mathbf{x}_k^i \, \mathbf{x}_k^i \right) \\
\mathcal{N}_k &= \sum_{i=1}^{NP_k} m_k^i \left(\mathbf{x}_k^i \bullet {}^T\mathbf{u}_k^i \, \mathbf{U} - \mathbf{x}_k^i \, {}^T\mathbf{u}_k^i \right) \\
\mathcal{M}_k &= \sum_{i=1}^{NP_k} m_k^i \left({}^T\mathbf{u}_k^i \bullet \mathbf{x}_k^i \, \mathbf{U} - {}^T\mathbf{u}_k^i \, \mathbf{x}_k^i \right)
\end{aligned}
\tag{B.2}
$$

in which $\mathbf{U}$ is the unit dyadic.

Equations (2.40,3.13,3.15) are used to derive the generalized load vector, f_m. which after simplifications becomes:

$$
f_m = F_m - f_m^* \tag{B.3}
$$

where f_m^* is defined as,

$$
\begin{aligned}
f_m^* = & \sum_{k=1}^{N} \left\{ m_k \left(\sum_{p=1}^{NS} \dot{\mathbf{V}}_{kp}^{O_k} \, \dot{q}_p \right) \bullet \mathbf{V}_{km}^{O_k} + \left(\mathcal{I}_k^{O_k} + \mathcal{N}_k + \mathcal{M}_k \right) \bullet \left(\sum_{p=1}^{NS} \dot{\Omega}_{kp} \, \dot{q}_p \right) \bullet \Omega_{km} \right. \\
& \left. + \left(\sum_{p=1}^{NS} \Omega_{kp} \, \dot{q}_p \right) \times \left[\left(\mathcal{I}_k^{O_k} + \mathcal{N}_k + \mathcal{M}_k \right) \bullet \left(\sum_{p=1}^{NS} \Omega_{kp} \, \dot{q}_p \right) \right] \bullet \Omega_{km} \right\} \\
& + \sum_{k=1}^{N} \sum_{i=1}^{NP_k} \left\{ - \left(\mathbf{x}_k^i + {}^T\mathbf{u}_k^i \right) \times \left(\sum_{p=1}^{NS} \dot{\Omega}_{kp} \, \dot{q}_p \right) \bullet \mathbf{V}_{km}^{O_k} \right.
\end{aligned}
$$

$$
\begin{aligned}
&+\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\times\left[\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\times\left(\mathbf{x}_k^i+{}^{T}\mathbf{u}_k^i\right)\right]\bullet\mathbf{V}_{km}^{O_k}\\
&+2\left[\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\times\left(\sum_{p=1}^{NS}{}^{T}\mathbf{U}_{kp}^i\,\dot{q}_p\right)\right]\bullet\mathbf{V}_{km}^{O_k}\\
&+2\left(\mathbf{x}_k^i+{}^{T}\mathbf{u}_k^i\right)\times\left[\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\times\left(\sum_{p=1}^{NS}{}^{T}\mathbf{U}_{kp}^i\,\dot{q}_p\right)\right]\bullet\Omega_{km}\\
&+{}^{T}\mathbf{u}_k^i\times\left[\left(\sum_{p=1}^{NS}\dot{\Omega}_{kp}\,\dot{q}_p\right)\times{}^{T}\mathbf{u}_k^i+\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\times\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\times{}^{T}\mathbf{u}_k^i\right]\bullet\Omega_{km}\\
&+\left(\mathbf{x}_k^i+{}^{T}\mathbf{u}_k^i\right)\times\left(\sum_{p=1}^{NS}\dot{\mathbf{V}}_{kp}^{O_k}\,\dot{q}_p\right)\bullet\Omega_{km}+\left(\sum_{p=1}^{NS}\dot{\mathbf{V}}_{kp}^{O_k}\,\dot{q}_p\right)\bullet{}^{T}\mathbf{U}_{km}^i\\
&+\left(\sum_{p=1}^{NS}\dot{\Omega}_{kp}\,\dot{q}_p\right)\bullet\left(\mathbf{x}_k^i+{}^{T}\mathbf{u}_k^i\right)\times{}^{T}\mathbf{U}_{km}^i\\
&+2\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\bullet\left[\left(\sum_{p=1}^{NS}{}^{T}\mathbf{U}_{kp}^i\,\dot{q}_p\right)\times{}^{T}\mathbf{U}_{km}^i\right]\\
&+\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\times\left[\left(\sum_{p=1}^{NS}\Omega_{kp}\,\dot{q}_p\right)\times\left(\mathbf{x}_k^i+{}^{T}\mathbf{u}_k^i\right)\right]\bullet{}^{T}\mathbf{U}_{\mathbf{km}}^{\mathbf{i}}\Bigg\}
\end{aligned}
\qquad (B.4)
$$

Appendix C

Program DAFMS for Dynamical Analysis of a General Flexible Multibody System

PROGRAM DAFMS

DYNAMICAL ANALYSIS OF FLEXIBLE MULTIBODY SYSTEMS

(3-D Formulation)

