Spring 5-31-1988

Modeling and simulation of robot arms with flexible links

John G. Danis
New Jersey Institute of Technology

ABSTRACT

Title of Thesis: Modeling and Simulation of Robot Arms with Flexible Links

John G. Danis, Master of Science, 1988

Thesis directed by: Dr. Frederick D. Chichester F.D.C.

The design of lightweight robot arms introduces a degree of flexiblity in the individual links which renders the arm difficult to control. Solution of the control problem requires accurate and detailed mathematical models of the arm dynamics. A comprehensive survey of the current literature in this area has shown that although many such models exist, there is a great diversity in their structure, function, and applicability. The different objectives and techniques of model development which lead to this diversity are examined and summarized in this thesis. Bases for classification of the mathematical models and techniques of development are established, and a general development methodology is proposed for each class of model. Computer simulations of relevant portions of the model development are used to support these general development methodologies. The model development and classification processes are demonstrated by their application to several current models.

MODELING AND SIMULATION OF ROBOT ARMS
WITH FLEXIBLE LINKS

by
John G. Danis

Thesis submitted to the Faculty of the Graduate School of
the New Jersey Institue of Technology in partial fulfillment of
the requirements for the degree of
Master of Science in Electrical Engineering
1988

APPROVAL SHEET

Title of Thesis: Modeling and Simulation of Robot Arms with Flexible Links

Name of Candidate: John G. Danis
Master of Science, 1988

Thesis and Abstract Approved: ______________________ ________
Dr. F. D. Chichester Date

Thesis Committee members: ______________________ ________
Date

______________________ ________
Date

VITA

Name: John G. Danis

Degree and date to be conferred: MSEE, May, 1989

Secondary Education: Garfield High School, 1980

Collegiate institutions attended:	Degree:	Date of Degree:
New Jersey Institute of Technology	MSEE	1989
New Jersey Institute of Technology	BSEE	1985

Major: Electrical Engineering

Publications: "Modeling the Dynamics of Flexible Robot Arms", F. Chichester and J. Danis, Proceedings of the Nineteenth Annual Pittsburgh Conference on Modeling and Simulation, University of Pittsburgh, May, 1988

"Classification of Dynamics Models of Robot Arms with Flexible Links", F. Chichester and J. Danis, Proceedings of Advances in Communication and Control Systems, Louisiana State University, Oct., 1988

Positions held: Inertial Systems Analyst, Kearfott Guidance and Navigation Corp., Wayne, NJ

Blank Page

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Robotic arms constructed of lightweight links have many advantages over their heavier counterparts such as increased speed and manueverability, lower energy consumption, and reduced costs. However, the lighter weight results in increased link compliance to the point where flexibility effects can no longer be ignored when controlling the arm if high accuracy is required. The development of control systems of robotic arms with flexible, lightweight links requires detailed mathematical models of the arm dynamics so that the appropriate control laws may be formulated. Once developed, the models may be used to verify the control laws and test the acceptability of system response under simulated operating conditions. The models may also serve as the foundation for developing sophisticated real-time computer simulations of a flexible robot arm performing various tasks.

The objective of the work reported in this thesis was to examine the current mathematical modeling and simulation of flexible robotic arms and classify these models according to relevant characteristics. An extensive survey of recent literature in this area was conducted, and existing models were examined as to their modeling objectives, development, structure, and applicability. These areas were investigated and used to establish bases for classification of such models and to formulate general model development procedures. Selected portions of these model development procedures were demonstrated on a personal computer.

II. OBJECTIVES OF MODEL DEVELOPMENT

Mathematical models of flexible robotic arms are developed to investigate certain aspects of a particular arm's behavior under specified conditions. The examination of those specific aspects of behavior defines the objective of the modeling. The objectives of the model must focus on certain aspects to the neglect of others due to the inherent complexity of modeling robotic arms with flexible links.

Examination of existing models suggests that the various aspects of behavior may be divided into two general areas, from which the basic modeling objectives follow. The first basic objective is to develop the equations of motion of the entire arm, taking into account the flexible deflections, usually by incorporating previously calculated or assumed vibratory modes. Those models developed to accomplish this or some variation of this objective can be classified as kinematic models. The second basic modeling objective is the characterization of the vibratory modes of a single link of the arm by examining its eigenvalues and eigenfunctions. Models which have this as their primary objective are classified as structural models. This distinction between kinematic models and structural models, based upon model objectives, forms the most general classification of mathematical models of robot arms with flexible links established in the work reported here. Subdivisions of each of these classes are based upon specific techniques used in the modeling process.

III. KINEMATIC MODELS

1. Foundations

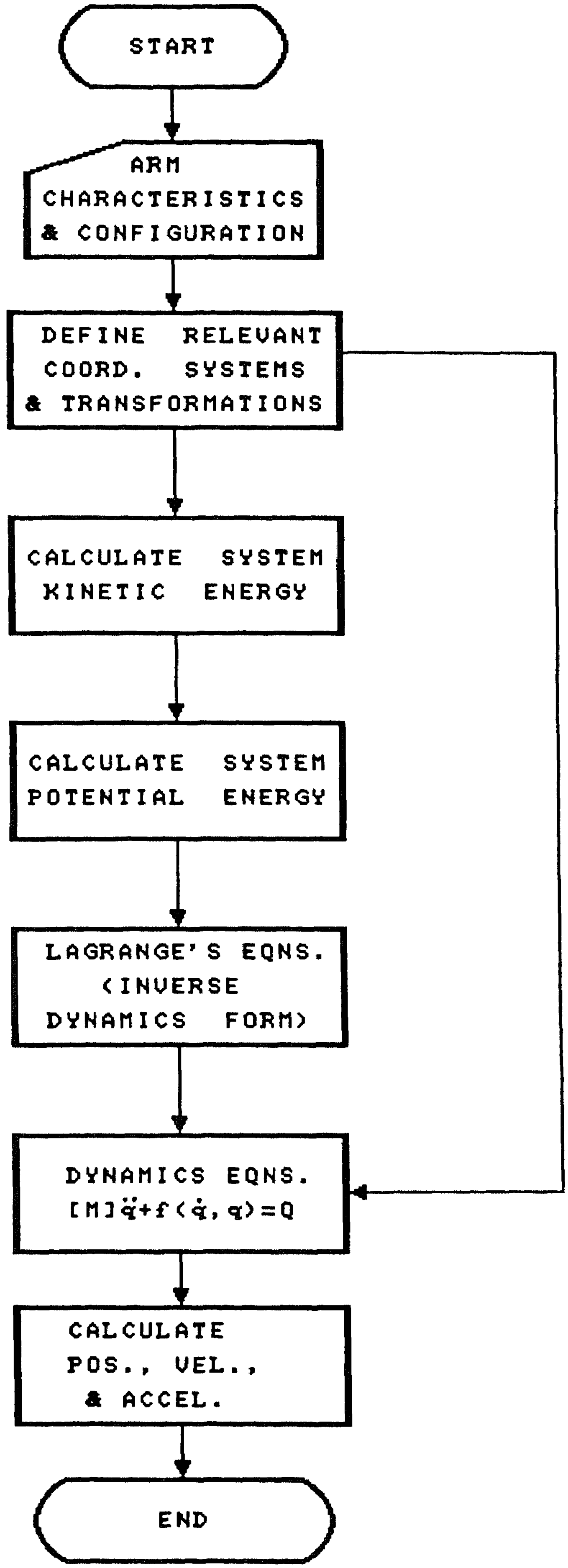
Kinematic models of robot arms with flexible links basically consist of sets of equations which describe the rigid-body motion and flexible deflections of the arm. Emphasis is placed on the examination of the kinematic aspects of the arm motion; that is, being able to describe the position, velocity, and acceleration of points on the arm as a function of initial conditions and applied forces. It is the effects of the flexible deflections on these parameters which are being investigated, not the structural properties of the arm which give rise to those deflections. Accordingly, the vibratory modes used in the model are either assumed or previously calculated.

The generation of the equations of motion for robot arms with flexible links is in many ways similar to that for robot arms with rigid links. In fact, many kinematic models of robot arms with flexible links were derived from similar rigid link models which, for the sake of simplicity, neglected flexibility of the links. To improve accuracy, flexibility of the links is taken into account.

2. Model Development

Examination of existing kinematic models shows that there is a large degree of similarity for all kinematic models because they are all based on the same underlying concepts. In fact, it is possible to specify a general model development procedure for most mathematical models which investigate the kinematic aspects of a robot arm with flexible links. Such a model development algorithm is shown in Figure III - 1. This algorithm highlights

FIGURE III - 1 : KINEMATIC ARM MODEL DEVELOPMENT ALGORITHM



the main points of kinematic model development and serves as a guide in the future development of kinematic models.

A kinematic model is a state representation of the system, typically using generalized coordinates to define the state of the system. The generalized coordinates may include variables related to joint angles and variables which describe the link flexure. Generally, kinematic models are models of multi-link arms and it is necessary to define these coordinates for each link, which requires defining a coordinate frame for each link. One frequently used method of defining link coordinate frames and homogeneous transformations between coordinate frames is the technique developed by Denavit and Hartenberg, described below [11].

A robotic arm with n degrees of freedom will have n links and n joints. A right-handed orthogonal coordinate frame is assigned to each link, with the position of the origin and direction of the coordinate axes determined by the type of joints associated with the link, and their respective orientations. A coordinate frame also is assigned to the base of the arm, which is sometimes referred to as link 0. It often is necessary to define motions with respect to this coordinate frame, and for this reason, homogeneous transformations between links must be defined.