Programed by

ZHICHENG ZHAO

September, 1991

UNIVERSITY OF MANITOBA

Winnipeg, Manitoba, Canada

```
C
C     MAIN PROGRAM DAFMS
C
      COMMON/ROOM/MR(100000)
      COMMON/PSIZE/MAX
      COMMON/QSIZE/MAXRM
      OPEN(UNIT=5,FILE='FMBD.DAT')
      OPEN(UNIT=6,FILE='FMBD.OUT')
      MAX=100000
      MAXRM=20000
      READ (5,1000) N,NDM
      WRITE(6,2000) N,NDM
      NDMM=3
      IF(NDM.LT.3) NDMM=1
      N1=1
      N2=N1+N
      N3=N2+N*4
      N4=N3+N*4
      N5=N4+N
      N6=N5+N
      CALL QCONTR(MR(N2),MR(N3),MR(N4),MR(N5)
     1             ,N,NDM,NDMM)
 1000 FORMAT(2I5)
```

```
 2000 FORMAT(5X,'==N==NDM==',2I5)
      STOP
      END

      SUBROUTINE QCONTR(NT,NR,NUMNP,NDF,N,NDM,NDMM)
      LOGICAL PCOMP
      EXTERNAL FCN,FCNJ
C..............................FCM,FCMJ
      COMMON M(20000)
      COMMON/ROOM/MR(100000)
      COMMON/QDATA/Q0,QHEAD(20),IPR
      COMMON/POINT0/N1,N2,N3,N4,N5,N6
      COMMON/POINTF/NF1,NF2,NF3,NF4,NF5,NF6,NF7,NF8
      COMMON/POINT1/N9,N10,N11,N12,N13,N14,N15,N16,N17,N18,N19,N20
      COMMON/POINT2/N21,N22,N23,N24,N25,NXL,N26,N27,N28,N29
      COMMON/POINT3/NUI,N31,N32,NYL,NYI,NYD,ND1,ND2,ND3
      COMMON/POINT4/N41,N42,N43,N44,N45,N46,N47
      COMMON/POINT5/N51,N52,N53,N54,N55
      COMMON/POINT6/N61,N62,N63,N64
      COMMON/POINT7/N71,N72,N73,N74,N75
      DIMENSION QTITL(20),QWD(3),NT(N,1),NR(N,1),NUMNP(1),NDF(1)
      DATA QWD/4HDFMB,4HSTAR,4HSTOP/
C
C READ A CARD AND COMPARE FIRST 4 COLUMS WITH MACRO LIST
C
 1    READ(5,1000) QTITL
      IF(PCOMP(QTITL(1),QWD(1))) GOTO 100
      IF(PCOMP(QTITL(1),QWD(2))) GOTO 200
      IF(PCOMP(QTITL(1),QWD(3))) RETURN
      GO TO 1
C
C READ AND PRINT CONTROL INFORMATION
C
 100  DO 101 I=1,20
        QHEAD(I)=QTITL(I)
 101  CONTINUE
      WRITE(6,2000)  QHEAD
      DO 103 K=1,N
       READ(5,1001) KK,(NT(K,I),I=1,4)
```

```
         WRITE(6,2001)  KK,(NT(K,I),I=1,4)
  103   CONTINUE
        DO 105 K=1,N
         READ(5,1001) KK,(NR(K,I),I=1,4)
         WRITE(6,2002) KK,(NR(K,I),I=1,4)
  105   CONTINUE
        READ(5,1001) (NUMNP(K),K=1,N)
        READ(5,1001) (NDF(K),K=1,N)
        WRITE(6,2003)  (NUMNP(K),K=1,N)
        WRITE(6,2004)  (NDF(K),K=1,N)
        READ(5,1007) IT
        READ(5,1008) H
        WRITE(6,2007) IT
        WRITE(6,2008) H
C
C SET POINTERS FOR ALLOCATION OF DATA ARRAYS
C
        NS=0
        DO 110 K=1,N
          NS=NS+NT(K,1)+NR(K,1)+NUMNP(K)*NDF(K)
  110   CONTINUE
        WRITE(6,2005) NS
        NPMAX=50
        NMMAX=100
        NDFMAX=3
        N1 =1
        N2 =N1 +N
        N3 =N2 +N*4
        N4 =N3 +N*4
        N5 =N4 +N
        N6 =N5 +N
C.............................................
        NF1=N6 +N+MOD(N6+N,IPR)
        NF2=NF1+N*NPMAX*3*IPR
        NF3=NF2+N*NPMAX*3*IPR
        NF4=NF3+N*NPMAX*IPR
        NF5=NF4+N*IPR
        NF6=NF5+N*NPMAX*NDFMAX*IPR
        NF7=NF6+NMMAX*IPR
```

```
      NF8=NF7+N*NMMAX*NMMAX*IPR
C...............FEAP DATA INPUT
      CALL DFIN(MR(N4),MR(N5),MR(NF1),MR(NF2),MR(NF3),MR(NF4),MR(NF5),
     1          MR(NF6),MR(NF7),MR(NF8),N,NS,NPMAX,NMMAX)
      N9 =NF8+(NMMAX*(NMMAX+1)/2)*IPR
      N10=N9 +N*IPR
      N11=N10+N*3*3*IPR
      N12=N11+N*3*IPR
      N13=N12+N*3*IPR
      N14=N13+N*4*IPR
      N15=N14+N*4*IPR
      N16=N15+N*3*IPR
      N17=N16+N*3*IPR
      N18=N17+N*3*3*IPR
      N19=N18+N*3*3*IPR
      N20=N19+N*3*3*IPR
      N21=N20+N*3*NS*IPR
      N22=N21+N*3*NS*IPR
      N23=N22+N*3*NS*IPR
      N24=N23+N*3*3*NS*IPR
      N25=N24+N*3*NS*IPR
      NXL=N25+N*3*IPR
      N26=NXL+NS*IPR
      N27=N26+N*NPMAX*3*NS*IPR
      N28=N27+3*NS*IPR
      N29=N28+N*3*IPR
      NUI=N29+3*NS*IPR
      N31=NUI+N*NPMAX*3
      N32=N31+N*3*3*IPR
      NYL=N32+N*3*3*IPR
      NYI=NYL+2*NS*IPR
      NYD=NYI+2*NS*IPR
C
C     CALL SYSTEM DISCRIPTION INPUT SUBROUTINE TO READ AND PRINT
C     ALL GEOMETRIC AND KINEMETIC INITIAL DATA.
C
      ND1=NYD+NS*IPR
      ND2=ND1+N*IPR
      ND3=ND2+N*IPR
```

```
      CALL TSYSIN(MR(N1),MR(N6),MR(N9),MR(N10),MR(N11),MR(N12),
     1  MR(N13),MR(N14),MR(NXL),MR(NYL),MR(N15),MR(N16),MR(N28),
     1  MR(ND1),MR(ND2),MR(ND3),N,NS,NDM,NDMM)
      GO TO 1
 200  CONTINUE
      N41=ND3+N*IPR
      N42=N41+N*3*3*IPR
      N43=N42+N*3*3*IPR
      N44=N43+N*3*3*IPR
      N45=N44+N*3*NS*IPR
      N46=N45+N*3*3*NS*IPR
      N47=N46+N*3*IPR
      N51=N47+N*NPMAX*3*NS*IPR
      N52=N51+NS*NS*IPR
      N53=N52+NS*NS*IPR
      N54=N53+NS*IPR
      N55=N54+3*NS*IPR

      N61=N55+3*NS*IPR
      N62=N61+NS*IPR
      N63=N62+NS*IPR
      N64=N63+NS*IPR
      N71=N64+NS*IPR
      N72=N71+NS*IPR
      N73=N72+NS*IPR
      N74=N73+NS*IPR
      N75=N74+NS*IPR
      CALL CDGEAR(MR(NYL),MR(NYI),N,NS)
      GO TO 1
C
C     INPUT/OUTPUT FORMATS
C
 1000 FORMAT (20A4)
 1001 FORMAT (10I5)
 1007 FORMAT (I5)
 1008 FORMAT (F10.5)
 2000 FORMAT (20A4)
 2001 FORMAT (5X,30HNUMBER OF DOF OF TRANSLATION =,10I5)
 2002 FORMAT (5X,30HNUMBER OF DOF OF ROTATION    =,10I5)
```

```
 2003 FORMAT (5X,30HNUMBER OF NODES IN BODY K     =,10I5)
 2004 FORMAT (5X,30HNUMBER OF DOF OF NODE IN BK  =,10I5)
 2005 FORMAT (5X,30HDOF OF THE SYSTEM             =,10I5)
 2006 FORMAT (5X,30HPOINTERS                      =,5I10)
 2007 FORMAT (5X,30HAMOUNT OF INTEGRAL STEP       =,  I5)
 2008 FORMAT (5X,30HLENGTH OF TIME OF ONE STEP    =,F10.5)
 2011 FORMAT (/5X,'==NUMBER OF INTEGRAL STEP ==',I5,'=='/)
 2015 FORMAT (2X,'===X===',14F8.3)
 5001 FORMAT (2X,'T-Y',7F12.5)
 5002 FORMAT (2X,'T-X',7F12.5)
 5003 FORMAT (2X,'-TOL,XEND,H,T,METH,MITER,INDEX',4F10.5,3I5)
 5004 FORMAT (2X,'**Y**',10F10.5)
      END


      SUBROUTINE TSYSIN(LC,NP,BM,BII,XI,PH,EPT,EPL,X,Y,HF,HM,RC,
     1                  IDS,F1,TIME,N,NS,NDM,NDMM)
      LOGICAL PRT,ERR,PCOMP
      COMMON/QDATA/Q0,QHEAD(20),IPR
      DIMENSION LC(1),BM(1),BII(N,3,1),XI(N,1),NP(1),PH(N,1),EPT(N,1),
     1  EPL(N,1),X(1),Y(1),HF(N,1),HM(N,1),RC(N,1),IDS(1),F1(1),
     1  TIME(1),WD(17)
      DATA WD/4HCONN,4HMASS,4HINER,4HXI   ,4HQ(k),4HNOFP,4HEULT,
     1       4HEULL,4HX(L),4HY(L),4HFORC,4HTOQU,
     1       4HEND ,4HPRIN,4HNOPR,4HRC(K,4HDESN/  ,LIST/17/,PRT/.TRUE./
C     *************************************************************
C     *                                                           *
C     *          READ MULTIBODY SYSTEM INFORMATION                *
C     *                                                           *
C     *************************************************************
C
C INITIALIZE ARRAYS
C
      ERR=.FALSE.
 10   READ(5,1000) CC
      DO 20 I=1,LIST
        IF(PCOMP(CC,WD(I))) GO TO 30
 20   CONTINUE
      GO TO 10
```

```
 30   GO TO (1,2,3,4,5,6,7,8,9,11,12,13,14,15,16,17,18),I
C
C CONNECTION ARRAY DATA INPUT
C
 1    READ(5,1001) (LC(K),K=1,N)
      IF(PRT) WRITE(6,2001) QHEAD,(LC(K),K=1,N)
      GO TO 10
C
C MASS OF BODIES DATA INPUT
C
 2    DO 201 K=1,N
       READ(5,1002) IK,BM(K)
       IF(PRT) WRITE(6,2002) IK,BM(K)
 201  CONTINUE
      GO TO 10
C
C INERTIA PROPETIESABOUT REFERENCE POINT OK DATA BII INPUT
C
 3    DO 301 K=1,N
        DO 302 I=1,3
        DO 302 M=1,3
          BII(K,I,M)=0.
 302    CONTINUE
        READ(5,1003) LK,(BII(K,I,I),I=1,3)
        IF(PRT) WRITE(6,2003) LK,(BII(K,I,I),I=1,3)
 301  CONTINUE
      GO TO 10
C
C RELATIVE POSITION VECTOR OF JOINT Pk TO JOINT Hk XI(GIVEN)
C
 4    DO 401 K=1,N
        READ(5,1004) IK,(XI(K,M),M=1,3)
        IF(PRT) WRITE(6,2004) IK,(XI(K,M),M=1,3)
 401  CONTINUE
      GO TO 10
C
C RELATIVE POSITION VECTOR OF JOINT HL(K) TO JOINT PK Qk(FIXED IN BL(k))
C
 5    DO 501 K=1,N
```

```
         READ(5,1005) JK,(PH(K,M),M=1,3)
         IF(PRT) WRITE(6,2005) JK,(PH(K,M),M=1,3)
 501   CONTINUE
       GO TO 10
C
C NUMBER OF NODE WHICH THE JOINT POINT P IS IN
C
 6     DO 601 K=1,N
         READ(5,1006) JK,NP(K)
         IF(PRT) WRITE(6,2006) JK,NP(K)
 601   CONTINUE
       GO TO 10
C
C RELATIVE EULAR PARAMETER EPT OF B(K) TO B(LC(K) INPUT
C
 7     DO 701 K=1,N
         READ(5,1007) KK,(EPT(K,M),M=1,4)
         IF(PRT) WRITE(6,2007) KK,(EPT(K,M),M=1,4)
 701   CONTINUE
       GO TO 10
C
C DEFORMATION EULAR PARAMETER EPL OF OF HINGE P TO H IN B(K)
C
 8     DO 801 K=1,N
         READ(5,1008) MK,(EPL(K,M),M=1,4)
         IF(PRT) WRITE(6,2008) MK,(EPL(K,M),M=1,4)
 801   CONTINUE
       GO TO 10
C
C INITIAL VELUE OF UNKNOWN COORDINATES X(L)
C
 9     READ(5,1009) (X(L),L=1,NS)
       IF(PRT) WRITE(6,2009) (X(L),L=1,NS)
       GO TO 10
C
C INITIAL VELUE OF UNKNOWN COORDINATES Y(L)
C
 11    NS2=NS*2
       READ(5,1011) (Y(L),L=1,NS2)
```

```
C      READ(5,1019) (Y(L),L=1,NS2)
       IF(PRT) WRITE(6,2011) (Y(L),L=1,NS2)
       GO TO 10
C
C FORCES OF HINGE DATA INPUT
C
 12    DO 1201 K=1,N
         READ(5,1012) KI,(HF(K,M),M=1,3)
         IF(PRT) WRITE(6,2012) KI,(HF(K,M),M=1,3)
 1201 CONTINUE
       GO TO 10
C
C TOQUES OF HINGE DATA INPUT
C
 13    DO 1301 K=1,N
         READ(5,1013) KJ,(HM(K,M),M=1,3)
         IF(PRT) WRITE(6,2013) KJ,(HM(K,M),M=1,3)
 1301 CONTINUE
       GO TO 10
 14    RETURN
 15    PRT= .TRUE.
       GO TO 10
 16    PRT= .FALSE.
       GO TO 10
C
C POSITION VECTER OF CENTER OF GRAVITY
C
 17    DO 1701 K=1,N
          READ(5,1017) KM,(RC(K,M),M=1,3)
          IF(PRT) WRITE(6,2017) KM,(RC(K,M),M=1,3)
 1701 CONTINUE
       GO TO 10
C
C DESIGN HM
C
 18    DO 1801 K=1,N
          READ(5,1018) KM,IDS(K),F1(K),TIME(K)
          IF(PRT) WRITE(6,2018) KM,IDS(K),F1(K),TIME(K)
 1801 CONTINUE
```

```
      GO TO 10
 1000 FORMAT(A4)
 1001 FORMAT(10I5)
 1002 FORMAT(I5,F10.5)
 1003 FORMAT(I5,3F10.5)
 1004 FORMAT(I5,3F10.5)
 1005 FORMAT(I5,3F10.5)
 1006 FORMAT(I5,I5)
 1007 FORMAT(I5,4F10.5)
 1008 FORMAT(I5,4F10.5)
 1009 FORMAT(5F10.5)
 1011 FORMAT(5F10.5)
 1019 FORMAT(5E14.6)
 1012 FORMAT(I5,3F10.5)
 1013 FORMAT(I5,3F10.5)
 1017 FORMAT(I5,3F10.5)
 1018 FORMAT(2I5,4F10.5)
 2001 FORMAT(20A4/5X,'LC==',10I5)
 2002 FORMAT(     5X,'BM==',I5,F10.5)
 2003 FORMAT(     5X,'BII=',I5,3F10.5)
 2004 FORMAT(     5X,'XI==',I5,3F10.5)
 2005 FORMAT(     5X,'PH==',I5,3F10.5)
 2006 FORMAT(     5X,'NP==',I5,I10)
 2007 FORMAT(     5X,'EPT=',I5,4F10.5)
 2008 FORMAT(     5X,'EPL=',I5,4F10.5)
 2009 FORMAT(     5X,'X(L)',5F10.5)
 2011 FORMAT(     5X,'Y(L)',5F10.5)
 2012 FORMAT(     5X,'HF==',I5,3F10.5)
 2013 FORMAT(     5X,'HM==',I5,3F10.5)
 2017 FORMAT(     5X,'RC==',I5,3F10.5)
 2018 FORMAT(     5X,'IDS',2I5,4F10.5)
      END

      SUBROUTINE DFIN(NUMNP,NDF,CO,PI,BMI,NEQ,ID,JDI,BK,BKP,
     1                 N,NS,NPMAX,NMMAX)
      DIMENSION NUMNP(1),NDF(1),CO(N,NPMAX,1),PI(N,NPMAX,1),
     1          BMI(N,1),NEQ(1),ID(1),JDI(1),BK(N,NMMAX,1),BKP(1)
C     *****************************************************************
C     *                                                               *
```

```
C     *              READ FLEXIBLE BODY INFORMATION FROM FEM            *
C     *                                                                 *
C     *****************************************************************
      DO 200 K=1,N
         INDF=1
         IF(NDF(K) .EQ. 2) INDF=2
         IF(NDF(K) .EQ. 1) INDF=3
         NPK=NUMNP(K)
         NMK=NDF(K)*NUMNP(K)
         IF(NPK .EQ. 0) GOTO 200
C
C     NODAL COORDINATE DATA INPUT
C
         DO 120 I=1,NPK
            DO 110 M=1,3
 110           CO(K,I,M)=0.
            READ(5,1001) II,(CO(K,I,M),M=2,3)
            WRITE(6,2001) K,II,(CO(K,I,M),M=1,3)
 120     CONTINUE
C
C     FORCE/DISPL DATA INPUT
C
         DO 140 I=1,NPK
            DO 130 M=1,3
 130           PI(K,I,M)=0.
            READ(5,1001) II,(PI(K,I,M),M=2,3)
            WRITE(6,2002) K,I,(PI(K,I,M),M=1,3)
 140     CONTINUE
C
C     NUMBER OF EQUATIONS INPUT
C
         READ(5,1003) NEQ(K)
         WRITE(6,2003)NEQ(K)
C
C     ID(NDF(K),NUMNP(K)) ARRAY INPUT
C
         READ(5,1004) (ID(I),I=1,NMK)
         WRITE(6,2004)(ID(I),I=1,NMK)
C
```

```
C     COMPILE POINTER JDIAG INPUT
C
         READ(5,1005) (JDI(J),J=1,NMK)
C
C     COMPILE STIFFNESS MATRIX INPUT
C
         JMAX=JDI(NMK)
         READ(5,1006) (BKP(I),I=1,JMAX)
C
C     LUMPED MASS OF NODES INPUT
C
         READ(5,1007) (BMI(K,I),I=1,NPK)
         WRITE(6,2005)(JDI(J),J=1,NMK)
         WRITE(6,2006) (BKP(I),I=1,JMAX)
         WRITE(6,2010) K
         WRITE(6,2007) (BMI(K,I),I=1,NPK)
C
C     CALCULATE GRAVITY OF THE NOTES
C
         DO 143 I=1,NPK
            PI(K,I,2)=PI(K,I,2)+BMI(K,I)*9.8
            WRITE(6,2008) K,I, (PI(K,I,M),M=1,3)
 143     CONTINUE
         DO  145 I=1,NMK
          DO 145 J=1,NMK
             BK(K,I,J)=0.
 145     CONTINUE
         BK(K,1,1)=BKP(JDI(1))
         IF(NMK .EQ. 1) GOTO 188
         DO 160 J=2,NMK
            BK(K,J,J)=BKP(JDI(J))
            L=JDI(J)-JDI(J-1)-1
            DO 150 I=1,L
               BK(K,J-I,J)=BKP(JDI(J)-I)
 150        CONTINUE
 160     CONTINUE
         NMK1=NMK-1
         DO 180 J=1,NMK1
            M=NMK-J
```

```
          DO 170 I=1,M
             BK(K,J+I,J)=BK(K,J,J+I)
 170      CONTINUE
 180    CONTINUE
 188    DO 190 I=1,NMK
          WRITE(6,2009) (BK(K,I,J),J=1,NMK)
C         WRITE(1,2009) (BK(K,I,J),J=1,NMK)
 190    CONTINUE
 200  CONTINUE
 1001 FORMAT(I5,3F10.5)
 1003 FORMAT(I5)
 1004 FORMAT(10I5)
 1005 FORMAT(10I5)
 1006 FORMAT(6E13.5)
 1007 FORMAT(10F10.5)
 2001 FORMAT(2X,'COOR',2I5,3F10.5)
 2002 FORMAT(2X,'=PI=',2I5,3F10.5)
 2003 FORMAT(2X,'NEQ=',I5)
 2004 FORMAT(2X,'=ID=',10I5)
 2005 FORMAT(2X,'JDIA',10I5)
 2006 FORMAT(2X,'BKCP',6E12.4)
 2007 FORMAT(2X,'BMI=',10F10.5)
 2008 FORMAT(2X,'PIP=',2I5,3F10.5)
 2009 FORMAT(2X,'=BK=',6E12.4)
 2010 FORMAT(10X,'====K====',I5)
      RETURN
      END

      SUBROUTINE SOK(LC,EPT,EPL,X,S,ST,SL,N,NDM)
      DIMENSION  LC(1),EPT(N,1),EPL(N,1),S(N,3,1),ST(N,3,1),
     1           SL(N,3,1),S1(3,3),X(1)
C     ****************************************************
C     *                                                  *
C     *    SET UP TRANSFORMATION METRIX OF EACH BODY     *
C     *                                                  *
C     ****************************************************
      DO 10 K=1,N
         EPT(K,1)=SIN(X(K)/2.)
         EPT(K,2)=0.
```

```
            EPT(K,3)=0.
            EPT(K,4)=COS(X(K)/2.)
            DO 9 I=1,3
               ST(K,I,I)=2.*EPT(K,I)**2+2.*EPT(K,4)**2-1.
               I1=I-1
               I2=I+1
               IF(I1.EQ.0) I1=3
               IF(I2.EQ.4) I2=1
               ST(K,I,I2)=(EPT(K,I)*EPT(K,I2)-EPT(K,4)*EPT(K,I1))*2.
               ST(K,I2,I)=(EPT(K,I)*EPT(K,I2)+EPT(K,4)*EPT(K,I1))*2.
9           CONTINUE
10       CONTINUE
         DO 20 K=1,N
            DO 19 I=1,3
               SL(K,I,I)=2.*EPL(K,I)**2+2.*EPL(K,4)**2-1.
               I1=I-1
               I2=I+1
               IF(I1.EQ.0) I1=3
               IF(I2.EQ.4) I2=1
               SL(K,I,I2)=(EPL(K,I)*EPL(K,I2)-EPL(K,4)*EPL(K,I1))*2.
               SL(K,I2,I)=(EPL(K,I)*EPL(K,I2)+EPL(K,4)*EPL(K,I1))*2.
19          CONTINUE
20       CONTINUE
         IF(N. EQ. 1) GOTO 44
         DO 40 K=2,N
            DO 39 I=1,3
               DO 38 J=1,3
                  S(K,I,J)=0.
                  DO 37 II=1,3
                     S(K,I,J)=S(K,I,J)+SL(K,I,II)*ST(K,II,J)
37                CONTINUE
                  IF( ABS( S(K,I,J) ) .LT. 1.E-20 ) S(K,I,J)=0.
38             CONTINUE
39          CONTINUE
40       CONTINUE
44       DO 50 I=1,3
            DO 49 J=1,3
               S(1,I,J)=ST(1,I,J)
49          CONTINUE
```

```
 50    CONTINUE
       IF(N .EQ. 1) GOTO 88
       DO 80 K=2,N
          LK=LC(K)
          DO 60 I=1,3
             DO 59 J=1,3
                S1(I,J)=0.
                DO 58 M=1,3
                   S1(I,J)=S1(I,J)+S(LK,I,M)*S(K,M,J)
 58             CONTINUE
 59          CONTINUE
 60       CONTINUE
          DO 70 I=1,3
             DO 68 J=1,3
                S(K,I,J)=S1(I,J)
 68          CONTINUE
 70       CONTINUE
 80    CONTINUE
 88    CONTINUE
C       WRITE(6,1001) (((ST(K,I,J),J=1,3),I=1,3),K=1,N)
C       WRITE(6,1002) (((SL(K,I,J),J=1,3),I=1,3),K=1,N)
       WRITE(1,1003) (((S(K,I,J),J=1,3),I=1,3),K=1,N)
 1001 FORMAT(5X,'==ST==',3F16.6)
 1002 FORMAT(5X,'==SL==',3F16.6)
 1003 FORMAT(5X,'==S===',3F16.6)
       RETURN
       END

       SUBROUTINE WKLM(LC,NR,S,W,N,NS,NDM)
       DIMENSION  LC(1),NR(N,1),S(N,3,1),W(N,3,1)
C      ****************************************************
C      *                                                  *
C      *   SET UP PARTIAL ANGULAR VELOCITY OF EACH BODY   *
C      *                                                  *
C      ****************************************************
       DO 200 K=1,N
          DO 120 M=1,3
             DO 110 L=1,NS
                W(K,M,L)=0.
```

```
110          CONTINUE
120       CONTINUE
          J=K
133       IF (J .EQ. 1) GO TO 177
          LJ=LC(J)
          LI=0
          J1=J-1
          DO 140 I=1,J1
             LI=LI+NR(I,1)
140       CONTINUE
          JJ=1
          DO 160 LL=1,3
             IF (NR(J,LL+1) .EQ. 1) THEN
                DO 150 M=1,3
                   W(K,M,LI+JJ)=S(LJ,M,LL)
150             CONTINUE
                JJ=JJ+1
             ENDIF
160       CONTINUE
          J=LJ
          GOTO 133
177       JJ=1
          DO 180 LL=1,3
             IF (NR(J,LL+1) .EQ. 1) THEN
                W(K,LL,JJ)=1.
                JJ=JJ+1
             ENDIF
180       CONTINUE
          DO 190 M=1,3
             WRITE(1,1000) (W(K,M,L),L=1,NS)
190       CONTINUE
200    CONTINUE
1000   FORMAT(2X,'===W===',12F8.4)
       RETURN
       END

       SUBROUTINE WKV(E,LC,NT,NR,NUMNP,NDF,NP,PH,S,SL,W,V,
      1           WK,VMM,VMX,X,N,NS,NDM)
       DIMENSION  E(3,3,3),LC(1),NT(N,1),NR(N,1),NUMNP(1),NDF(1),
```

```
     1           NP(1),PH(N,1),S(N,3,1),SL(N,3,1),W(N,3,1),V(N,3,1),
     1           W1(3),WK(N,3,3,1),VMM(N,3,1),VMX(N,1),X(1)
C     ******************************************************
C     *                                                    *
C     *   SET UP PARTIAL VELOCITY OF EACH JOIN POINTOR H   *
C     *                                                    *
C     ******************************************************
      INR=0
      INRT=0
      DO 105 K=1,N
         INR=INR+NR(K,1)
         INRT=INRT+NR(K,1)+NT(K,1)
 105  CONTINUE
      INR1=INR+1
      INT11=INR+NT(1,1)
      INDM=1
      IF(NDM.LT.3) INDM=2
C
C  SET UP WK(K,M,IK,L)=-E(M,J,I)*W(K,I,L)*S(J,IK)
C
      DO 144 K=1,N
         DO 143 M=1,3
            DO 142 L=1,NS
               DO 120 J=1,3
                  W1(J)=0.
                  DO 115 I=1,3
                     W1(J)=W1(J)-E(M,J,I)*W(K,I,L)
 115              CONTINUE
 120           CONTINUE
C               WRITE(6,8001) (W1(J),J=1,3)
               DO 141 IK=1,3
                  WK(K,M,IK,L)=0.
                  DO 130 J=1,3
                     WK(K,M,IK,L)=WK(K,M,IK,L)+W1(J)*S(K,J,IK)
 130              CONTINUE
 141           CONTINUE
 142        CONTINUE
 143     CONTINUE
 144  CONTINUE
```

```
      DO 40 K=1,N
         DO 20 IK=1,3
C            WRITE(1,8002) ((WK(K,M,IK,L),L=1,NS),M=1,3)
 20      CONTINUE
 40   CONTINUE
 8001 FORMAT(5X,'--W1---',5F10.5)
 8002 FORMAT(2X,'==WK===',14F8.3)
C
C  SET UP V(K,M,L)=XI(1,M,L)
C
      DO 200 K=1,N
         DO 170 M=1,3
            DO 160 L=1,NS
               V(K,M,L)=0.
 160        CONTINUE
 170     CONTINUE
         JJ=1
         IF (NT(1,1) .EQ. 0) GOTO 200
         DO 180 M=1,3
            IF (NT(K,M+1) .EQ. 1) THEN
              V(K,M,INR+JJ)=1.
              JJ=JJ+1
            ENDIF
 180     CONTINUE
 200  CONTINUE
C
C  SET UP VMM(K,M,L)=U(K,IH,L)+SL(LC(K),IH,IN)*XI(K,IN,L)
C