A homogeneous transformation of coordinates describes the position and orientation of the coordinate frame of one link with respect to the coordinate frame of another link. These transformation matrices are often designated A matrices in robotics analyses [11]. A_n describes the position and orientation of the coordinate frame of link n with respect to that of link n - 1. The physical significance of an A matrix is that it represents the translations and rotations necessary to make the n - 1 coordinate frame coincident with the n frame. By the multiplication of successive transformations, the position and orientation of the n th link with respect to the base can be shown.

$$T_n = A_1 \ A_2 \ \ldots \ A_{n-1} \ A_n$$

The matrix which transforms the coordinate frame of the last link to that of the end effector is denoted as E. Therefore, for a robotic arm with n links, the transformation from the base coordinate frame to that of the end effector is the product of the transformations between all links, and is given as T_e .

$$T_e = A_1 \ A_2 \ \ldots \ A_{n-1} \ A_n \ E$$

Establishing link coordinate frames and the transformations discussed above are the necessary first steps in developing a kinematic model of a robotic arm. Given the arm dimensions and configuration, this process can be carried out readily by a computer program. A short FORTRAN program was written to demonstrate this. (See Appendix A.) The program calculates the A matrices for a hypothetical serial manipulator and demonstrates how T_e, the transformation from the base to the end effector, is obtained. This program then could be integrated into a larger computer simulation of a kinematic model, serving as the first steps in the model development.

The method of defining link coordinate systems and homogeneous transformations described above originally was developed for rigid link models. However, with some modifications to include the effects of the flexible deflections of each link, it also can be applied to robot arms with flexible links. Each A matrix can be multiplied by a matrix F which represents the small change in position and orientation of the link due to small flexible deflections. The transformation from the coordinate frame of link n - 1 to that of link n then becomes $A_n F_n$. When this is done for each link, the transformation from the base coordinate frame to the end effector coordinate frame includes the cumulative effect of all the links, as shown below.

$$T_e = A_1 \, F_1 \, A_2 \, F_2 \, \ldots \, A_{n-1} \, F_{n-1} \, A_n \, F_n \, E \, F_e$$

The F matrix for each link represents the small displacements Δx, Δy, and Δz, and the small angle rotations, θ's, of the n coordinate frame due to link flexure. The F_n matrix for link n is as follows.

$$F_n = \begin{bmatrix} 1 & -\theta_{zn} & \theta_{yn} & \Delta x \\ \theta_{zn} & 1 & -\theta_{xn} & \Delta y \\ -\theta_{yn} & \theta_{xn} & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The elements of F_n represent the net effect of all the vibratory modes included in the model for the n th link. Note that as the small angle rotations and displacements approach zero, the F_n matrix approaches the identity matrix. This implies that if the flexibility effects for each link were to be reduced to zero, each F matrix would become an identity matrix, and the resulting model would be identical to the rigid link model developed earlier. This is an important result because it suggests that flexibility effects may be added to existing rigid link models to produce models of flexible robotic arms.

The elements of F_n also are time varying parameters, and a given F_n represents the flexure of the link at some instant in time. The calculation of the F matrices and their incorporation into the homogeneous transformation of coordinates of the links also can be accomplished by a short computer program. (This is shown in Appendix A.)

With the link coordinate frames and transformations defined, the equations of motion of the robotic arm can now be developed. Different methods of formulation are possible, but Lagrange's formulation of the system dynamics is utilized most often in investigating the kinematic aspects of flexible robotic arms. The formulation of Lagrange's equations requires expressions for the kinetic and potential energies of the system. This is the next basic step in the development of flexible arm kinematic models.

Kinetic energy is defined first for a point or small element of a single link, and integrated over the length of that link to yield the total contribution of that link to the system kinetic energy. The kinetic energy of a point on the n th link is given by:

$$dK_n = \frac{1}{2} dm_n \dot{r}_n^T \dot{r}_n$$

where dm_n is the point mass, r_n is the absolute position vector and $\dot{r}_n$ is the absolute velocity vector of the point on the link in base coordinates. It follows that the kinetic energy for the link is:

$$K_n = \int_0^{l_n} dK_n$$

kinetic energy of the system is obtained by summing the contribution from each link.

$$K_{SYS} = \sum_{n=1}^{m} K_n$$

where m is the total number of links of the robotic arm.

The potential energy of a flexible link is due to gravity and elasticity of the link, sometimes referred to as strain energy. Potential energy due to elasticity of the link is a function of the material and construction of the link as well as the magnitude of the deflection. Potential energy due to gravity is formulated for a point, and then integrated over the length of the link. Potential energies of all the links are then summed to yield the total potential energy of the system.

Once the kinetic and potential energies of the system have been obtained, the various derivatives and partial derivatives are taken with respect to the generalized coordinates. With expressions for all the necessary terms available, Lagrange's equations of motion may then be formulated in the inverse dynamic form as shown below.

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_j}\right) - \left(\frac{\partial K}{\partial q_j}\right) + \left(\frac{\partial P}{\partial q_j}\right) = Q_j$$

where K is the system kinetic energy, P is the system potential energy, q_j are the generalized coordinates for j degrees of freedom, and Q_j are the generalized forces (torque applied by the joint actuators). Depending upon the specific formulation of the model, the generalized coordinates may include variables related to joint angles and variables related to the flexible deflections. In this form, the model may be used to study the inverse kinematic problem of robot arms with flexible links.

Since it is desired to model the arm's behavior as a response to known initial conditions and specified control inputs, this formulation of the arm kinematics must be converted to its dynamic form. This involves the formulation of an inertia or mass matrix, which is multiplied by the second derivative with respect to time of the state vector, where the elements of the state vector are the generalized coordinates. The resulting model is shown below.

$$[M]\ddot{q} - f(\dot{q},q) = Q$$

where [M] is the inertia matrix, Q is the generalized input vector, and f is a vector of nonlinear functions of $\dot{q}$ and q. This is the general form of most kinematic models and their corresponding computer simulations. The complexity of the model is determined by such factors as the number of links, the number of modes included, the amplitude of the vibrations, and nonlinear effects present.

Kinematic models have a high degree of applicability in problems involving the determination of the arm's response under varying operating conditions. Thus they are well suited for solution of inverse dynamics problems, trajectory planning, workspace and task analysis, and related applications.

IV. STRUCTURAL MODELS

1. Foundations

Structural models of robot arms with flexible links are used to determine the elastic deformation of the links as a function of time and position along the length of the undeformed link. These models use the material properties and physical configuration of the arm to obtain the mode frequencies and associated mode shapes of the arm. The model must be truncated at some point, so only those modes of interest are retained in the final model formulation. As with kinematic models, the model formulation of structural models follows a general pattern, allowing a model development algorithm to be constructed. Such an algorithm is shown in Figure IV - 1, and can be used to aid in the development of future structural models.

2. Model Development

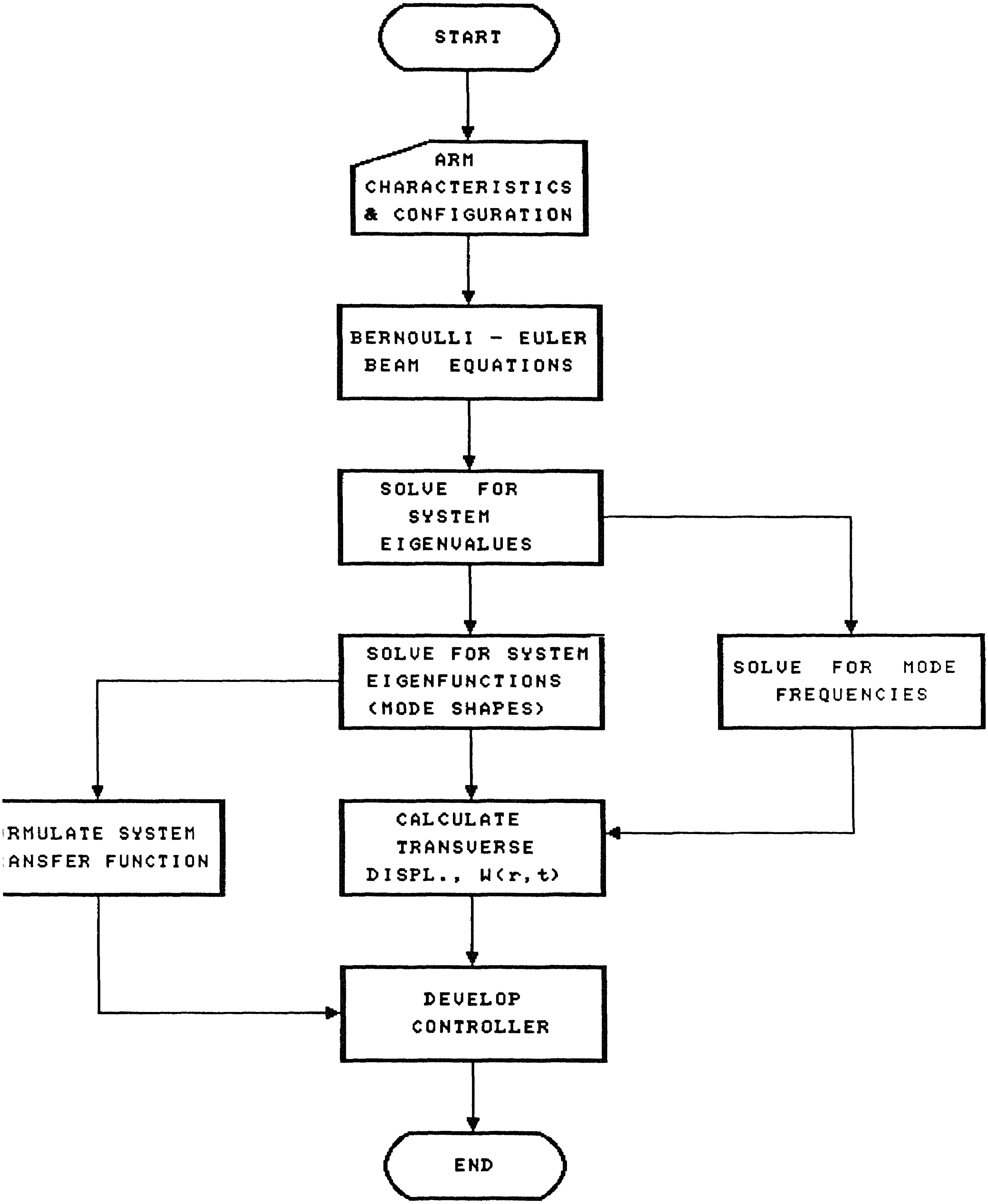
The Bernoulli-Euler beam equations form the basis for the development of structural models. In its simplest form, a Bernoulli-Euler beam is modeled by the partial differential equation shown below.

$$\frac{\partial^2 w}{\partial t^2} = - \frac{EI}{\rho} \frac{\partial^4 w}{\partial x^4}$$

where w is the transverse displacement of the link, EI is the bending stiffness, ρ is the mass per unit length, and x is the distance along the axis of the undeformed link.

The boundary conditions used in describing the system are determined by the configuration of the arm and reflect the constraints at each end of the link. Most structural models of

FIGURE IV - 1 : STRUCTURAL ARM MODEL DEVELOPMENT ALGORITHM



robot arms with flexible links are initially developed as a single flexible clamped-free beam. As a result, the transverse displacement and slope at the base of the link, along with the bending moment and shear force at the tip of the link, are required to be zero, producing the following boundary conditions.

$$w(0,t) = w'(0,t) = w''(\ell,t) = w'''(\ell,t) = 0$$

Additional boundary conditions must be specified if there is a load at the tip of the link.

The transverse displacement of normal mode vibrations can be expressed as:

$$w(x,t) = \phi(x)\,\sin(\omega t + \epsilon)$$

Substituting this into the Bernoulli-Euler beam equation and assuming a uniform beam, so that E and I are constant along the length of the beam, yields the ordinary differential equation below.

$$\omega^2 \phi = (EI/\rho)\,\phi''''$$

These equations, along with the associated boundary conditions for the links, form the eigenvalue problem whose solutions will characterize the vibratory modes of the links. The general solution of this eigenvalue problem is of the form:

$$\phi(x) = C_1 \cosh(\lambda x/\ell) + C_2 \sinh(\lambda x/\ell) + C_3 \cos(\lambda x/\ell) + C_4 \sin(\lambda x/\ell)$$

where: $\lambda = \ell(\rho\,\omega^2/EI)^{1/4}$

ℓ = length of the link

The eigenvalues are determined by the system boundary conditions and, for the basic clamped-free beam, they are the solutions to the following equation, sometimes referred to as the frequency equation [5].

$$\cosh\lambda \, \cos\lambda + 1 = 0$$

Once an eigenvalue is known, it may be used to solve directly for its associated mode frequency. For the clamped-free beam, mode frequencies are given by:

$$\omega_n = (\lambda_n^2 / \ell^2) \sqrt{EI/\rho}$$

where ω_n is the radian frequency of the n th mode, λ_n is its associated eigenvalue, and ℓ is the length of the beam. When λ_n is substituted into the expression for $\phi(x)$, the mode shape associated with that eigenvalue is produced.

The net transverse displacement actually is the sum of the contributions from each vibratory mode. Since a distributed parameter model of a flexible link is an infinite-dimensional system, with an infinite number of vibratory modes present, the system must be approximated by truncating the vibratory modes at some point. The transverse displacement may then be expressed as:

$$w(x,t) = \sum_{n=1}^{m} \phi_n(x) \; (A_n \cos\omega_n t + B_n \sin\omega_n t)$$

where m is the number of included modes, and A_n and B_n are defined below [5].

$$A_n = \frac{1}{m_n} \int_0^\ell w_o \, \rho \, \phi_n \, dx$$

$$B_n = \frac{1}{\omega_n m_n} \int_0^\ell \dot{w}_o \, \rho \, \phi_n \, dx$$

where w_o and $\dot{w}_o$ are the initial displacement and velocity of the tip of the link, and

$$m_n = \int_0^\ell \rho \, \phi_n^2 \, dx \qquad = \text{generalized mass}$$

Several short FORTRAN programs were written to demonstrate the procedure for formulation of a basic structural model. (See

Appendix B.) It is shown in these simulations how the mode frequencies and shapes are determined from the eigenvalues and physical properties of the links. Also shown are the methods by which the total transverse displacement is determined, and the motion of the tip of the link is simulated.

Most current structural models, after being used to examine the transverse displacement of the link by characterizing its vibratory modes, are used in the development of systems to control the flexible deflections. All control systems developed by using structural models have the same basic objective, which is the stabilization of the arm by damping out deflections due to elastic deformation as quickly as possible. Two basic methods of control are used very frequently in current models of robotic arms. One involves the development and implementation of a dynamic compensator which controls joint actuator torques in such a way that flexible deflections are quickly damped out [12]. The other method requires having two coaxially-mounted links with a force actuator mounted in between them to actively damp out any deflection [3],[18]. Further development of structural models entails the examination of the controllability of the arm, and simulation of the influence of the control system on the dynamic behavior of the arm.

The modeling objectives and techniques employed in the development of structural models render them well-suited for use in control system design. The system equations used to characterize the vibratory modes also can be used to derive a transfer function for the arm, which serves as the plant for the control system. The angular acceleration (actuator torque) is the control input, and the beam bending moment, measured at the base is the output. Structural models also have applications in structural analysis, stress analysis, and performance analysis.

V. CLASSIFICATION AND ANALYSIS OF CURRENT MODELS

The distinction between kinematic models and structural models, along with other bases of classification, resulted from the careful examination of many existing models. Variations in modeling objectives, techniques, and intended applications provide much information about the basic process of modeling robot arms with flexible links. One of the objectives of this effort was to study existing models and determine which modeling techniques are best suited for a given situation. These points can be investigated further by a detailed review of some current models.

1. Book Model (1984)

A nonlinear distributed parameter model of a robot arm consisting of several flexible links was developed in 1984 by Book [1]. The objective of the model is to simulate the equations of motion of the arm, and ultimately express the state of the system as a function of time for known initial conditions and specified control inputs. With this as the modeling objective, the model is therefore classified as a kinematic model.

The model development is based on describing the flexible deflections, as well as the joint angles, in terms of 4x4 transformation matrices. Variables representing the time-varying amplitudes of the vibratory modes, together with variables representing the joint angles, comprise the state vector of the system. This time-varying state vector is expressed as a function of inputs and initial conditions after Lagrange's equations have been formulated and converted into simulation form. It is the determination of this state vector for which the model is developed.

The transformations used to describe the link flexure are composed of assumed modal shapes for each link. Small deflections of the link are assumed, and are expressed in terms of a link-based coordinate frame as:

$$ {}^{i}h_i(\eta) = \begin{bmatrix} 1 \\ \eta \\ 0 \\ 0 \end{bmatrix} + \sum_{j=1}^{m} \delta_{ij} \begin{bmatrix} 0 \\ x_{ij}(\eta) \\ y_{ij}(\eta) \\ z_{ij}(\eta) \end{bmatrix} $$

where η is the distance from the joint along the axis of the undeformed link, δ_{ij} is the time-varying amplitude of the displacement, and m is the number of modes being modeled. The transformation matrix of the link deflection is expressed as:

$$ E_i = [\ H_i + \sum_{j=1}^{m} \delta_{ij} M_{ij}\ ] $$

where:

$$ H_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_i & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

and:

$$ M_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ x_{ij} & 0 & -\theta_{zij} & \theta_{yij} \\ y_{ij} & \theta_{zij} & 0 & -\theta_{xij} \\ z_{ij} & -\theta_{yij} & \theta_{xij} & 0 \end{bmatrix} $$

where x_{ij}, y_{ij}, and z_{ij} represent the displacement of the link due to the vibratory mode and the Θ's are small angle rotations representing any twisting of the link.

Once the transformations for all the joints and all the links have been defined, along with their various derivatives and partial derivatives, Lagrange's equations are developed. In the formulation of Lagrange's equations, the system kinetic and potential energies must be obtained. Kinetic energy is due to rotation of the joints and the small motions of the vibrating links. Potential energy of the arm is due to gravity and energy stored in the elastic deformation of the flexible links.