      INPT=INRT
      DO   205 K=1,N
       DO  205 M=1,3
        DO 205 L=1,NS
           VMM(K,M,L)=0.
 205  CONTINUE
      IF(N .EQ. 1) GOTO 333
      DO 300 K=2,N
         NPK=NUMNP(K)
         IF(NPK .EQ. 0) GOTO 300
         LK=LC(K)
```

```
         INP=INPT+(NP(LK)-1)*NDF(LK)
         NDFK=NDF(LK)
         DO 230 LP=1,NDFK
            VMM(K,INDM+LP-1,INP+LP)=1.
 230     CONTINUE
         INPT=INPT+NUMNP(LK)*NDF(LK)
         INTT=INT11
         JJ=1
         DO 240 LL=1,3
            IF (NT(K,LL+1) .EQ. 1) THEN
               DO 235 M=1,3
                  VMM(K,M,INTT+JJ)=SL(LK,M,LL)
 235           CONTINUE
               JJ=JJ+1
            ENDIF
 240     CONTINUE
         INTT=INTT+NT(K,1)
 300  CONTINUE
 333  CONTINUE
      DO 30 K=1,N
         DO 22 M=1,3
 22      CONTINUE
 30   CONTINUE
 8003 FORMAT(2X,'==VMM==',12F8.4)
C
C  SET UP VMX(K,I)=VMM(K,M,L)*X(L)
C
      DO 400 K=1,N
         DO 390 I=1,3
            VMX(K,I)=0.
            DO 380 L=1,NS
               VMX(K,I)=VMX(K,I)+VMM(K,I,L)*X(L)
 380        CONTINUE
 390     CONTINUE
 400  CONTINUE
C
C  SET UP V(K,M,L)=V(K,M,L)+WK(LJ,M,II,L)*(PH(J,II)+VMX(J,II)
C
      DO 500 K=1,N
```

```
      J=K
444   IF(J.LE.1) GOTO 500
      DO 490 M=1,3
         LJ=LC(J)
         INT111=INT11+1
         DO 480 L=1,NS
            DO 470 II=1,3
               V(K,M,L)=V(K,M,L)+WK(LJ,M,II,L)*(PH(J,II)+VMX(J,II))
470         CONTINUE
480      CONTINUE
490   CONTINUE
      J=LJ
      GOTO 444
500   CONTINUE
      DO 800 K=1,N
         J=K
         IF(J.LE.1) GOTO 800
         LJ=LC(J)
         DO 790 M=1,3
            DO 780 L=1,NS
               DO 770 I=1,3
               V(K,M,L)=V(K,M,L)+S(LJ,M,I)*VMM(K,I,L)
770            CONTINUE
780         CONTINUE
790      CONTINUE
800   CONTINUE
      DO 900 K=1,N
         DO 890 M=1,3
890      CONTINUE
900   CONTINUE
791   FORMAT(2X,'===V===',12F8.4)
      RETURN
      END

      SUBROUTINE WKVI(NT,NR,NUMNP,NDF,S,V,WK,X,VI,VIM,CO,N,NS,NPMAX)
      DIMENSION  NT(N,1),NR(N,1),NUMNP(1),NDF(1),
     1   S(N,3,1),V(N,3,1),WK(N,3,3,1),X(1),
     1   VI(N,NPMAX,3,1),VIM(3,1),CO(N,NPMAX,1),VIMX(3)
C     ***********************************************
```

```
C     *                                             *
C     *    SET UP PARTIAL VELOCITY OF EACH NODE     *
C     *                                             *
C     ***********************************************
      INRT=0
      DO 105 K=1,N
         INRT=INRT+NR(K,1)+NT(K,1)
 105  CONTINUE
      DO 800 K=1,N
         INDF=1
         IF(NDF(K) .EQ. 2) INDF=2
         IF(NDF(K) .EQ. 1) INDF=3
         NPK=NUMNP(K)
         IF(NPK .EQ. 0) GOTO 800
         DO 700 II=1,NPK
            DO 140 M=1,3
               DO 130 L=1,NS
                  VIM(M,L)=0.
 130           CONTINUE
 140        CONTINUE
            NDFK=NDF(K)
            DO 150 M=1,NDFK
               VIM(INDF-1+M,INRT+(II-1)*NDF(K)+M)=1.
 150        CONTINUE
            DO 170 M=1,3
               VIMX(M)=0.
               DO 160 L=1,NS
                  VIMX(M)=VIMX(M)+VIM(M,L)*X(L)
 160           CONTINUE
 170        CONTINUE
            DO 200 L=1,NS
               DO 190 M=1,3
                  VI(K,II,M,L)=V(K,M,L)
                  DO 180 IH=1,3
                     VI(K,II,M,L)=VI(K,II,M,L)+
     1                            WK(K,M,IH,L)*(CO(K,II,IH)+VIMX(IH))+
     1                            S(K,M,IH)*VIM(IH,L)
 180              CONTINUE
 190           CONTINUE
```

```
 200        CONTINUE
            DO 300 M=1,3
 300        CONTINUE
 700      CONTINUE
          INRT=INRT+NUMNP(K)*NDF(K)
 800  CONTINUE
 1001 FORMAT(5X,3F10.5)
 2001 FORMAT(/2X,'--R(M)-',3F10.5/)
 2002 FORMAT( 2X,'==VI===',12F8.4)
 3002 FORMAT( 2X,14F8.3)
 8000 FORMAT( 2X,'--VIM--',12F8.4)
      RETURN
      END

      SUBROUTINE SDWD(E,LC,NR,S,W,Y,SD,WD,N,NS,NDM)
      DIMENSION  E(3,3,3),LC(1),NR(N,1),S(N,3,1),W(N,3,1),
     1           Y(1),SD(N,3,1),WD(N,3,1),WA(3),SD1(3)
C     ********************************************************
C     *                                                      *
C     *   SET UP THE DERIVERTIVE OF TRANSFORMATION MATRICES  *
C     *                                                      *
C     ********************************************************
      DO 200 K=1,N
         DO 120 J=1,3
            WA(J)=0.
            DO 110 L=1,NS
               WA(J)=WA(J)+W(K,J,L)*Y(L)
 110        CONTINUE
 120     CONTINUE
         DO 170 I=1,3
            DO 140 M=1,3
               SD1(M)=0.
               DO 130 J=1,3
                  SD1(M)=SD1(M)-E(I,M,J)*WA(J)
 130           CONTINUE
 140        CONTINUE
            DO 160 J=1,3
               SD(K,I,J)=0.
               DO 150 M=1,3
```

```
                SD(K,I,J)=SD(K,I,J)+SD1(M)*S(K,M,J)
150           CONTINUE
160         CONTINUE
170       CONTINUE
200     CONTINUE
        DO 300 K=1,N
          DO 220 M=1,3
            DO 210 L=1,NS
              WD(K,M,L)=0.
210         CONTINUE
220       CONTINUE
          J=K
233       IF (J .EQ. 1) GO TO 300
          LJ=LC(J)
          LI=0
          J1=J-1
          DO 240 I=1,J1
            LI=LI+NR(I,1)
240       CONTINUE
          JJ=1
          DO 260 LL=1,3
            IF (NR(J,LL+1) .EQ. 1) THEN
              DO 250 M=1,3
                WD(K,M,LI+JJ)=SD(LJ,M,LL)
250           CONTINUE
              JJ=JJ+1
            ENDIF
260       CONTINUE
          J=LJ
          GOTO 233
300     CONTINUE
         WRITE(1,2001) (((SD(K,I,J),J=1,3),I=1,3),K=1,N)
         WRITE(1,2002) (((WD(K,M,L),L=1,NS),M=1,3),K=1,N)
2001    FORMAT(2X,'==SD===',3F16.6)
2002    FORMAT(2X,'==WD===',12F8.4)
        RETURN
        END

        SUBROUTINE WKDVD(E,LC,NT,NR,NDF,PH,S,W,WK,VMM,VMX,
```

```
     1         Y,SD,WD,VD,WKD,VMY,N,NS,NDM)
      DIMENSION  E(3,3,3),LC(1),NT(N,1),NR(N,1),NDF(1),
     1           PH(N,1),S(N,3,1),W(N,3,1),
     1           W1(3),W2(3),WK(N,3,3,1),VMM(N,3,1),VMX(N,1),Y(1),
     1           SD(N,3,1),WD(N,3,1),VD(N,3,1),WKD(N,3,3,1),VMY(N,1)
C     ******************************************************************
C     *                                                                *
C     *  SET UP DERIVATIVE OF PARTIAL VELOCITY OF EACH JOIN POINTOR H  *
C     *                                                                *
C     ******************************************************************
      INR=0
      INRT=0
      DO 105 K=1,N
         INR=INR+NR(K,1)
         INRT=INRT+NR(K,1)+NT(K,1)
 105  CONTINUE
      INR1=INR+1
      INT11=INR+NT(1,1)
      INDM=1
      IF(NDM.LT.3) INDM=2
      DO 144 K=1,N
         DO 143 M=1,3
            DO 142 L=1,NS
               DO 120 J=1,3
                  W1(J)=0.
                  W2(J)=0.
                  DO 115 I=1,3
                     W1(J)=W1(J)-E(M,J,I)*WD(K,I,L)
                     W2(J)=W2(J)-E(M,J,I)*W(K,I,L)
 115              CONTINUE
 120           CONTINUE
               DO 141 IK=1,3
                  WKD(K,M,IK,L)=0.
                  DO 130 J=1,3
                     WKD(K,M,IK,L)=WKD(K,M,IK,L)+W1(J)*S(K,J,IK)
     1                                         +W2(J)*SD(K,J,IK)
 130              CONTINUE
 141           CONTINUE
 142        CONTINUE
```

```
143      CONTINUE
144   CONTINUE
      DO 40 K=1,N
         DO 20 IK=1,3
            WRITE(6,8002) ((WKD(K,M,IK,L),L=1,NS),M=1,3)
20       CONTINUE
40    CONTINUE
8001 FORMAT(5X,'--W1---',5F10.5)
8002 FORMAT(2X,'==WKD==',12F8.4)
      DO 400 K=1,N
         DO 390 I=1,3
            VMY(K,I)=0.
            DO 380 L=1,NS
               VMY(K,I)=VMY(K,I)+VMM(K,I,L)*Y(L)
380         CONTINUE
390      CONTINUE
400   CONTINUE
      DO 500 K=1,N
         DO 420 M=1,3
            DO 410 L=1,NS
               VD(K,M,L)=0.
410         CONTINUE
420      CONTINUE
         J=K
444      IF(J.LE.1) GOTO 500
         DO 490 M=1,3
            LJ=LC(J)
            INT111=INT11+1
            DO 480 L=1,NS
               DO 470 II=1,3
                 VD(K,M,L)=VD(K,M,L)+WKD(LJ,M,II,L)*(PH(J,II)+VMX(J,II))
    1                               +WK(LJ,M,II,L)*VMY(J,II)
470            CONTINUE
480         CONTINUE
490      CONTINUE
         J=LJ
         GOTO 444
500   CONTINUE
      DO 800 K=1,N
```

```
      J=K
      IF(J.LE.1) GOTO 800
      LJ=LC(J)
      DO 790 M=1,3
         DO 780 L=1,NS
            DO 770 I=1,3
            VD(K,M,L)=VD(K,M,L)+SD(LJ,M,I)*VMM(K,I,L)
 770        CONTINUE
 780     CONTINUE
 790  CONTINUE
 800  CONTINUE
      DO 900 K=1,N
         DO 890 M=1,3
            WRITE(1,791) (VD(K,M,L),L=1,NS)
 890     CONTINUE
 900  CONTINUE
 791  FORMAT(2X,'===VD==',12F8.4)
      RETURN
      END

      SUBROUTINE WKVID(NT,NR,NUMNP,NDF,WK,X,VIM,
     1                 Y,CO,SD,VD,WKD,VID,N,NS,NPMAX)
      DIMENSION  NT(N,1),NR(N,1),NUMNP(1),NDF(1),WK(N,3,3,1),X(1),
     1           VIM(3,1),VIMX(3),VIMY(3),Y(1),CO(N,NPMAX,1),
     1           SD(N,3,1),VD(N,3,1),WKD(N,3,3,1),VID(N,NPMAX,3,1)
C     ***********************************************************
C     *                                                         *
C     *    SET UP DERIVATIVE OF PARTIAL VELOCITY OF EACH NODE   *
C     *                                                         *
C     ***********************************************************
      INRT=0
      DO 105 K=1,N
         INRT=INRT+NR(K,1)+NT(K,1)
 105  CONTINUE
      DO 800 K=1,N
         INDF=1
         IF(NDF(K) .EQ. 2) INDF=2
         IF(NDF(K) .EQ. 1) INDF=3
         NPK=NUMNP(K)
```

```
      IF(NPK .EQ. 0) GOTO 800
      DO 700 II=1,NPK
         DO 140 M=1,3
            DO 130 L=1,NS
               VIM(M,L)=0.
130         CONTINUE
140      CONTINUE
         NDFK=NDF(K)
         DO 150 M=1,NDFK
            VIM(INDF-1+M,INRT+(II-1)*NDF(K)+M)=1.
150      CONTINUE
         WRITE(1,8000) ((VIM(M,L),L=1,NS),M=1,3)
         DO 170 M=1,3
            VIMX(M)=0.
            VIMY(M)=0.
            DO 160 L=1,NS
               VIMX(M)=VIMX(M)+VIM(M,L)*X(L)
               VIMY(M)=VIMY(M)+VIM(M,L)*Y(L)
160         CONTINUE
170      CONTINUE
         DO 200 L=1,NS
            DO 190 M=1,3
               VID(K,II,M,L)=VD(K,M,L)
               DO 180 IH=1,3
                  VID(K,II,M,L)=VID(K,II,M,L)+
     1                          WKD(K,M,IH,L)*(CO(K,II,IH)+VIMX(IH))+
     1                          WK(K,M,IH,L)*VIMY(IH)+
     1                          SD(K,M,IH)*VIM(IH,L)
180            CONTINUE
190         CONTINUE
200      CONTINUE
         WRITE(1,2002) ((VID(K,II,M,L),L=1,NS),M=1,3)
700   CONTINUE
      INRT=INRT+NUMNP(K)*NDF(K)
800   CONTINUE
1001  FORMAT(5X,3F10.5)
2002  FORMAT( 2X,'==VID==',12F8.4)
8000  FORMAT( 2X,'--VIM--',12F8.4)
      RETURN
```

```
      END

      SUBROUTINE ALP(NT,NR,NUMNP,NDF,CO,BMI,BM,BII,RC,S,W,V,VI,VIM,VIN,
     1               UI,BIN,BIM,A,AI,AM,AMI,N,NS,NDM,NPMAX)
      DIMENSION NT(N,1),NR(N,1),NUMNP(1),NDF(1),CO(N,NPMAX,1),BMI(N,1),
     1  BM(1),BII(N,3,1),S(N,3,1),W(N,3,1),V(N,3,1),VI(N,NPMAX,3,1),
     1  VIM(3,1),VIN(3,1),UI(N,NPMAX,1),BIN(N,3,1),BIM(N,3,1),
     1  A(NS,1),AI(NS,1),AM(3,1),AMI(3,1),RU(3,3),UT(3,3),
     1  SCO(3),SUI(3),RC(N,1),SRC(3)
C     ***********************************************************
C     *                                                         *
C     *             SET UP THE GENERALIZED MASS MATRIX          *
C     *                                                         *
C     ***********************************************************
      INRT=0
      DO 100 K=1,N
         INRT=INRT+NR(K,1)+NT(K,1)
  100 CONTINUE
      DO  101 L=1,NS
       DO 101 J=1,NS
         A(L,J)=0.
  101 CONTINUE
      DO 900 K=1,N
         INDF=1
         IF(NDF(K) .EQ. 2) INDF=2
         IF(NDF(K) .EQ. 1) INDF=3
         NPK=NUMNP(K)
         DO 200 L=1,NS
            DO 120 M=1,3
               AM(M,L)=0.
               DO 110 I=1,3
                  AM(M,L)=AM(M,L)
     1                   +(BII(K,M,I)+BIN(K,M,I)+BIM(K,M,I))*W(K,I,L)
  110          CONTINUE
  120       CONTINUE
            DO 190 J=1,NS
               DO 130 M=1,3
                  A(L,J)=A(L,J)+BM(K)*V(K,M,L)*V(K,M,J)+AM(M,L)*W(K,M,J)
  130          CONTINUE
```

```
 190        CONTINUE
 200      CONTINUE
          DO 201 L=1,NS
 201      CONTINUE
          IF( NPK .EQ. 0 ) THEN
C
C     CO,UI COORDINATE TRANSFOMATION FOR EACH BODY
C
          DO 13 M=1,3
            SRC(M)=0.
            DO 12 I=1,3
              SRC(M)=SRC(M)+S(K,M,I)*RC(K,I)
 12        CONTINUE
 13      CONTINUE
C
C     SET UP CO,UI INSYMMATRIC MATRICES
C
            DO 16 I=1,3
              RU(I,I)=0.
  16        CONTINUE
            RU(1,2)=-SRC(3)
            RU(1,3)=+SRC(2)
            RU(2,3)=-SRC(1)
            RU(2,1)=-RU(1,2)
            RU(3,1)=-RU(1,3)
            RU(3,2)=-RU(2,3)
            DO 20 L=1,NS
             DO 20 M=1,3
               AM(M,L)=0.
               DO 15 I=1,3
                 AM(M,L)=AM(M,L)+RU(M,I)*V(K,I,L)
  15           CONTINUE
  20        CONTINUE
            DO 40 L=1,NS
              DO 34 J=1,NS
                DO 33 M=1,3
                  A(L,J)=A(L,J)
     1                           +BM(K)*( AM(M,L)*W(K,M,J)
     1                                  + AM(M,J)*W(K,M,L) )
```

```
 33             CONTINUE
 34           CONTINUE
 40         CONTINUE
            ENDIF
          IF(NPK .EQ. 0) GOTO 900
C
C     CO,UI COORDINATE TRANSFOMATION FOR EACH NODE
C
          DO 800 IP=1,NPK
            DO 203 M=1,3
              SCO(M)=0.
              SUI(M)=0.
              DO 202 I=1,3
                SCO(M)=SCO(M)+S(K,M,I)*CO(K,IP,I)
                SUI(M)=SUI(M)+S(K,M,I)*UI(K,IP,I)
 202          CONTINUE
 203        CONTINUE
C
C     SET UP CO,UI INSYMMATRIC MATRICES
C
            DO 206 I=1,3
              RU(I,I)=0.
              UT(I,I)=0.
 206        CONTINUE
            RU(1,2)=-SCO(3)-SUI(3)
            RU(1,3)=+SCO(2)+SUI(2)
            RU(2,3)=-SCO(1)-SUI(1)
            RU(2,1)=-RU(1,2)
            RU(3,1)=-RU(1,3)
            RU(3,2)=-RU(2,3)
            UT(1,2)=-SUI(3)
            UT(1,3)=+SUI(2)
            UT(2,3)=-SUI(1)
            UT(2,1)=-UT(1,2)
            UT(3,1)=-UT(1,3)
            UT(3,2)=-UT(2,3)
C
C     SET UP PARTIAL VELOCITY OF EACH NODE
C
```

```
      DO 280 M=1,3
         DO 270 L=1,NS
            VIN(M,L)=0.
270      CONTINUE
280   CONTINUE
      NDFK=NDF(K)
      DO 290 M=1,NDFK
         VIN(INDF-1+M,INRT+(IP-1)*NDF(K)+M)=1.
290   CONTINUE
      DO  292 M=1,3
       DO 292 L=1,NS
          VIM(M,L)=0.
          DO 291 I=1,3
             VIM(M,L)=VIM(M,L)+S(K,M,I)*VIN(I,L)
291       CONTINUE
292   CONTINUE
      DO 298 M=1,3
298   CONTINUE
      DO 320 L=1,NS
         DO 320 M=1,3
            AM(M,L)=0.
            DO 310 I=1,3
               AM(M,L)=AM(M,L)+RU(M,I)*(V(K,I,L)+VIM(I,L))
310         CONTINUE
320   CONTINUE
      DO 400 L=1,NS
         DO 340 J=1,NS
            DO 330 M=1,3
               A(L,J)=A(L,J)
     1                          +BMI(K,IP)*( AM(M,L)*W(K,M,J)
     1                                     + AM(M,J)*W(K,M,L) )
330         CONTINUE
340      CONTINUE
400   CONTINUE
      DO 1400 L=1,NS
         DO 1340 J=1,NS
            DO 1330 M=1,3
               A(L,J)=A(L,J)+BMI(K,IP)*
     1                                    ( V(K,M,J)*VIM(M,L)
```

```
     1                                         + V(K,M,L)*VIM(M,J) )
1330              CONTINUE
1340           CONTINUE
1400        CONTINUE
            DO 500 L=1,NS
               DO 440 J=1,NS
                  DO 430 M=1,3
                     A(L,J)=A(L,J)+BMI(K,IP)*VIM(M,L)*VIM(M,J)
430               CONTINUE
440            CONTINUE
500         CONTINUE
            DO 544 L=1,NS
               DO 520 M=1,3
                  AM(M,L)=0.
                  DO 510 I=1,3
                     AM(M,L)=AM(M,L)-UT(M,I)*W(K,I,L)
510               CONTINUE
520            CONTINUE
               DO 540 M=1,3
                  AMI(M,L)=0.
                  DO 530 I=1,3
                     AMI(M,L)=AMI(M,L)+UT(M,I)*AM(I,L)
530               CONTINUE
540            CONTINUE
544         CONTINUE
            DO 600 L=1,NS
               DO 560 J=1,NS
                  DO 550 M=1,3
                     A(L,J)=A(L,J)+BMI(K,IP)*AMI(M,J)*W(K,M,L)
550               CONTINUE
560            CONTINUE
600         CONTINUE
800      CONTINUE
         INRT=INRT+NUMNP(K)*NDF(K)
900   CONTINUE
      N1=N+1
      DO  901 L=1,N
       DO 901 J=N1,NS
         A(L,J)=0.
```

```
901  CONTINUE
     DO  902 L=1,NS
      DO 902 J=1,NS
         AI(L,J)=A(L,J)
902  CONTINUE
     DO 908 L=1,NS
        WRITE(1,2008) (A(L,J),J=1,NS)
908  CONTINUE
     DO 909 L=1,NS
        WRITE(1,2010) (AI(L,J),J=1,NS)
909  CONTINUE
2001 FORMAT(5X,'==A.1==',12F8.4)
2002 FORMAT(5X,'==SCO===',12F8.4)
2003 FORMAT(5X,'==SUI===',12F8.4)
2004 FORMAT(5X,'==VIM==',12F8.4)
2005 FORMAT(5X,'==A.2==',10F10.5)
2006 FORMAT(5X,'==A.3==',10F10.5)
2007 FORMAT(5X,'==A.4==',10F10.5)
2008 FORMAT(2X,'==A==',10F10.5)
2010 FORMAT(2X,'==AI=',10F10.5)
2009 FORMAT(2X,7F10.6)
8001 FORMAT(5X,'==AM==',4F10.4)
     RETURN
     END


     SUBROUTINE FLL(KEY,NT,NR,NUMNP,NDF,BM,HF,HM,IDS,RC,S,W,V,
    1  X,T,VIM,VIN,UI,CO,PI,BK,A,AI,F,FI,FL,FM,N,NS,NPMAX,NMMAX)
     DIMENSION NT(N,1),NR(N,1),NUMNP(1),NDF(1),BM(1),HF(N,1),HM(N,1),
    1 RC(N,1),S(N,3,1),W(N,3,1),V(N,3,1),X(1),VIM(3,1),VIN(3,1),IDS(1),
    1 UI(N,NPMAX,1),CO(N,NPMAX,1),PI(N,NPMAX,1),BK(N,NMMAX,1),A(NS,1),
    1 F(1),FL(1),FM(1),Q1(3),Q(3),SCO(3),SUI(3),RU(3,3),HMP(3),SRC(3),
    1 AI(NS,1),FI(1)
     COMMON/ROOM/MR(100000)
     COMMON/POINT3/NUI,N31,N32,NYL,NYI,NYD,ND1,ND2,ND3
     COMMON/POINT1/N9,N10,N11,N12,N13,N14,N15,N16,N17,N18,N19,N20
C    *********************************************************
C    *                                                       *
C    *           SET UP THE GENERALIZED LOAD VECTOR          *
```

```
C     *                                                               *
C     ****************************************************************
 8001 FORMAT(5X,'--X--',8F13.8)
      DO 100 L=1,NS
         FL(L)=0.
 100  CONTINUE
      SIDA=0.
      DO 190 K=1,N
         NPK=NUMNP(K)
         DO 102 M=1,3
            HMP(M) =0.
 102     CONTINUE
         IF(NPK .EQ. 0) GOTO 155
C
C     CO,UI COORDINATE TRANSFOMATION FOR EACH NODE
C
         DO 150 IP=1,NPK
            DO 120 M=1,3
               SCO(M)=0.
               SUI(M)=0.
               DO 110 I=1,3
                  SCO(M)=SCO(M)+S(K,M,I)*CO(K,IP,I)
                  SUI(M)=SUI(M)+S(K,M,I)*UI(K,IP,I)
 110           CONTINUE
 120        CONTINUE
C
C     SET UP CO,UI INSYMMATRIC MATRICES
C
            DO 130 I=1,3
               RU(I,I)=0.
 130        CONTINUE
            RU(1,2)=-SCO(3)-SUI(3)
            RU(1,3)=+SCO(2)+SUI(2)
            RU(2,3)=-SCO(1)-SUI(1)
            RU(2,1)=-RU(1,2)
            RU(3,1)=-RU(1,3)
            RU(3,2)=-RU(2,3)
            DO  140 M=1,3
             DO 140 I=1,3
```

```
            HMP(M)=HMP(M)+RU(M,I)*PI(K,IP,I)
140       CONTINUE
          IF(K .EQ. 1 .AND. IP .EQ. 2) THEN
            KK=K+1
            DO 144 M=1,3
               HMP(M)=HMP(M)+RU(M,KK)*BM(KK)*9.8
144         CONTINUE
          ENDIF
155     IF(NPK .NE. 0) GOTO 177
        DO 170 M=1,3
           SRC(M)=0.
           DO 160 I=1,3
              SRC(M)=SRC(M)+S(K,M,I)*RC(K,I)
160        CONTINUE
170     CONTINUE
8008    FORMAT(2X,'--SRC--', 3F10.5)
        HMP(1)=-SRC(3)*BM(K)*9.8
        HMP(3)=+SRC(1)*BM(K)*9.8
177     CONTINUE