Once formulated, Lagrange's equations with respect to the joint variables become:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_j}\right) - \left(\frac{\partial K}{\partial q_j}\right) + \frac{\partial V_e}{\partial q_j} + \frac{\partial V_g}{\partial q_j} = F_j$$

The equations with respect to the vibratory mode variables are expressed as:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\delta}_{jf}}\right) - \left(\frac{\partial K}{\partial \delta_{jf}}\right) + \frac{\partial V_e}{\partial \delta_{jf}} + \frac{\partial V_g}{\partial \delta_{jf}} = 0$$

Inertia coefficients of the state variables are obtained from these equations, and a square inertia matrix is formed. This inertia matrix is used in converting the equations to their dynamic form for simulation, shown below.

$$J\ddot{z} = R$$

where J is the inertia matrix, z is the state vector, and R is the vector of control inputs and remaining dynamics. This model is formulated in a very efficient manner and, due to its recursive properties, it can be simulated easily. The accuracy of the model can be improved, at some cost in computational efficiency, by increasing the number of modes.

2. Hastings/Book Model (1987)

A linear distributed parameter model of a single flexible link was formulated by Hastings and Book in 1987 [6]. The modeling objective was to develop a dynamic model of a flexible link that would simulate the rigid body motion and the flexible deflections. Hence, this model can be considered a kinematic model.

Since this kinematic model is developed for a single link, only one coordinate frame (at the joint) and no homogeneous transformations of coordinates need be defined. Consequently, the position of each point on the link is a function of the joint angle, θ, and displacement due to the flexible deflection, $w(x,t)$.

The kinetic energy of the link and the potential energy due to the elastic deformation are calculated, and Lagrange's equations of motion are then formulated as shown below.

$$\frac{d}{dt}\left|\frac{\partial K}{\partial \dot{z}_i}\right| - \frac{\partial P}{\partial z_i} = Q_i$$

where Q_i are the generalized forces (inputs), and z_i are the generalized coordinates. The generalized coordinates include the joint angle and variables related to the vibratory modes.

$$z = [\,\theta,\ q_1(t),\ q_2(t),\ \ldots\ ,\ q_n(t)\,]$$

for the n modes included in the model.

The equations of motion are converted to their dynamic form, given by:

$$[M]\ddot{z} + [K]z = [Q]$$

where [M] is the inertia matrix, [K] is a matrix representing bending stiffness, and [Q] is the control input.

An important aspect of this model is that the vibratory modes included are assumed or previously calculated. Their effect on the equations of motion and the response of the arm are the primary objectives of the simulation. The model was later validated by an experimental setup in a laboratory environment.

3. Usoro/Nadira/Mahil Model (1986)

This model, developed by Usoro, Nadira, and Mahil in 1986 [15], uses finite element techniques in the development of Lagrange's equations of motion for a robotic arm with two flexible links. (These same techniques could be expanded to model an arm of any number of links.) Since the model development places emphasis on incorporating the flexible deflections into the equations of motion, it can be classified as a kinematic model.

The model is based upon treating each link as a finite number of small flexible, distributed elements, with the flexible deflection of each element expressed in terms of assumed shape functions. Use of these shape functions (Hermitian polynomials) requires some prior knowledge of the nature of the arm flexure, that is, the magnitude of the displacement and slope of the flexed link at various points along each link. Those points serve as the interfaces between adjacent elements.

Kinetic and potential energies are calculated in generalized coordinates for each element. The contributions from all the elements are summed to yield the kinetic and potential energies for each link. Lagrange's equations are used to produce the dynamic equations for the arm, which are of the form:

$$[M]\ddot{q} - f(\dot{q},q) = Q$$

where [M] is the generalized inertia matrix, Q is the generalized input vector, and f is a vector of nonlinear functions of $\dot{q}$ and q.

The complexity of the model can be altered by varying the number of elements for each link. With a higher number of elements, the order of the resulting model will be higher. This requires specification of more points on the link and hence, more knowledge of the nature of the link flexure. Also, if additional links are included, their corresponding coordinate frames and transformation matrices must be defined.

4. Sakawa/Matsuno/Fukushima Model (1986)

In 1985, Sakawa, et. al. [12] developed a mathematical model of a flexible single-link robotic arm with the objective of accurately describing the transverse vibration and its underlying causes, so that a controller may be developed. The controller would act to damp out the vibration as quickly as possible. With these objectives, the model can be considered a structural model.

The model development is based on a single, flexible clamped-free link, which is modeled as a Bernoulli-Euler beam, shown below.

$$\ddot{w}(x,t) + 2\delta\alpha\dot{w}''''(x,t) + \alpha w''''(x,t) = -x\dot{\omega}(t)$$

with the boundary conditions:

$$w(0,t) = w'(0,t) = w''(\ell,t) = 0$$

$$w'''(\ell,t) + (m/\rho)w''''(\ell,t) = 0$$

where α is the bending stiffness, δ is a damping constant, and w is the transverse displacement of the link. The dynamic equation for the joint actuator motor is given by:

$$J_m N \dot{\omega}(t) + \mu N \omega(t) = \tau(t) + \frac{EI}{N} w''(0,t)$$

where J_m is the motor inertia, μ is the viscous friction coefficient, and τ is the torque output of the motor. With these dynamics equations, the associated eigenvalue problem can be formulated.

The beam dynamics equation, along with its associated boundary conditions, is used to formulate a frequency equation:

$$1 + \cosh\beta \cos\beta + \frac{m}{\rho\ell}(\sinh\beta \cos\beta - \cosh\beta \sin\beta) = 0$$

The solutions of the frequency equation are designated β_i and yield the system eigenvalues which follow.

$$\lambda_i = \alpha \left(\frac{\beta_i}{\ell}\right)^4 \qquad i = 1, 2, \ldots$$

The associated eigenfunctions are expressed as:

$$\phi(x) = \frac{1}{C_i}\left[\cosh\frac{\beta_i x}{\ell} - \cos\frac{\beta_i x}{\ell} - \gamma_i\left(\sinh\frac{\beta_i x}{\ell} - \sin\frac{\beta_i x}{\ell}\right)\right]$$

where $\gamma_i = (\cosh\beta_i + \cos\beta_i)/(\sinh\beta_i + \sin\beta_i)$

The eigenfunctions describe the mode shapes corresponding to specific eigenvalues.

The general expression for the transverse displacement as a function of position along the longitudinal axis of the undeformed arm is given by:

$$w(x) = C_1 \cos\frac{\beta x}{\ell} + C_2 \cosh\frac{\beta x}{\ell} + C_3 \sin\frac{\beta x}{\ell} + C_4 \sinh\frac{\beta x}{\ell}$$

where the C's are arbitrary constants.

An experimental arm was fabricated, and a controller was constructed based upon the dynamics exhibited in the model. This controller was a dynamic compensator and was implemented by a microcomputer. Motor angle of rotation and transverse vibration were controlled by the input voltage to the armature of the actuator motor. Fed back to the microcomputer were angular velocity, as measured by a tachometer, and strain measured at the base of the arm.

5. Davis/Hirschorn Model (1988)

Developed by Davis and Hirschorn in 1988 [3], this model is a structural model of a single flexible link of a robotic arm. The objective is to model the dynamics of the link as a distributed parameter system and to model the joint actuator and load as discrete systems so that a tracking controller can be developed.

The model is based on a Bernoulli-Euler beam mounted on the shaft of a servo motor. To insure that the model is exponentially stable, a frequency-dependent damping term is added to the basic beam equation to yield:

$$\rho \frac{\partial^2 w}{\partial t^2} = - EI \frac{\partial^4 w}{\partial x^4} + D \frac{\partial^3 w}{\partial x^2 \partial t}$$

where ρ is the mass per unit length, EI is the bending stiffness, and D is the damping coefficient. For boundary conditions, shear and bending moments S(x,t), M(x,t) and compressive stress, T(x,t) along the beam centerline are calculated. After some manipulation of these equations, an expression for the transverse motion of the vibrating beam is obtained. The servo motor is modeled as a discrete component, and includes the motor inertia and beam reaction torque produced by the bending.

Since the model was developed for the purpose of constructing a tracking controller, the motion of the vibrating beam is referenced to a nominal tracking path. Also, after the basic beam equations have been derived, in order to develop a robust controller, the authors assume the configuration of the system actually to be two coaxially mounted beams with a force actuator mounted at the tip of the inner beam. The dynamics of both beams are modeled as described earlier.

VI. SUMMARY AND CONCLUSION

An extensive survey of current literature in the area of mathematical modeling of robot arms with flexible links shows that there is a great diversity in many aspects of these models. Most current models can be classified as either kinematic or structural models. Kinematic models generally are developed for the study of arm dynamics during large changes in configuration, and are used to examine the effects of the vibratory modes on the equations of motion of the system. Structural models emphasize developing partial differential equations of the flexural dynamics which are used to solve the related eigenvalue problem and characterize the vibratory modes of each link of the arm.

The models in each class are very similar conceptually and, as a result, follow the same general pattern of development, outlined in this thesis. Several short FORTRAN programs were developed to demonstrate the basic patterns of development for kinematic and structural models. However, individual models in each class may differ significantly due to such factors as effects modeled and assumptions made. These serve as additional bases for further classification of the models.

Examination of current models indicates that many areas in the modeling of robotic arms with flexible links can be developed further. One important area is the development of hybrid models which simulate both the kinematic and structural characteristics of flexible robotic arms. With these, the development of real-time controllers to control large changes in arm configuration could be made easier. A hybrid model also could be used to develop structural models of multi-link arms, and examine the coupling effects between several links and actuator motors. Finally, many nonlinear effects such as Coriolis forces, backlash, and joint spring effects could be added to existing models to improve accuracy.

The ultimate goal of developing effective control techniques for robot arms with flexible links is the reason for studying the modeling techniques of those arms. Many potential benefits are to be gained by improving the techniques by which mathematical models of flexible robotic arms are developed.

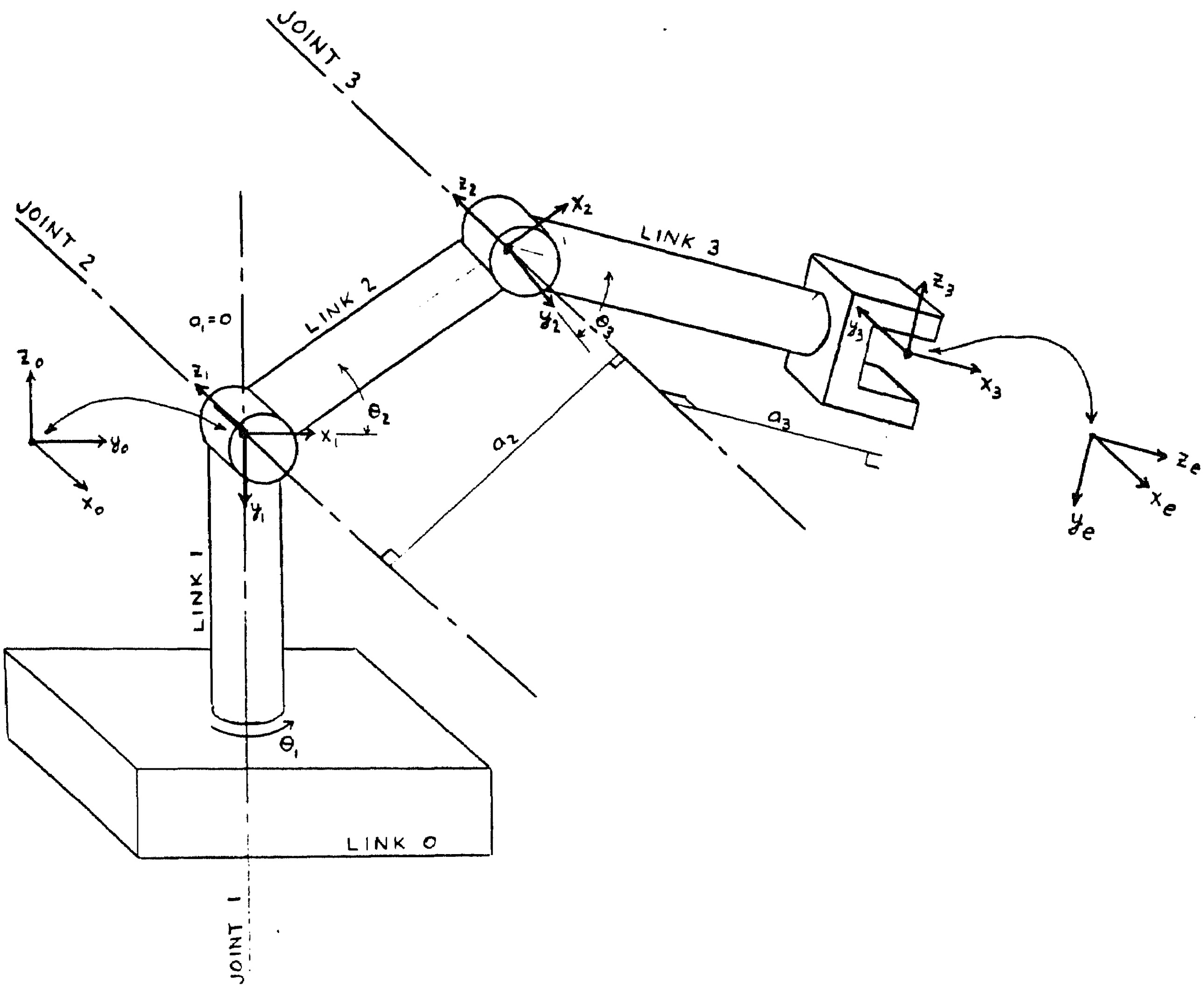
APPENDIX A: Kinematic Model Development Computer Simulations

The computer simulations described in this section were developed to demonstrate certain phases of the model development process for kinematic models of robot arms with flexible links. The example three degree of freedom serial manipulator shown in Figure A - 1 was used in developing the simulations. The link parameter table [11] describes the configuration of each link. α denotes the twist angle between the two joints of a specific link. a represents the length of the link normal to the two joint axes, and d represents the distance between the two normals for a particular joint. θ depicts the measured joint angle, which is controlled by the actuator motor.

The first program illustrates how the Denavit-Hartenberg transformation matrices are devloped for a specified configuration of the manipulator, and assumes no flexible deflections. The second program calculates the transformation matrices for the same configuration of the manipulator, but also uses the F matrices associated with each link to take the flexible deflections into account. Note that the matrix T_e denotes the position and orientation of the end effector coordinate frame with respect to the base coordinate frame.

The third kinematic model program was based on the Usoro/Nadira/Mahil model [15], and was developed to calculate the time-varying position coordinates of a vibrating link. It divides a link into a finite number of distributed elements and, assuming a specified modal shape, uses Hermitian polynomials to calculate position, velocity, and acceleration of an arbitrary element of the link.

LINK COORDINATE FRAME ASSIGNMENT FOR A 3 DEGREE OF FREEDOM SERIAL MANIPULATOR



LINK PARAMETER TABLE

LINK	α	a	d	θ
1	-90	0	0	$\theta_1 + 90$
2	0	a_2	0	$-\theta_2$
3	90	a_3	0	$-\theta_3 + 90$
e	-90	0	0	-90