        HF(K,2)=BM(K)*9.8
        IF(IDS(K) .NE. 0) THEN
          CALL DESN(K,T,MR(N16),N)
        END IF
        DO  180 L=1,NS
         DO 180 M=1,3
           IF(KEY .NE. 20) THEN
             FL(L)=FL(L)+                W(K,M,L)*( HM(K,M)+HMP(M) )
           ELSE
             FL(L)=FL(L)+               W(K,M,L)*HM(K,M)
           ENDIF
180     CONTINUE
190  CONTINUE
     II=0
     DO 204 K=1,N
        II=II+NR(K,1)+NT(K,1)
204  CONTINUE
     DO 290 K=1,N
        NPK=NUMNP(K)
        IF(NPK .EQ. 0) GOTO 290
```

```
      NMK=NDF(K)*NUMNP(K)
      NDFK=NDF(K)
      INDF=1
      IF(NDF(K) .EQ. 2) INDF=2
      IF(NDF(K) .EQ. 1) INDF=3
      I1=INDF-1
      IM=0
      II1=II+1
      II2=II+NMK
      DO 280 IP=1,NPK
         DO 220 M=1,3
            DO 210 L=1,NS
               VIN(M,L)=0.
 210        CONTINUE
 220     CONTINUE
         DO 230 M=1,NDFK
            VIN(INDF-1+M,II+(IP-1)*NDF(K)+M)=1.
 230     CONTINUE
         DO  234 M=1,3
          DO 234 L=1,NS
             VIM(M,L)=0.
              DO 232 I=1,3
                 VIM(M,L)=VIM(M,L)+S(K,M,I)*VIN(I,L)
 232          CONTINUE
 234     CONTINUE
         DO 240 M=1,3
 240     CONTINUE
         DO 250 M=1,3
            Q1(M)=0.
 250     CONTINUE
         DO 270 M=INDF,3
            DO 260 J=1,NMK
               Q1(M)=Q1(M)+BK(K,IM+M-I1,J)*X(II+J)
 260        CONTINUE
 270     CONTINUE
C        WRITE(1,3006) K,IP,(Q1(M),M=1,3)
         DO 272 M=1,3
            Q(M)=0.
            DO 271 I=1,3
```

```
                  Q(M)=Q(M)+S(K,M,I)*Q1(I)
 271           CONTINUE
 272        CONTINUE
C           WRITE(1,3002) K,IP,(Q(M),M=1,3)
            DO  273 L=1,NS
             DO 273 M=1,3
               FL(L)=FL(L)-VIM(M,L)*Q(M)
 273        CONTINUE
            IM=IM+3-INDF+1
            SIDA=SIDA+X(K)
            IF(K .EQ. 1) THEN
              DO  276 L=1,NS
               DO 276 M=1,3
                FL(L)=FL(L) + VIM(M,L)* PI(K,IP,M)
                IF(IP .EQ. 2) THEN
                  FL(L)=FL(L) - VIM(M,L)* HF(2,M)*SIN(SIDA)
                ENDIF
 276          CONTINUE
            ENDIF
 280     CONTINUE
         II=II+NDF(K)*NUMNP(K)
 290  CONTINUE
      IF(KEY .EQ. 20) THEN
        DO 381 L=1,NS
           F(L)=FL(L)-FM(L)
 381    CONTINUE
      ELSE
        DO 400 L=1,NS
           F(L)=0.
           DO 390 J=1,NS
              F(L)=F(L)+AI(L,J)*( FL(J)-FM(J) )
 390       CONTINUE
 400    CONTINUE
      ENDIF
      WRITE(1,9003) T,(F(L),L=1,NS)
      IF(KEY .EQ. 1) THEN
        N1=N+1
        DO 500 L=1,N
           FI(L)=0.
```

```
          DO 490 J=N1,NS
             FI(L)=FI(L)+A(J,L)*F(J)
490       CONTINUE
          F(L)=F(L)-FI(L)
500     CONTINUE
      ENDIF
2001  FORMAT(2X,'=FL.1=',7F10.5)
2002  FORMAT(2X,'=FL.2=',7F10.5)
2003  FORMAT(2X,'=FL.3=',7F10.5)
2004  FORMAT(/2X,'==F===',7F10.5)
3001  FORMAT(2X,'=VIM=',7F10.5)
3002  FORMAT(2X,'===Q===',2I5,3F18.8)
3003  FORMAT(2X,'==HMP==',2I5,3F15.5)
3004  FORMAT(2X,'--HMP--',I5,3F15.5)
3005  FORMAT(2X,'==HF==&',I5,5F15.5)
3006  FORMAT(2X,'===Q1==',2I5,3F18.8)
3007  FORMAT(2X,'==HM==&',I5,5F15.5)
5001  FORMAT(2X,'==SCO==',3F15.5)
5002  FORMAT(2X,'==SUI==',3F15.5)
9001  FORMAT(2X,'===PI==',3F15.5)
9002  FORMAT(2X,'===RU==',3F15.5)
9003  FORMAT(3F15.5)
      RETURN
      END

      SUBROUTINE DEFY(KEY,LC,NT,NR,NUMNP,NDF,S,W,WD,V,X,VI,T,Y,YI,
     1                BK,SD,VD,VID,A,AI,F,N,NS,NDM,NPMAX,NMMAX)
      COMMON/ROOM/MR(100000)
      COMMON/POINTO/N1,N2,N3,N4,N5,N6
      COMMON/POINTF/NF1,NF2,NF3,NF4,NF5,NF6,NF7,NF8
      COMMON/POINT1/N9,N10,N11,N12,N13,N14,N15,N16,N17,N18,N19,N20
      COMMON/POINT2/N21,N22,N23,N24,N25,NXL,N26,N27,N28,N29
      COMMON/POINT3/NUI,N31,N32,NYL,NYI,NYD,ND1,ND2,ND3
      COMMON/POINT4/N41,N42,N43,N44,N45,N46,N47
      COMMON/POINT5/N51,N52,N53,N54,N55
      COMMON/POINT6/N61,N62,N63,N64
      COMMON/POINT7/N71,N72,N73,N74,N75
      DIMENSION LC(1),NT(N,1),NR(N,1),NUMNP(1),NDF(1),S(N,3,1),W(N,3,1),
     1 WD(N,3,1),V(N,3,1),X(1),VI(N,NPMAX,3,1),Y(1),YI(1),BK(N,NMMAX,1),
```

```
     1 SD(N,3,1),VD(N,3,1),VID(N,NPMAX,3,1),A(NS,1),AI(NS,1),F(1),
     1 E(3,3,3),
     1 C(2,2),CI(2,2),D(2,5),CID(2,5),
     1 ALF(7,5),AA(7,7),A1(7,5),AN(5,5),F1(5),FN(12),YN(12)
C     ************************************************************
C     *                                                          *
C     *          SET UP RIGHT FUNCTION OF THE GOVERNING EQUATIONS *
C     *                                                          *
C     *             KEY=0                                        *
C     *             KEY=1  FOR T1(KANE'S EXAMPLE )               *
C     *             KEY=7                                        *
C     *             KEY=20 Close loop                            *
C     *                                                          *
C     ************************************************************
      NS1=NS+1
      NS2=NS*2
      WRITE(1,8001) (X(I),I=1,NS)
      WRITE(1,8005) (Y(I),I=1,NS2)
 8001 FORMAT(/2X,'=====X=====',6F8.4)
 8005 FORMAT(/2X,'=====Y=====',6F8.4)
      IF(KEY .NE. 20) THEN
        DO 200 I=1,NS
           X(I)=Y(I+NS)
 200    CONTINUE
        DO 201 I=NS1,NS2
           F(I)=Y(I-NS)
 201    CONTINUE
      ELSE
        DO 110 I=1,12
           YN(I)=Y(I)
 110    CONTINUE
        DO 120 I=3,6
           II=I-1
           Y(I)=YN(II)
 120    CONTINUE
        DO 130 I=1,2
           JJ=I+10
           Y(I)=F(JJ)
 130    CONTINUE
```

```
        DO 140 I=7,8
           KK=I+4
           Y(I)=YN(KK)
140     CONTINUE
        DO 150 I=9,12
           II=I-2
           Y(I)=YN(II)
150     CONTINUE
        DO 160 I=1,6
           X(I)=Y(I+6)
160     CONTINUE
      ENDIF
        WRITE(1,3001) (X(I),I=1,6)
        WRITE(1,3005) (Y(I),I=1,12)
      IF(KEY .EQ. 7) GOTO 777
      CALL SOK(LC,MR(N13),MR(N14),X,S,MR(N18),MR(N19),N,NDM)
      CALL WKLM(LC,NR,S,W,N,NS,NDM)
      DO   210 I=1,3
       DO  210 J=1,3
        DO 210 K=1,3
           E(I,J,K)=0.
210   CONTINUE
      E(1,2,3)=1.
      E(2,3,1)=1.
      E(3,1,2)=1.
      E(2,1,3)=-1.
      E(1,3,2)=-1.
      E(3,2,1)=-1.
      CALL WKV(E,LC,NT,NR,NUMNP,NDF,MR(N6),MR(N12),S,MR(N19),
     1          W,V,MR(N23),MR(N24),MR(N25),X,N,NS,NDM)
      CALL WKVI(NT,NR,NUMNP,NDF,S,V,MR(N23),X,VI,MR(N27),
     1           MR(NF1),N,NS,NPMAX)
      CALL BINM(NT,NR,NUMNP,NDF,MR(NF1),MR(NF3),MR(N10),S,X,
     1           MR(NUI),MR(N31),MR(N32),N,NS,NPMAX)
      CALL ALP(NT,NR,NUMNP,NDF,MR(NF1),MR(NF3),MR(N9),MR(N10),MR(N28),
     1   S,W,V,VI,MR(N27),MR(N29),MR(NUI),MR(N31),MR(N32),
     1   A,AI,MR(N54),MR(N55),N,NS,NDM,NPMAX)
      CALL ATA(AI,NS)
666   CALL SDWD(E,LC,NR,S,W,Y,SD,WD,N,NS,NDM)
```

```
      CALL WKDVD(E,LC,NT,NR,NDF,MR(N12),S,W,MR(N23),MR(N24),MR(N25),
     1                    Y,SD,WD,VD,MR(N45),MR(N46),N,NS,NDM)
      CALL WKVID(NT,NR,NUMNP,NDF,MR(N23),X,MR(N27),
     1             Y,MR(NF1),SD,VD,MR(N45),VID,N,NS,NPMAX)
      CALL FML(NT,NR,NUMNP,NDF,MR(NF1),MR(NF3),MR(N9),MR(N10),
     1 S,W,WD,V,VI,MR(N27),MR(N29),MR(NUI),MR(N31),MR(N32),Y,VD,VID,
     1 A,MR(N54),MR(N55),MR(N63),N,NS,NDM,NPMAX)
      CALL FLL(KEY,NT,NR,NUMNP,NDF,MR(N9),MR(N15),MR(N16),MR(ND1),
     1 MR(N28),S,W,V,X,T,MR(N27),MR(N29),MR(NUI),MR(NF1),MR(NF2),
     1 BK,A,AI,F,MR(N64),MR(N62),MR(N63), N,NS,NPMAX,NMMAX)
        IF(KEY .EQ. 1) THEN
          N1=N+1
          N2=N*2
          DO 300 I=N1,N2
             F(I)=YI(I-N)
300       CONTINUE
        ENDIF
       IF(KEY .NE. 7) GOTO 778
777    CALL FT1(X,Y,T,F,N,NS)
778    CONTINUE
3001 FORMAT(/2X,'=X=',6F8.4)
3002 FORMAT(2X,'--Y--',5E15.6)
3003 FORMAT(2X,'--Y--',5E15.6)
3005 FORMAT(/2X,'=Y=',6F8.4)
3006 FORMAT(/2X,'=YN',6F8.4)
4001 FORMAT(2X,'S===',10F10.5)
4002 FORMAT(2X,'W===',10F10.5)
4003 FORMAT(2X,'V===',10F10.5)
4004 FORMAT(2X,'SD==',10F10.5)
4005 FORMAT(2X,'WD==',10F10.5)
4006 FORMAT(2X,'VD==',10F10.5)
5001 FORMAT(2X,'VI==',10F10.5)
5002 FORMAT(2X,'VID=',10F10.5)
5003 FORMAT(2X,'FM==',10F10.5)
5004 FORMAT(2X,'FL==',10F10.5)
5005 FORMAT(2X,'F===',10F10.5)
5006 FORMAT(2X,'AINV=',10F10.5)
     RETURN
     END
```

```
      SUBROUTINE IVSN(A,B,C,N,ME,DE,EP)
      DIMENSION A(2,2),B(2),C(2),ME(100)
C     ***********************************************************
C     *                                                         *
C     *            CALCULATE THE INVERSE OF MATRICES            *
C     *                                                         *
C     ***********************************************************
      DE=1.
      DO 10 J=1,N
 10   ME(J)=J
      DO 20 I=1,N
      Y=0.
      DO 30 J=I,N
      IF(ABS(A(I,J)).LE.ABS(Y)) GOTO 30
      K=J
      Y=A(I,J)
 22   CONTINUE
 30   CONTINUE
      DE=DE*Y
      IF(ABS(Y).LT.EP) THEN
        WRITE(3,4444)
        STOP
      ENDIF
      Y=1./Y
      DO 40 J=1,N
      C(J)=A(J,K)
      A(J,K)=A(J,I)
      A(J,I)=-C(J)*Y
      B(J)=A(I,J)*Y
 40   A(I,J)=A(I,J)*Y
      A(I,I)=Y
      J=ME(I)
      ME(I)=ME(K)
      ME(K)=J
      DO 11 K=1,N
      IF(K.EQ.I) GO TO 11
      DO 12 J=1,N
      IF(J.EQ.I) GO TO 12
```

```
      A(K,J)=A(K,J)-B(J)*C(K)
 12   CONTINUE
 11   CONTINUE
 20   CONTINUE
      DO 33 I=1,N
      DO 44 K=1,N
      IF(ME(K).EQ.I) GO TO 55
 44   CONTINUE
 55   IF(K.EQ.I) GO TO 33
      DO 66 J=1,N
      W=A(I,J)
      A(I,J)=A(K,J)
 66   A(K,J)=W
      IW=ME(I)
      ME(I)=ME(K)
      ME(K)=IW
      DE=-DE
 33   CONTINUE
4444  FORMAT(/2X,'444-444-444')
      RETURN
      END

      SUBROUTINE ATA(A,NS)
      DIMENSION A(NS,1),AA(25,25),AINV(7,7),B(25),C(25),ME(25),
     1          WKAREA(25),EI(25,25)
C     ***********************************************************
C     *                                                         *