FIGURE A-1

```
C
C                                                   JOHN G. DANIS
C                                                   MASTER'S THESIS
C                                                   ADVISOR: DR. F. CHICHESTER
C
C     KINEMATIC MODEL: GENERATION OF TRANSFORMATION MATRICES
C
C     THIS PROGRAM GENERATES THE DENAVIT-HARTENBERG MATRICES
C     USED TO PERFORM A HOMOGENEOUS TRANSFORMATION OF
C     COORDINATES BETWEEN SUCCESSIVE LINKS OF A SERIAL MANIPULATOR
C
      DIMENSION AN1(4,4),AN2(4,4),AN3(4,4),E(4,4),T2(4,4),T3(4,4),
     &          TE(4,4)
C
      REAL JTANG1,JTANG2,JTANG3
C
C     LINK PARAMETERS FOR LINK 1:
      ALPHA1 = -90.0
      A1 = 0.0
      D1 = 0.0
      JTANG1 = 0.0
      THETA1 = 90.0 + JTANG1
      CALL ANCOMP(AN1,ALPHA1,A1,D1,THETA1)
C
C     LINK PARAMETERS FOR LINK 2:
      ALPHA2 = 0.0
      A2 = 10.0
      D2 = 0.0
      JTANG2 = 30.0
      THETA2 = -JTANG2
      CALL ANCOMP(AN2,ALPHA2,A2,D2,THETA2)
C
C     LINK PARAMETERS FOR LINK 3:
      ALPHA3 = 90.0
      A3 = 12.0
      D3 = 0.0
      JTANG3 = 60.0
      THETA3 = 90.0 - JTANG3
      CALL ANCOMP(AN3,ALPHA3,A3,D3,THETA3)
C
C     LINK PARAMETERS FOR THE END EFFECTOR:
      ALPHAE = -90.0
      AE = 0.0
      DE = 0.0
      THETAE = -90.0
      CALL ANCOMP(E,ALPHAE,AE,DE,THETAE)
C
C     [TE] = [AN1]*[AN2]*[AN3]*[E]
C
      NUM = 2
      CALL MATMUL(AN1,AN2,T2,NUM)
      NUM = 3
      CALL MATMUL(T2,AN3,T3,NUM)
      NUM = 4
      CALL MATMUL(T3,E,TE,NUM)
C
      STOP
      END
C
C
      SUBROUTINE ANCOMP(AN,ALPHA,A,D,THETA)
      DIMENSION AN(4,4)
      CONVF = 1.74533E-2
      ALPHA = ALPHA*CONVF
      THETA = THETA*CONVF
```

```
      AN(1,1) = COS(THETA)
      AN(1,2) = -SIN(THETA)*COS(ALPHA)
      AN(1,3) = SIN(THETA)*SIN(ALPHA)
      AN(1,4) = A*COS(THETA)
      AN(2,1) = SIN(THETA)
      AN(2,2) = COS(THETA)*COS(ALPHA)
      AN(2,3) = -COS(THETA)*SIN(ALPHA)
      AN(2,4) = A*SIN(THETA)
      AN(3,1) = 0.0
      AN(3,2) = SIN(ALPHA)
      AN(3,3) = COS(ALPHA)
      AN(3,4) = D
      AN(4,1) = 0.0
      AN(4,2) = 0.0
      AN(4,3) = 0.0
      AN(4,4) = 1.0
      RETURN
      END
C
C
      SUBROUTINE MATMUL(A,B,PR,NUM)
      DIMENSION A(4,4),B(4,4),PR(4,4)
      PR(1,1)=A(1,1)*B(1,1)+A(1,2)*B(2,1)+A(1,3)*B(3,1)+A(1,4)*B(4,1)
      PR(1,2)=A(1,1)*B(1,2)+A(1,2)*B(2,2)+A(1,3)*B(3,2)+A(1,4)*B(4,2)
      PR(1,3)=A(1,1)*B(1,3)+A(1,2)*B(2,3)+A(1,3)*B(3,3)+A(1,4)*B(4,3)
      PR(1,4)=A(1,1)*B(1,4)+A(1,2)*B(2,4)+A(1,3)*B(3,4)+A(1,4)*B(4,4)
      PR(2,1)=A(2,1)*B(1,1)+A(2,2)*B(2,1)+A(2,3)*B(3,1)+A(2,4)*B(4,1)
      PR(2,2)=A(2,1)*B(1,2)+A(2,2)*B(2,2)+A(2,3)*B(3,2)+A(2,4)*B(4,2)
      PR(2,3)=A(2,1)*B(1,3)+A(2,2)*B(2,3)+A(2,3)*B(3,3)+A(2,4)*B(4,3)
      PR(2,4)=A(2,1)*B(1,4)+A(2,2)*B(2,4)+A(2,3)*B(3,4)+A(2,4)*B(4,4)
      PR(3,1)=A(3,1)*B(1,1)+A(3,2)*B(2,1)+A(3,3)*B(3,1)+A(3,4)*B(4,1)
      PR(3,2)=A(3,1)*B(1,2)+A(3,2)*B(2,2)+A(3,3)*B(3,2)+A(3,4)*B(4,2)
      PR(3,3)=A(3,1)*B(1,3)+A(3,2)*B(2,3)+A(3,3)*B(3,3)+A(3,4)*B(4,3)
      PR(3,4)=A(3,1)*B(1,4)+A(3,2)*B(2,4)+A(3,3)*B(3,4)+A(3,4)*B(4,4)
      PR(4,1)=A(4,1)*B(1,1)+A(4,2)*B(2,1)+A(4,3)*B(3,1)+A(4,4)*B(4,1)
      PR(4,2)=A(4,1)*B(1,2)+A(4,2)*B(2,2)+A(4,3)*B(3,2)+A(4,4)*B(4,2)
      PR(4,3)=A(4,1)*B(1,3)+A(4,2)*B(2,3)+A(4,3)*B(3,3)+A(4,4)*B(4,3)
      PR(4,4)=A(4,1)*B(1,4)+A(4,2)*B(2,4)+A(4,3)*B(3,4)+A(4,4)*B(4,4)
C
      WRITE(*,5)
    5 FORMAT('0')
      DO 40 I = 1,4
         DO 30 J = 1,4
            IF (NUM.EQ.4) GO TO 20
            CONTINUE
            WRITE(*,10) NUM,I,J,PR(I,J)
   10       FORMAT(' ','T',I1,'(',I1,',',I1,') = ',F10.6)
            GO TO 30
   20       WRITE(*,25) I,J,PR(I,J)
   25       FORMAT(' ','TE(',I1,',',I1,') = ',F10.6)
   30    CONTINUE
   40 CONTINUE
      RETURN
      END
```

```
T2(1,1) =   -.000001
T2(1,2) =    .000000
T2(1,3) =  -1.000000
T2(1,4) =   -.000009
T2(2,1) =    .866025
T2(2,2) =    .500000
T2(2,3) =   -.000001
T2(2,4) =   8.660254
T2(3,1) =    .500000
T2(3,2) =   -.866025
T2(3,3) =   -.000001
T2(3,4) =   5.000002
T2(4,1) =    .000000
T2(4,2) =    .000000
T2(4,3) =    .000000
T2(4,4) =   1.000000


T3(1,1) =   -.000001
T3(1,2) =  -1.000000
T3(1,3) =    .000000
T3(1,4) =   -.000016
T3(2,1) =   1.000000
T3(2,2) =   -.000001
T3(2,3) =    .000000
T3(2,4) =  20.660250
T3(3,1) =    .000000
T3(3,2) =    .000000
T3(3,3) =   1.000000
T3(3,4) =   5.000002
T3(4,1) =    .000000
T3(4,2) =    .000000
T3(4,3) =    .000000
T3(4,4) =   1.000000


TE(1,1) =   1.000000
TE(1,2) =    .000000
TE(1,3) =    .000000
TE(1,4) =   -.000016
TE(2,1) =    .000000
TE(2,2) =   -.000001
TE(2,3) =   1.000000
TE(2,4) =  20.660250
TE(3,1) =    .000000
TE(3,2) =  -1.000000
TE(3,3) =   -.000001
TE(3,4) =   5.000002
TE(4,1) =    .000000
TE(4,2) =    .000000
TE(4,3) =    .000000
TE(4,4) =   1.000000
Stop - Program terminated.
```

```
C
C                                                   JOHN G. DANIS
C                                                   MASTER'S THESIS
C                                                   ADVISOR: DR. F. CHICHESTER
C
C     KINEMATIC MODEL: GENERATION OF TRANSFORMATION MATRICES
C
C     THIS PROGRAM GENERATES THE DENAVIT-HARTENBERG MATRICES
C     USED TO PERFORM A HOMOGENEOUS TRANSFORMATION OF
C     COORDINATES BETWEEN SUCCESSIVE LINKS OF A SERIAL MANIPULATOR
C
C     THIS VERSION OF THE MODEL ACCOUNTS FOR LINK FLEXIBILITY BY
C     USING F MATRICES IN THE COORDINATE TRANSFORMATIONS
C
      DIMENSION AN1(4,4),AN2(4,4),AN3(4,4),E(4,4),T2(4,4),T3(4,4),
     &          TE(4,4),FN1(4,4),FN2(4,4),FN3(4,4),FNE(4,4),
     &          A1F1(4,4),A2F2(4,4),A3F3(4,4),EFE(4,4)
C
      REAL JTANG1,JTANG2,JTANG3
C
C     LINK PARAMETERS FOR LINK 1:
      ALPHA1 = -90.0
      A1 = 0.0
      D1 = 0.0
      JTANG1 = 0.0
      THETA1 = 90.0 + JTANG1
      CALL ANCOMP(AN1,ALPHA1,A1,D1,THETA1)
C
C     FLEXIBILITY PARAMETERS FOR LINK 1:
      THX1 = 1.0
      THY1 = -1.0
      THZ1 = 0.5
      DELX1 = 0.5
      DELY1 = 0.5
      DELZ1 = -0.25
      CALL FNCOMP(FN1,THX1,THY1,THZ1,DELX1,DELY1,DELZ1)
C
C     LINK PARAMETERS FOR LINK 2:
      ALPHA2 = 0.0
      A2 = 10.0
      D2 = 0.0
      JTANG2 = 30.0
      THETA2 = -JTANG2
      CALL ANCOMP(AN2,ALPHA2,A2,D2,THETA2)
C
C     FLEXIBILITY PARAMETERS FOR LINK 2:
      THX2 = -2.0
      THY2 = 1.0
      THZ2 = -0.5
      DELX2 = 0.25
      DELY2 = 0.25
      DELZ2 = -0.5
      CALL FNCOMP(FN2,THX2,THY2,THZ2,DELX2,DELY2,DELZ2)
C
C     LINK PARAMETERS FOR LINK 3:
      ALPHA3 = 90.0
      A3 = 12.0
      D3 = 0.0
      JTANG3 = 60.0
      THETA3 = 90.0 - JTANG3
      CALL ANCOMP(AN3,ALPHA3,A3,D3,THETA3)
C
C     FLEXIBILITY PARAMETERS FOR LINK 3:
      THX3 = -1.0
      THY3 = 1.0
```

```
      THZ3 = 0.5
      DELX3 = 0.5
      DELY3 = 0.25
      DELZ3 = -0.25
      CALL FNCOMP(FN3,THX3,THY3,THZ3,DELX3,DELY3,DELZ3)
C
C     LINK PARAMETERS FOR THE END EFFECTOR:
      ALPHAE = -90.0
      AE = 0.0
      DE = 0.0
      THETAE = -90.0
      CALL ANCOMP(E,ALPHAE,AE,DE,THETAE)
C
C     FLEXIBILITY PARAMETERS OF THE END EFFECTOR:
      THXE = 0.1
      THYE = 0.05
      THZE = -0.05
      DELXE = 0.01
      DELYE = -0.01
      DELZE = 0.0
      CALL FNCOMP(FNE,THXE,THYE,THZE,DELXE,DELYE,DELZE)
C
      NUM = 1
      CALL MATMUL(AN1,FN1,A1F1,NUM)
      CALL MATMUL(AN2,FN2,A2F2,NUM)
      CALL MATMUL(AN3,FN3,A3F3,NUM)
      CALL MATMUL(E,FNE,EFE,NUM)
C
C     [TE] = [A1F1]*[A2F2]*[A3F3]*[EFE]
C
      NUM = 2
      CALL MATMUL(A1F1,A2F2,T2,NUM)
      NUM = 3
      CALL MATMUL(T2,A3F3,T3,NUM)
      NUM = 4
      CALL MATMUL(T3,EFE,TE,NUM)
C
      STOP
      END
C
C
      SUBROUTINE ANCOMP(AN,ALPHA,A,D,THETA)
      DIMENSION AN(4,4)
      CONVF = 1.74533E-2
      ALPHA = ALPHA*CONVF
      THETA = THETA*CONVF
      AN(1,1) = COS(THETA)
      AN(1,2) = -SIN(THETA)*COS(ALPHA)
      AN(1,3) = SIN(THETA)*SIN(ALPHA)
      AN(1,4) = A*COS(THETA)
      AN(2,1) = SIN(THETA)
      AN(2,2) = COS(THETA)*COS(ALPHA)
      AN(2,3) = -COS(THETA)*SIN(ALPHA)
      AN(2,4) = A*SIN(THETA)
      AN(3,1) = 0.0
      AN(3,2) = SIN(ALPHA)
      AN(3,3) = COS(ALPHA)
      AN(3,4) = D
      AN(4,1) = 0.0
      AN(4,2) = 0.0
      AN(4,3) = 0.0
      AN(4,4) = 1.0
      RETURN
      END
C
C
```

```
      SUBROUTINE FNCOMP(FN,THX,THY,THZ,DELX,DELY,DELZ)
      DIMENSION FN(4,4)
      CONVF = 1.74533E-2
      THX = THX*CONVF
      THY = THY*CONVF
      THZ = THZ*CONVF
      FN(1,1) = 1.0
      FN(1,2) = -THZ
      FN(1,3) = THY
      FN(1,4) = DELX
      FN(2,1) = THZ
      FN(2,2) = 1.0
      FN(2,3) = -THX
      FN(2,4) = DELY
      FN(3,1) = -THY
      FN(3,2) = THX
      FN(3,3) = 1.0
      FN(3,4) = DELZ
      FN(4,1) = 0.0
      FN(4,2) = 0.0
      FN(4,3) = 0.0
      FN(4,4) = 1.0
      RETURN
      END
C
C
      SUBROUTINE MATMUL(A,B,PR,NUM)
      DIMENSION A(4,4),B(4,4),PR(4,4)
      PR(1,1)=A(1,1)*B(1,1)+A(1,2)*B(2,1)+A(1,3)*B(3,1)+A(1,4)*B(4,1)
      PR(1,2)=A(1,1)*B(1,2)+A(1,2)*B(2,2)+A(1,3)*B(3,2)+A(1,4)*B(4,2)
      PR(1,3)=A(1,1)*B(1,3)+A(1,2)*B(2,3)+A(1,3)*B(3,3)+A(1,4)*B(4,3)
      PR(1,4)=A(1,1)*B(1,4)+A(1,2)*B(2,4)+A(1,3)*B(3,4)+A(1,4)*B(4,4)
      PR(2,1)=A(2,1)*B(1,1)+A(2,2)*B(2,1)+A(2,3)*B(3,1)+A(2,4)*B(4,1)
      PR(2,2)=A(2,1)*B(1,2)+A(2,2)*B(2,2)+A(2,3)*B(3,2)+A(2,4)*B(4,2)
      PR(2,3)=A(2,1)*B(1,3)+A(2,2)*B(2,3)+A(2,3)*B(3,3)+A(2,4)*B(4,3)
      PR(2,4)=A(2,1)*B(1,4)+A(2,2)*B(2,4)+A(2,3)*B(3,4)+A(2,4)*B(4,4)
      PR(3,1)=A(3,1)*B(1,1)+A(3,2)*B(2,1)+A(3,3)*B(3,1)+A(3,4)*B(4,1)
      PR(3,2)=A(3,1)*B(1,2)+A(3,2)*B(2,2)+A(3,3)*B(3,2)+A(3,4)*B(4,2)
      PR(3,3)=A(3,1)*B(1,3)+A(3,2)*B(2,3)+A(3,3)*B(3,3)+A(3,4)*B(4,3)
      PR(3,4)=A(3,1)*B(1,4)+A(3,2)*B(2,4)+A(3,3)*B(3,4)+A(3,4)*B(4,4)
      PR(4,1)=A(4,1)*B(1,1)+A(4,2)*B(2,1)+A(4,3)*B(3,1)+A(4,4)*B(4,1)
      PR(4,2)=A(4,1)*B(1,2)+A(4,2)*B(2,2)+A(4,3)*B(3,2)+A(4,4)*B(4,2)
      PR(4,3)=A(4,1)*B(1,3)+A(4,2)*B(2,3)+A(4,3)*B(3,3)+A(4,4)*B(4,3)
      PR(4,4)=A(4,1)*B(1,4)+A(4,2)*B(2,4)+A(4,3)*B(3,4)+A(4,4)*B(4,4)
C
      IF (NUM.EQ.1) GO TO 50
      CONTINUE
      WRITE(*,5)
    5 FORMAT('0')
      DO 40 I = 1,4
         DO 30 J = 1,4
            IF (NUM.EQ.4) GO TO 20
            CONTINUE
            WRITE(*,10) NUM,I,J,PR(I,J)
   10       FORMAT(' ','T',I1,'(',I1,',',I1,') = ',F10.6)
            GO TO 30
   20       WRITE(*,25) I,J,PR(I,J)
   25       FORMAT(' ','TE(',I1,',',I1,') = ',F10.6)
   30    CONTINUE
   40 CONTINUE
   50 CONTINUE
      RETURN
      END
```

```
T2(1,1) =    .011272
T2(1,2) =    .011009
T2(1,3) =  -1.000944
T2(1,4) =    .678550
T2(2,1) =    .866396
T2(2,2) =    .500648
T2(2,3) =    .014927
T2(2,4) =   9.553322
T2(3,1) =    .499734
T2(3,2) =   -.866701
T2(3,3) =   -.004335
T2(3,4) =   4.321214
T2(4,1) =    .000000
T2(4,2) =    .000000
T2(4,3) =    .000000
T2(4,4) =   1.000000


T3(1,1) =    .006600
T3(1,2) =  -1.001009
T3(1,3) =   -.021101
T3(1,4) =    .620122
T3(2,1) =   1.000782
T3(2,2) =    .006201
T3(2,3) =    .017350
T3(2,4) =  22.065210
T3(3,1) =   -.018067
T3(3,2) =   -.021790
T3(3,3) =   1.000366
T3(3,4) =   4.062912
T3(4,1) =    .000000
T3(4,2) =    .000000
T3(4,3) =    .000000
T3(4,4) =   1.000000


TE(1,1) =   1.000985
TE(1,2) =    .021986
TE(1,3) =    .007437
TE(1,4) =    .629921
TE(2,1) =   -.007060
TE(2,2) =   -.015609
TE(2,3) =   1.000806
TE(2,4) =  22.065320
TE(3,1) =    .022679
TE(3,2) =  -1.000378
TE(3,3) =   -.016303
TE(3,4) =   4.073133
TE(4,1) =    .000000
TE(4,2) =    .000000
TE(4,3) =    .000000
TE(4,4) =   1.000000
Stop - Program terminated.
```

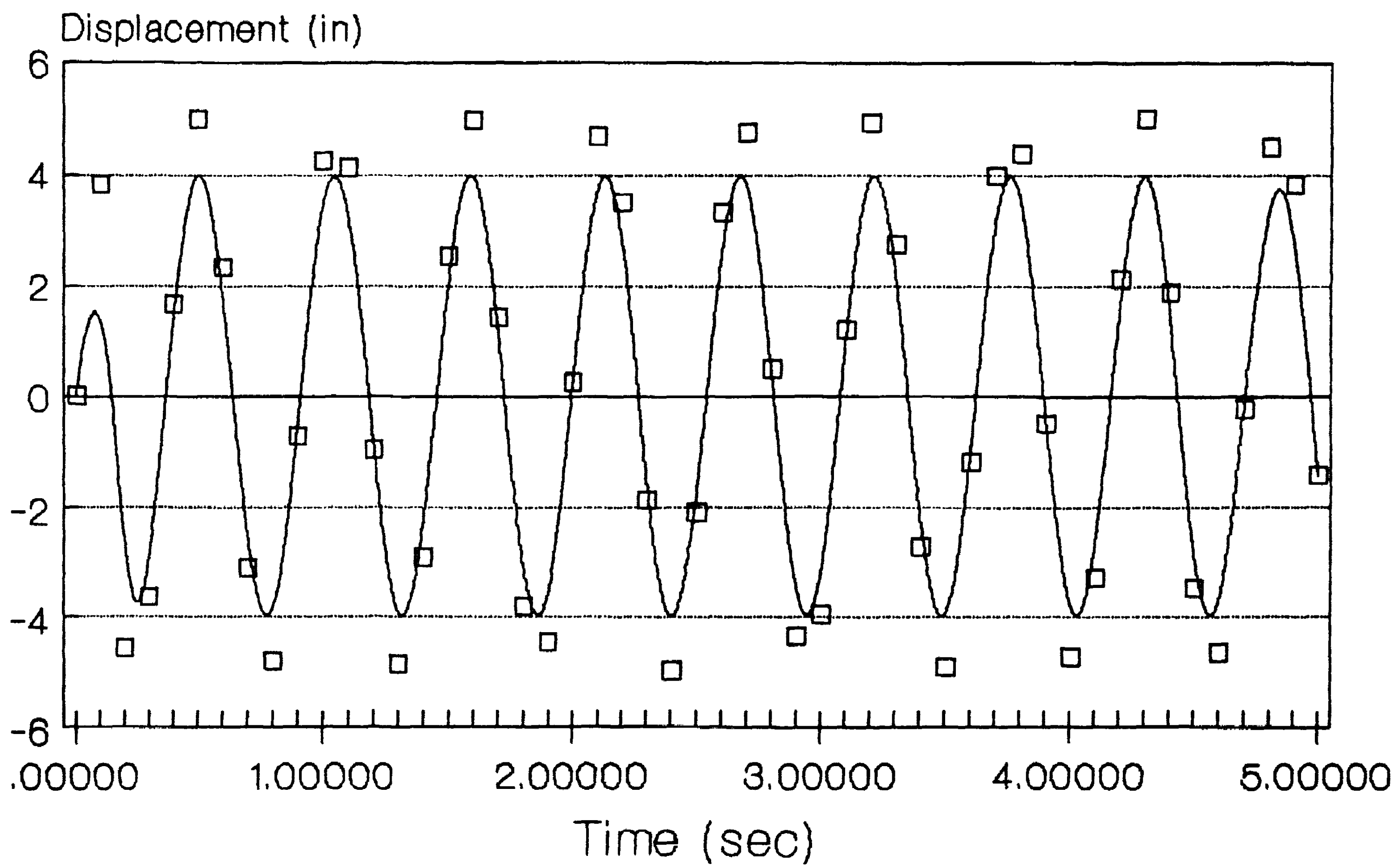
```
C
C                              JOHN G. DANIS
C                              MASTER'S THESIS
C                              ADVISOR: DR. FREDERICK CHICHESTER
C
C
C     THIS PROGRAM CALCULATES THE TIME-VARYING POSITION
C     COORDINATES OF AN ELEMENT OF A VIBRATING LINK
C
      DIMENSION YELM(100), VEL(100)
      WRITE(*,010)
 010  FORMAT(1X,'ROBOTIC ARM POSITION COORDINATES PROGRAM')
C     RLNK IS THE OVERALL LENGTH OF THE LINK
      RLNK=48.0
C     N IS THE NUMBER OF ELEMENTS IN THE LINK
      N=10
C     SF IS THE SCALE FACTOR DESCRIBING THE SHAPE OF THE BEND
      SF=0.02
C     W IS THE MODE FREQUENCY
      W=11.6
C     DELTT IS THE DELTA TIME INTERVAL
      DELTT=0.1
      NOBSV=50
      NDE=10
      R=0.5
      WRITE(*,68)
 068  FORMAT('0','    TIME          X           Y          VEL         ACC')
C     RELM IS THE LENGTH OF EACH ELEMENT
      RELM=RLNK/FLOAT(N)
C     X IS THE POSITION ALONG THE ELEMENT
      X=R*RELM
      L=NOBSV+1
      DO 100 I=1,L
      THETA=W*FLOAT(I-1)*DELTT
      U1=SIN(THETA)*SF*(FLOAT(NDE-1)*RELM)**2
      U2=2.0*SIN(THETA)*SF*FLOAT(NDE-1)*RELM
      U3=SIN(THETA)*SF*(FLOAT(NDE)*RELM)**2
      U4=2.0*SIN(THETA)*SF*FLOAT(NDE)*RELM
      PHI1=1-3*X**2/RELM**2+2*X**3/RELM**3
      PHI2=X-2*X**2/RELM+X**3/RELM**2
      PHI3=3*X**2/RELM**2-2*X**3/RELM**3
      PHI4=-X**2/RELM+X**3/RELM**2
      TIME=FLOAT(I-1)*DELTT
      XELM=FLOAT(NDE-1)*RELM+X
      YELM(I)=PHI1*U1+PHI2*U2+PHI3*U3+PHI4*U4
      VEL(I)=(YELM(I)-YELM(I-1))/DELTT
      ACC=(VEL(I)-VEL(I-1))/DELTT
      WRITE(*,070) TIME,XELM,YELM(I),VEL(I),ACC
 070  FORMAT(' ',F10.5,' ',F10.3,F10.3,' ',F10.3,' ',F10.3)
 100  CONTINUE
      STOP
      END
```