C     *             CALCULATE THE INVERSE OF MATRIX A           *
C     *                                                         *
C     ***********************************************************
      DO  200 L=1,NS
       DO 200 J=1,NS
          AA(L,J)=A(L,J)
 200  CONTINUE
C      CALL LINV1F(A,NS,NS,AINV,3,WKAREA,IER)
       EPS=10.0E-8
       CALL IVSN(A,B,C,NS,ME,DE,EPS)
      DO 400 L=1,NS
C         WRITE(1,2002) (A(L,J),J=1,NS)
```

```
 400  CONTINUE
 2001 FORMAT(2X,'==AA==',7F10.5)
 2002 FORMAT(2X,'=AINV=',10F10.5)
      DO  600 L=1,NS
       DO 600 J=1,NS
          EI(L,J)=0.
          DO 590 I=1,NS
             EI(L,J)=EI(L,J)+AA(L,I)*A(I,J)
 590      CONTINUE
 600  CONTINUE
      DO 700 L=1,NS
C        WRITE(1,4001) (EI(L,J),J=1,NS)
 700  CONTINUE
 4001 FORMAT(2X,'==EI==',14F8.3)
      RETURN
      END
      SUBROUTINE CTC(A,IK)
      DIMENSION A(2,2),AA(2,2),AINV(2,2),B(25),C(25),ME(25),
     1          WKAREA(25),EI(25,25)
      DO  200 L=1,IK
       DO 200 J=1,IK
          AA(L,J)=A(L,J)
 200  CONTINUE
       EPS=10.0E-8
       CALL IVSN(A,B,C,NS,ME,DE,EPS)
      DO 400 L=1,IK
         WRITE(1,2002) (A(L,J),J=1,IK)
 400  CONTINUE
 2001 FORMAT(2X,'==AA==',7F10.5)
 2002 FORMAT(2X,'=AINV=',10F10.5)
      DO  600 L=1,IK
       DO 600 J=1,IK
          EI(L,J)=0.
          DO 590 I=1,IK
             EI(L,J)=EI(L,J)+AA(L,I)*A(I,J)
 590      CONTINUE
 600  CONTINUE
      DO 700 L=1,IK
         WRITE(1,4001) (EI(L,J),J=1,IK)
```

```
 700  CONTINUE
 4001 FORMAT(2X,'==EI==',14F8.3)
      RETURN
      END

      SUBROUTINE BINM(NT,NR,NUMNP,NDF,CO,BMI,BII,S,X,UI,BIN,BIM,
     1                N,NS,NPMAX)
      DIMENSION  NT(N,1),NR(N,1),NUMNP(1),NDF(1),
     1  CO(N,NPMAX,1),BMI(N,1),BII(N,3,1),S(N,3,1),X(1),UI(N,NPMAX,1),
     1  BIN(N,3,1),BIM(N,3,1),BII1(3,3),BIN1(3,3),BIM1(3,3)
C     ************************************************************
C     *                                                          *
C     *   SET UP IVERTIA DYADIC FOR RIGID AND FLEXIBLE BODIES    *
C     *                                                          *
C     ************************************************************
      INRT=0
      DO 100 K=1,N
         INRT=INRT+NR(K,1)+NT(K,1)
 100  CONTINUE
      II=INRT
      DO 200 K=1,N
         NPK=NUMNP(K)
         IF(NPK .EQ. 0) GOTO 200
         NFK=NDF(K)
         INDF=1
         IF(NDF(K) .EQ. 2) INDF=2
         IF(NDF(K) .EQ. 1) INDF=3
         DO 190 I=1,NPK
            DO 170 M=1,3
               UI(K,I,M)=0.
 170        CONTINUE
            DO 180 M=1,NFK
               UI(K,I,M+INDF-1)=X(II+M)
 180        CONTINUE
            II=II+NFK
C            WRITE(6,2001) (UI(K,I,M),M=1,3)
 190     CONTINUE
 200  CONTINUE
C
```

```
C     SET UP BIN(K,3,3),BIM(K,3,3)
C
      DO 300 K=1,N
         NPK=NUMNP(K)
         DO  210 L=1,3
          DO 210 M=1,3
             BIN(K,L,M)=0.
             BIM(K,L,M)=0.
 210     CONTINUE
         IF(NPK .EQ. 0) GOTO 300
         DO 280 I=1,NPK
            DO 240 M=1,3
               M1=M+1
               IF(M1 .EQ. 4) M1=1
               M2=M1+1
               IF(M2 .EQ. 4) M2=1
               BIN(K,M,M)=BIN(K,M,M)+BMI(K,I)*(CO(K,I,M1)*UI(K,I,M1)
     1                                +CO(K,I,M2)*UI(K,I,M2))
 240        CONTINUE
            DO 260 L=1,3
               DO 250 M=1,3
                  IF(M .EQ. L) GOTO 250
                  BIN(K,L,M)=-BMI(K,I)*CO(K,I,L)*UI(K,I,M)
 250           CONTINUE
 260        CONTINUE
 280     CONTINUE
         DO  290 L=1,3
          DO 290 M=1,3
             BIM(K,L,M)=BIN(K,M,L)
 290     CONTINUE
 300  CONTINUE
C
C     COORDINATE TRANSFORMATION FROM LOCAL TO GLOBLE
C
      DO 400 K=1,N
         DO  320 M=1,3
          DO 320 J=1,3
             BII1(M,J)=0.
             BIN1(M,J)=0.
```

```
            BIM1(M,J)=0.
            DO 310 I=1,3
C               IF( S(K,M,I) .LT. 1.E-10 ) S(K,M,I)=0.
              BII1(M,J)=BII1(M,J)+S(K,M,I)*BII(K,I,J)
              BIN1(M,J)=BIN1(M,J)+S(K,M,I)*BIN(K,I,J)
              BIM1(M,J)=BIM1(M,J)+S(K,M,I)*BIM(K,I,J)
 310        CONTINUE
 320    CONTINUE
        DO  350 I=1,3
         DO 350 M=1,3
            BIIJ=0.
            BINJ=0.
            BIMJ=0.
            DO 340 J=1,3
               BIIJ=BIIJ+BII1(I,J)*S(K,M,J)
               BINJ=BINJ+BIN1(I,J)*S(K,M,J)
               BIMJ=BIMJ+BIM1(I,J)*S(K,M,J)
 340        CONTINUE
            BII(K,I,M)=BIIJ
            BIN(K,I,M)=BINJ
            BIM(K,I,M)=BIMJ
 350    CONTINUE
 400  CONTINUE
 2001 FORMAT(5X,'=UI===',3F10.5)
 2002 FORMAT(5X,'-BII--',3F10.5)
 2003 FORMAT(5X,'-BIN--',3F10.5)
 2004 FORMAT(5X,'-BIM--',3F10.5)
 2005 FORMAT(5X,'=BII==',3F10.5)
 2006 FORMAT(5X,'=BIN==',3F10.5)
 2007 FORMAT(5X,'=BIM==',3F10.5)
      RETURN
      END

      SUBROUTINE FML(NT,NR,NUMNP,NDF,CO,BMI,BM,BII,S,W,WD,V,VI,VIM,
     1   VIN,UI,BIN,BIM,Y,VD,VID,A,AM,AMI,FM,N,NS,NDM,NPMAX)
      DIMENSION NT(N,1),NR(N,1),NUMNP(1),NDF(1),CO(N,NPMAX,1),BMI(N,1),
     2BM(1),BII(N,3,1),S(N,3,1),W(N,3,1),WD(N,3,1),V(N,3,1),SCO(3),
     3VI(N,NPMAX,3,1),VIM(3,1),VIN(3,1),
     4UI(N,NPMAX,1),BIN(N,3,1),BIM(N,3,1),Y(1),VD(N,3,1),
```

```
     5VID(N,NPMAX,3,1),A(NS,1),AM(3,1),AMI(3,1),FM(1),RU(3,3),UT(3,3),
     6WY(3,3),WB(3),W1(3),W2(3),W3(3),SUI(3),G(6,6)
C      ***********************************************************
C      *                                                         *
C      *         SET UP PART OF THE GENERALIZED LOAD VECTOR      *
C      *                                                         *
C      ***********************************************************
      INRT=0
      DO 100 K=1,N
         INRT=INRT+NR(K,1)+NT(K,1)
 100  CONTINUE
      DO 101 L=1,NS
         FM(L)=0.
 101  CONTINUE
      DO  11 I=1,NS
       DO 11 J=1,NS
          G(I,J)=0.
 11   CONTINUE
      DO 9000 K=1,N
         INDF=1
         IF(NDF(K) .EQ. 2) INDF=2
         IF(NDF(K) .EQ. 1) INDF=3
         NPK=NUMNP(K)
         NDFK=NDF(K)
         DO 120 M=1,3
            W1(M)=0.
            DO 110 J=1,NS
               W1(M)=W1(M)+WD(K,M,J)*Y(J)
 110        CONTINUE
 120     CONTINUE
         DO 140 M=1,3
            W2(M)=0.
            DO 130 I=1,3
               W2(M)=W2(M)+( BII(K,M,I)+BIN(K,M,I)+BIM(K,M,I) )*W1(I)
 130        CONTINUE
 140     CONTINUE
         DO 160 L=1,NS
            DO 150 M=1,3
               FM(L)=FM(L)+W2(M)*W(K,M,L)
```

```
150         CONTINUE
160      CONTINUE
         DO  1 I=1,NS
          DO 1 J=1,NS
             DO 2 M=1,3
                G(I,J)=G(I,J)+VD(K,I,M)*V(K,M,J)
2            CONTINUE
1        CONTINUE
         DO 3 I=1,NS
3        CONTINUE
         DO 220 M=1,3
            W1(M)=0.
            DO 210 J=1,NS
               W1(M)=W1(M)+VD(K,M,J)*Y(J)
210         CONTINUE
220      CONTINUE
         DO 260 L=1,NS
            DO 250 M=1,3
               FM(L)=FM(L)+BM(K)*W1(M)*V(K,M,L)
250         CONTINUE
260      CONTINUE
         DO 420 M=1,3
            WB(M)=0.
            DO 410 J=1,NS
               WB(M)=WB(M)+W(K,M,J)*Y(J)
410         CONTINUE
420      CONTINUE
         DO 430 I=1,3
            WY(I,I)=0.
430      CONTINUE
         WY(1,2)=-WB(3)
         WY(1,3)=+WB(2)
         WY(2,3)=-WB(1)
         WY(2,1)=+WB(3)
         WY(3,1)=-WB(2)
         WY(3,2)=+WB(1)
         DO 450 I=1,3
            W2(I)=0.
            DO 440 M=1,3
```

```
            W2(I)=W2(I)+( BII(K,I,M)+BIN(K,I,M)+BIM(K,I,M) )*WB(M)
440       CONTINUE
450     CONTINUE
        DO 470 M=1,3
          W3(M)=0.
          DO 460 I=1,3
            W3(M)=W3(M)+WY(M,I)*W2(I)
460       CONTINUE
470     CONTINUE
        DO 490 L=1,NS
          DO 480 M=1,3
            FM(L)=FM(L)+W3(M)*W(K,M,L)
480       CONTINUE
490     CONTINUE
        IF(NPK .EQ. 0) GOTO 9000
        DO 8000 IP=1,NPK
          DO 280 M=1,3
            SCO(M)=0.
            SUI(M)=0.
            DO 270 I=1,3
              SCO(M)=SCO(M)+S(K,M,I)*CO(K,IP,I)
              SUI(M)=SUI(M)+S(K,M,I)*UI(K,IP,I)
270         CONTINUE
280       CONTINUE
          DO 433 I=1,3
            RU(I,I)=0.
            UT(I,I)=0.
433       CONTINUE
          RU(1,2)=-SCO(3)-SUI(3)
          RU(1,3)=+SCO(2)+SUI(2)
          RU(2,3)=-SCO(1)-SUI(1)
          RU(2,1)=-RU(1,2)
          RU(3,1)=-RU(1,3)
          RU(3,2)=-RU(2,3)
          UT(1,2)=-SUI(3)
          UT(1,3)=+SUI(2)
          UT(2,3)=-SUI(1)
          UT(2,1)=-UT(1,2)
          UT(3,1)=-UT(1,3)
```

```
      UT(3,2)=-UT(2,3)
      DO 320 M=1,3
         W1(M)=0.
         DO 310 J=1,NS
            W1(M)=W1(M)+WD(K,M,J)*Y(J)
310      CONTINUE
320   CONTINUE
      DO 340 M=1,3
         W2(M)=0.
         DO 330 I=1,3
            W2(M)=W2(M)+RU(M,I)*W1(I)
330      CONTINUE
340   CONTINUE
      DO 360 L=1,NS
         DO 350 M=1,3
            FM(L)=FM(L)-BMI(K,IP)*W2(M)*V(K,M,L)
350      CONTINUE
360   CONTINUE
      DO 504 M=1,3
         DO 502 L=1,NS
            VIN(M,L)=0.
502      CONTINUE
504   CONTINUE
      DO 506 M=1,NDFK
         VIN(INDF-1+M,INRT+(IP-1)*NDF(K)+M)=1.
506   CONTINUE
      DO  508 M=1,3
       DO 508 L=1,NS
          VIM(M,L)=0.
          DO 507 I=1,3
             VIM(M,L)=VIM(M,L)+S(K,M,I)*VIN(I,L)
507       CONTINUE
508   CONTINUE
      DO 509 M=1,3
509   CONTINUE
      DO 520 M=1,3
         W1(M)=0.
         DO 510 J=1,NS
            W1(M)=W1(M)+VIM(M,J)*Y(J)
```

```
510            CONTINUE
520         CONTINUE
            DO  550 I=1,3
                W2(I)=0.
                DO 540 M=1,3
                   W2(I)=W2(I)+WY(I,M)*W1(M)
540             CONTINUE
550         CONTINUE
            DO 570 M=1,3
               W3(M)=0.
               DO 560 I=1,3
                  W3(M)=W3(M)+RU(M,I)*W2(I)
560            CONTINUE
570         CONTINUE
            DO 590 L=1,NS
               DO 580 M=1,3
                  FM(L)=FM(L)+2*BMI(K,IP)*W3(M)*W(K,M,L)
580            CONTINUE
590         CONTINUE
            DO 604 M=1,3
               W1(M)=0.
               DO 602 J=1,NS
                  W1(M)=W1(M)+WD(K,M,J)*Y(J)
602            CONTINUE
604         CONTINUE
            DO 620 M=1,3
               W2(M)=0.
               DO 610 I=1,3
                  W2(M)=W2(M)-UT(M,I)*W1(I)
610            CONTINUE
620         CONTINUE
            DO  650 I=1,3
                W3(I)=0.
                DO 640 M=1,3
                   W3(I)=W3(I)+UT(I,M)*W2(M)
640             CONTINUE
650         CONTINUE
            DO 690 L=1,NS
               DO 680 M=1,3
```

```
            FM(L)=FM(L)+BMI(K,IP)*W3(M)*W(K,M,L)
 680      CONTINUE
 690    CONTINUE
        DO 704 I=1,3
          W1(I)=0.
          DO 702 M=1,3
            W1(I)=W1(I)+WY(I,M)*SUI(M)
 702      CONTINUE
          IF( ABS( W1(I) ) .LT. 1.E-20 ) W1(I)=0.
 704    CONTINUE
        DO 720 M=1,3
          W2(M)=0.
          DO 710 I=1,3
            W2(M)=W2(M)+WY(M,I)*W1(I)
 710      CONTINUE
 720    CONTINUE
        DO  750 I=1,3
           W3(I)=0.
           DO 740 M=1,3
             W3(I)=W3(I)+UT(I,M)*W2(M)
 740       CONTINUE
 750    CONTINUE
        DO 790 L=1,NS
          DO 780 M=1,3
            FM(L)=FM(L)+BMI(K,IP)*W3(M)*W(K,M,L)
 780      CONTINUE
 790    CONTINUE
        DO 820 I=1,3
          W1(I)=0.
          DO 810 J=1,NS
            W1(I)=W1(I)+VD(K,I,J)*Y(J)
 810      CONTINUE
 820    CONTINUE
        DO 840 M=1,3
          W2(M)=0.
          DO 830 I=1,3
            W2(M)=W2(M)+RU(M,I)*W1(I)
 830      CONTINUE
 840    CONTINUE
```

```
      DO 860 L=1,NS
        DO 850 M=1,3
          FM(L)=FM(L)+BMI(K,IP)*W2(M)*W(K,M,L)
850     CONTINUE
860   CONTINUE
      DO 920 I=1,3
        W1(I)=0.
        DO 910 J=1,NS
          W1(I)=W1(I)+VD(K,I,J)*Y(J)
910     CONTINUE
920   CONTINUE
      DO 960 L=1,NS
        DO 950 M=1,3
          FM(L)=FM(L)+BMI(K,IP)*W1(M)*VIM(M,L)
950     CONTINUE
960   CONTINUE
      DO 1004 M=1,3
        W1(M)=0.
        DO 1002 J=1,NS
          W1(M)=W1(M)+WD(K,M,J)*Y(J)
1002    CONTINUE
1004  CONTINUE
      DO 1020 I=1,3
        W2(I)=0.
        DO 1010 M=1,3
          W2(I)=W2(I)+RU(I,M)*W1(M)
1010    CONTINUE
1020  CONTINUE
      DO 1040 L=1,NS
        DO 1030 M=1,3
          FM(L)=FM(L)-BMI(K,IP)*W2(M)*VIM(M,L)
1030    CONTINUE
1040  CONTINUE
      DO 1104 M=1,3
        W1(M)=0.
        DO 1102 J=1,NS
          W1(M)=W1(M)+VIM(M,J)*Y(J)
1102    CONTINUE
1104  CONTINUE
```

```
            DO 1120 I=1,3
               W2(I)=0.
               DO 1110 M=1,3
                  W2(I)=W2(I)+WY(I,M)*W1(M)
1110           CONTINUE
1120        CONTINUE
            DO 1140 L=1,NS
               DO 1130 M=1,3
                  FM(L)=FM(L)+2*BMI(K,IP)*W2(M)*( V(K,M,L)+VIM(M,L) )
1130           CONTINUE
1140        CONTINUE
            DO 1220 I=1,3
               W2(I)=0.
               DO 1210 M=1,3
                  W2(I)=W2(I)+WY(I,M)*( SCO(M)+SUI(M) )
1210           CONTINUE
               IF( ABS( W2(I) ) .LT. 1.E-20 ) W2(I)=0.
1220        CONTINUE
            DO 1240 M=1,3
               DO 1230 I=1,3
                  W3(M)=W3(M)+WY(M,I)*W2(I)
1230           CONTINUE
1240        CONTINUE
            DO 1260 L=1,NS
               DO 1250 M=1,3
                  FM(L)=FM(L)+BMI(K,IP)*W3(M)*( V(K,M,L)+VIM(M,L) )
1250           CONTINUE
1260        CONTINUE
            WRITE(1,2012) ( FM(L),L=1,NS )
8000     CONTINUE
         INRT=INRT+NDF(K)*NUMNP(K)
9000  CONTINUE
2001  FORMAT(2X,'=FM.1=',12F8.4)
2002  FORMAT(2X,'=FM.2=',12F8.4)
2003  FORMAT(2X,'=FM.3=',12F8.4)
2004  FORMAT(2X,'=FM.4=',12F8.4)
2005  FORMAT(2X,'=FM.5=',12F8.4)
2006  FORMAT(2X,'=FM.6=',12F8.4)
2007  FORMAT(2X,'=FM.7=',12F8.4)
```

```
2008 FORMAT(2X,'=FM.8=',12F8.4)
2009 FORMAT(2X,'=FM.9=',12F8.4)
2010 FORMAT(2X,'=FM.10',12F8.4)
2011 FORMAT(2X,'=FM.11',12F8.4)
2012 FORMAT(2X,'=FM.12',12F8.4)
3001 FORMAT(2X,'=VIM==',12F8.4)
3002 FORMAT(2X,'=SCO==',2I5,12F8.4)
3003 FORMAT(2X,'=SUI==',2I5,12F8.4)
6001 FORMAT('V=',6E11.4)
6002 FORMAT('VD',6E11.4)
6003 FORMAT('G=',6E11.4)
     RETURN
     END

     SUBROUTINE FCN(NN,T,Y,YPRIME)
     REAL T,Y(NN),YPRIME(NN)
     CALL YP(NN,T,Y,YPRIME)
     RETURN
     END

     SUBROUTINE FCNJ(NN,T,Y,PD)
     DIMENSION Y(NN),PD(NN,NN)
     RETURN
     END

     SUBROUTINE YP(NN,T,Y,F)
     REAL T,Y(1),F(1)
     COMMON/ROOM/MR(100000)
     COMMON/POINTO/N1,N2,N3,N4,N5,N6
     COMMON/POINTF/NF1,NF2,NF3,NF4,NF5,NF6,NF7,NF8
     COMMON/POINT1/N9,N10,N11,N12,N13,N14,N15,N16,N17,N18,N19,N20
     COMMON/POINT2/N21,N22,N23,N24,N25,NXL,N26,N27,N28,N29
     COMMON/POINT3/NUI,N31,N32,NYL,NYI,NYD,ND1,ND2,ND3
     COMMON/POINT4/N41,N42,N43,N44,N45,N46,N47
     COMMON/POINT5/N51,N52,N53,N54,N55
     COMMON/POINT6/N61,N62,N63,N64
     COMMON/POINT7/N71,N72,N73,N74,N75
     IF(T .EQ. 0.0) THEN
       DO 10 I=11,12
```

```
         F(I)=0.01
10     CONTINUE
      ENDIF
      CALL DEFY(20,MR(N1),MR(N2),MR(N3),MR(N4),MR(N5),MR(N17),
     1   MR(N20),MR(N21),MR(N22),MR(NXL),MR(N26),T,Y,YI,
     1   MR(NF7),MR(N41),MR(N44),MR(N47),MR(N51),MR(N52),F,
     1   2,6,2,50,100)
      RETURN
      END

      SUBROUTINE CDGEAR(Y,YI,N,NS)
      EXTERNAL   FCN,FCNJ
      DIMENSION  Y(1),YI(1),IWK(2),WK(1000),COMP(100)
      NN=NS*2
      T     =  0.0
      TOL   = .000001
      H     = .00001
      METH  =  1
      MITER =  0
      INDEX =  1
      DO 700 KK=1,500
         TEND=FLOAT(KK)/100.
333      CALL DGEAR (NN,FCN,FCNJ,T,H,Y,TEND,TOL,METH,MITER,
     1                  INDEX,IWK,WK,IER)
         IF(IER.GT.128) GO TO 800
         NS1=NS+1
         NS2=NS+2
         NS3=NS+3
         NS4=NS+4
         NS5=NS+5
         NS6=NS+6
         NS7=NS+7
         NS8=NS+8
         NS9=NS+9
         NS10=NS+10
         NS11=NS+11
         NS12=NS+12
         NS13=NS+13
         NS14=NS+14
```

```
         YS=Y(NS1)+Y(NN)
         XS=Y(NS1)+Y(NS2)
      WRITE(2,5001) T,Y(NS) ,Y(NS1),Y(NS2)
      WRITE(3,5001) T,Y(NS3),Y(NS5),Y(NS6)
 700  CONTINUE
      RETURN
 800  WRITE(3,5003) TOL,TEND,H,T,METH,MITER,INDEX
      WRITE(3,5004) (Y(I),I=1,NN)
      RETURN
 5001 FORMAT(2F12.7,5E15.6)
 5002 FORMAT(2F12.7,5E15.6)
 5003 FORMAT(2X,'TOL,XEND,H,,METH,MITER,INDEX',4F10.5,3I5)
 5004 FORMAT(2X,'**Y**',7F10.5)
 5005 FORMAT(5E14.6)
      END

      LOGICAL FUNCTION PCOMP(A,B)
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PCOMP = .FALSE.
C.... IT MAY BE NECESSARY TO REPLACE THE FOLLOWING ALPHANUMERIC
C.... COMPARISON STATEMENT IF COMPUTER PRODUCES AN OVERFLOW
      IF(A.EQ.B) PCOMP = .TRUE.
      RETURN
      END
      BLOCK DATA
      COMMON/QDATA/QO,QHEAD(20),IPR
      DATA QO/1HO/,IPR/2/
      END

      SUBROUTINE DESN(K,T,HM,N)
      DIMENSION HM(N,1)
      HM(K,1)=0.
      IF(T .LT. 0.05) HM(K,1)=1.
      IF(T .GT. 0.1 .AND. T .LT. 0.15) HM(K,1)=-1.
      RETURN
      END
      SUBROUTINE FT1(X,Y,T,F,N,NS)
      DIMENSION X(1),Y(1),F(1)
      COMMON/ROOM/MR(100000)
```

```
      COMMON/POINT3/NUI,N31,N32,NYL,NYI,NYD,ND1,ND2,ND3
      COMMON/POINT1/N9,N10,N11,N12,N13,N14,N15,N16,N17,N18,N19,N20
8001  FORMAT(5X,'--X--',8F13.8)
      DO 100 L=1,NS
         F(L)=0.
100   CONTINUE
      BK11=19.6
      G   =9.8
      BM  =0.2
      BL  =0.4
      W1=SQRT(BK11/BM)
      W2=SQRT(G/BL)
      WRITE(1,2001) W1,W2
      I =0
      I1=I+1
      I2=I+2
      I3=I+3
      I4=I+4
      F(I1)= -W1**2*Y(I3)+(1.+Y(I3))*(Y(I2)**2)-W2**2*(1-COS(Y(I4)))
      F(I2)=-(2./(1.+Y(I3)))*Y(I1)*Y(I2)
     1                        -W2**2*( 1./(1.+Y(I3)) )*SIN(Y(I4))
2001  FORMAT(2X,'=W1--W2=',7F10.5)
2004  FORMAT(/2X,'==F===',7F10.5)
       RETURN
       END

      SUBROUTINE ADNL(N,BM,BC,BK,BK1,R,R1,U,UD,UDD,U1,U1D,U1DD,
     1               BMN,BCN,BKN,RN,U2,U2D,U2DD )
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  BM(N,1),BC(N,1),BK(N,1),BK1(N,1),R(1),R1(1),
     1           U(1),UD(1),UDD(1),U1(1),U1D(1),U1DD(1),
     2           BMN(N,1),BCN(N,1),BKN(N,1),RN(1),U2(1),U2D(1),U2DD(1)
      COMMON/ROOM/W(10000)
      COMMON/A0A7/A0,A1,A2,A3,A4,A5,A6,A7,TOL
      COMMON/N0112/N1 ,N2 ,N3 ,N4 ,N5 ,N6 ,N7 ,N8 ,N9 ,N10,N11,N12
      COMMON/N2127/N21,N22,N23,N24,N25,N26,N27
      DO  100 I=1,N
       DO 100 J=1,N
          BMN(I,J)=BM(I,J)
```

```
        BCN(I,J)=BC(I,J)
        BKN(I,J)=BK(I,J)
100   CONTINUE
      II=0
111   II=II+1
      BMN(1,1)=0.2*(0.4+U1(N))**2
      BKN(1,1)=0.2*9.8*(0.4+U1(N))
      RN(1)=-2*0.2*(0.4+U1(N))*U1D(1)*U1D(N)
      RN(N)=   0.2*(0.4+U1(N))*U1D(1)*U1D(1)-0.2*9.8*( 1.-DCOS(U1(1)) )
      DO 130 I=1,N
         DO 120 J=1,N
            R1(I)=R1(I)+BMN(I,J)*( A0*U(J)+A2*UD(J)+A3*UDD(J) )
     1                  +BCN(I,J)*( A1*U(J)+A4*UD(J)+A5*UDD(J) )
120      CONTINUE
         R1(I)=R1(I)+R(I)+RN(I)
130   CONTINUE
      DO  110 I=1,N
       DO 110 J=1,N
          BK1(I,J)=BKN(I,J)+A0*BM(I,J)+A1*BC(I,J)
110   CONTINUE
      CALL GAUSS(N,BK1,R1,ID)
      WRITE(6,*)  'ID=', ID
      WRITE(6,*) 'R1==', ( R1(I),I=1,N )
      DO 140 I=1,N
         U2  (I)=R1(I)
         U2DD(I)=A0*( U2(I)-U(I) )-A2*UD(I) -A3*UDD (I)
         U2D (I)=UD(I)             +A6*UDD(I)+A7*U2DD(I)
140   CONTINUE
      A1=DABS ( (U2(1)-U1(1))/U1(1) )
      A2=DABS ( (U2(N)-U1(N))/U1(N) )
      WRITE(6,*) 'A1,A2===', A1,A2
      AM=DMAX1( A1,A2 )
      DO 150 I=1,N
         U1  (I)=U2  (I)
         U1D (I)=U2D (I)
         U1DD(I)=U2DD(I)
150   CONTINUE
      IF( AM .GT. TOL ) GOTO  111
1001  FORMAT(F10.5)
```

```
 2001 FORMAT(2X,'==TOL==',F10.5)
      RETURN
      END

      FUNCTION   DOT(A,B,N)
      IMPLICIT REAL*8(A-H,O-Z)
C      GENERIC
C.... VECTOR DOT PRODUCT
C
      DIMENSION A(1),B(1)
      DOT = 0.0
      DO 100 I = 1,N
 100  DOT = DOT + A(I)*B(I)*1.E20
      RETURN
      END
```