ROBOTIC ARM POSITION COORDINATES PROGRAM

TIME	X	Y	VEL	ACC
.00000	45.600	.000	.000	.000
.10000	45.600	38.127	381.273	3812.728
.20000	45.600	30.451	-76.758	-4580.310
.30000	45.600	-13.806	-442.578	-3658.197
.40000	45.600	-41.478	-276.719	1658.585
.50000	45.600	-19.321	221.568	4982.873
.60000	45.600	26.047	453.681	2321.133
.70000	45.600	40.124	140.777	-3129.037
.80000	45.600	6.000	-341.245	-4820.225
.90000	45.600	-35.332	-413.323	-720.777
1.00000	45.600	-34.219	11.133	4244.555
1.10000	45.600	8.002	422.214	4110.814
1.20000	45.600	40.610	326.081	-961.331
1.30000	45.600	24.432	-161.780	-4878.615
1.40000	45.600	-21.097	-455.292	-2935.113
1.50000	45.600	-41.282	-201.851	2534.407
1.60000	45.600	-11.874	294.077	4959.282
1.70000	45.600	31.798	436.724	1426.472
1.80000	45.600	37.271	54.726	-3819.988
1.90000	45.600	-2.031	-393.016	-4477.417
2.00000	45.600	-38.893	-368.619	243.967
2.10000	45.600	-29.032	98.607	4672.269
2.20000	45.600	15.705	447.375	3487.677
2.30000	45.600	41.576	258.702	-1886.732
2.40000	45.600	17.500	-240.756	-4994.580
2.50000	45.600	-27.599	-450.988	-2102.320
2.60000	45.600	-39.543	-119.439	3315.493
2.70000	45.600	-3.983	355.595	4750.336
2.80000	45.600	36.361	403.445	478.506
2.90000	45.600	33.024	-33.372	-4368.178
3.00000	45.600	-9.986	-430.099	-3967.260
3.10000	45.600	-41.000	-310.138	1199.602
3.20000	45.600	-22.760	182.397	4925.357
3.30000	45.600	22.822	455.815	2734.180
3.40000	45.600	40.987	181.653	-2741.625
3.50000	45.600	9.914	-310.733	-4923.858
3.60000	45.600	-33.069	-429.829	-1190.960
3.70000	45.600	-36.325	-32.563	3972.664
3.80000	45.600	4.057	403.822	4363.845
3.90000	45.600	39.565	355.087	-487.352
4.00000	45.600	27.543	-120.222	-4753.083
4.10000	45.600	-17.567	-451.105	-3308.836
4.20000	45.600	-41.574	-240.067	2110.383
4.30000	45.600	-15.637	259.369	4994.355
4.40000	45.600	29.085	447.219	1878.505
4.50000	45.600	38.867	97.816	-3494.032
4.60000	45.600	1.957	-369.097	-4669.132
4.70000	45.600	-37.304	-392.604	-235.070
4.80000	45.600	-31.750	55.532	4481.359
4.90000	45.600	11.945	436.956	3814.246
5.00000	45.600	41.291	293.456	-1435.000

Stop - Program terminated.

FIG. A-2: KINEMATIC MODEL
Transverse Displacement



Finite element model

APPENDIX B: Structural Model Development Computer Simulation

Computer simulations of certain phases of structural model development are described in this section. The simulations treat a single aluminum link 2 inches in diameter and 48 inches long. The eigenvalues associated with the first three modes of this clamped-free link are used to determine the corresponding eigenfunctions. Mode shapes are determined for the first three modes and are depicted in the accompanying graphs.

The last program simulates the flexible deflections of the link as the sum of the contributions from the first three modes. Since the eigenfunctions of all three modes are present in this model, the overall shape of the arm can be shown at any instant in time. The model can also be used to plot the motion of any point on the arm with respect to time. An accompanying graph plots the motion of the tip of the link with respect to time.

```
C                          JOHN G. DANIS
C                          MASTER'S THESIS
C                          ADVISOR: DR. F. D. CHICHESTER
C
C     STRUCTURAL MODEL: EIGENFUNCTION CALCULATION
C
C     THIS PROGRAM CALCULATES THE VALUES OF THE EIGENFUNCTION
C     ASSOCIATED WITH A PARTICULAR MODE ALONG THE LENGTH OF THE ARM
C
      DIMENSION PHI(60)
C
      REAL *8 LEN,INER,ARG,PHI,SF
C
      WRITE(*,3)
    3 FORMAT(' ','STRUCTURAL MODEL - EIGENFUNCTION CALCULATION')
C
C     DEFINE ARM CHARACTERISTICS:
C     ASSUME AN ALUMINUM LINK, 2 IN. DIA., 48 IN. LONG
      PIE=3.1415927
      LEN=48.0
C     MASS PER UNIT LENGTH:
      U=1.1409E-3
      E=1.0E7
      INER=4.12E-4
C
      EIGNVAL = 1.8751
      WN = (EIGNVAL**2/LEN**2)*SQRT(E*INER/U)
      ALPHA = (COSH(EIGNVAL) + COS(EIGNVAL))/(SINH(EIGNVAL) +
     &        SIN(EIGNVAL))
         WRITE(*,6) WN,ALPHA
    6    FORMAT('0','WN = ',F8.4,' RAD/SEC',7X,'ALPHA = ',F8.4)
C
C     NUMINT IS THE NUMBER OF POINTS ALONG THE ARM
      NUMINT=51
      DX=LEN/(FLOAT(NUMINT) - 1.0)
      DO 10 I = 1,NUMINT
         X = (FLOAT(I) - 1.0)*DX
         ARG = EIGNVAL*X/LEN
         PHI(I) = COSH(ARG) - COS(ARG) - ALPHA*(SINH(ARG) - SIN(ARG))
   10 CONTINUE
C
C     CALCULATION OF SCALE FACTOR TO NORMALIZE EIGENFUNCTION
      SF = ABS(1.0/PHI(NUMINT))
C
      DO 20 I = 1,NUMINT
         X = (FLOAT(I) - 1.0)*DX
         PHI(I) = PHI(I)*SF
         WRITE(*,15) X,PHI(I)
   15    FORMAT(' ',F8.4,7X,F8.4)
   20 CONTINUE
      STOP
      END
```

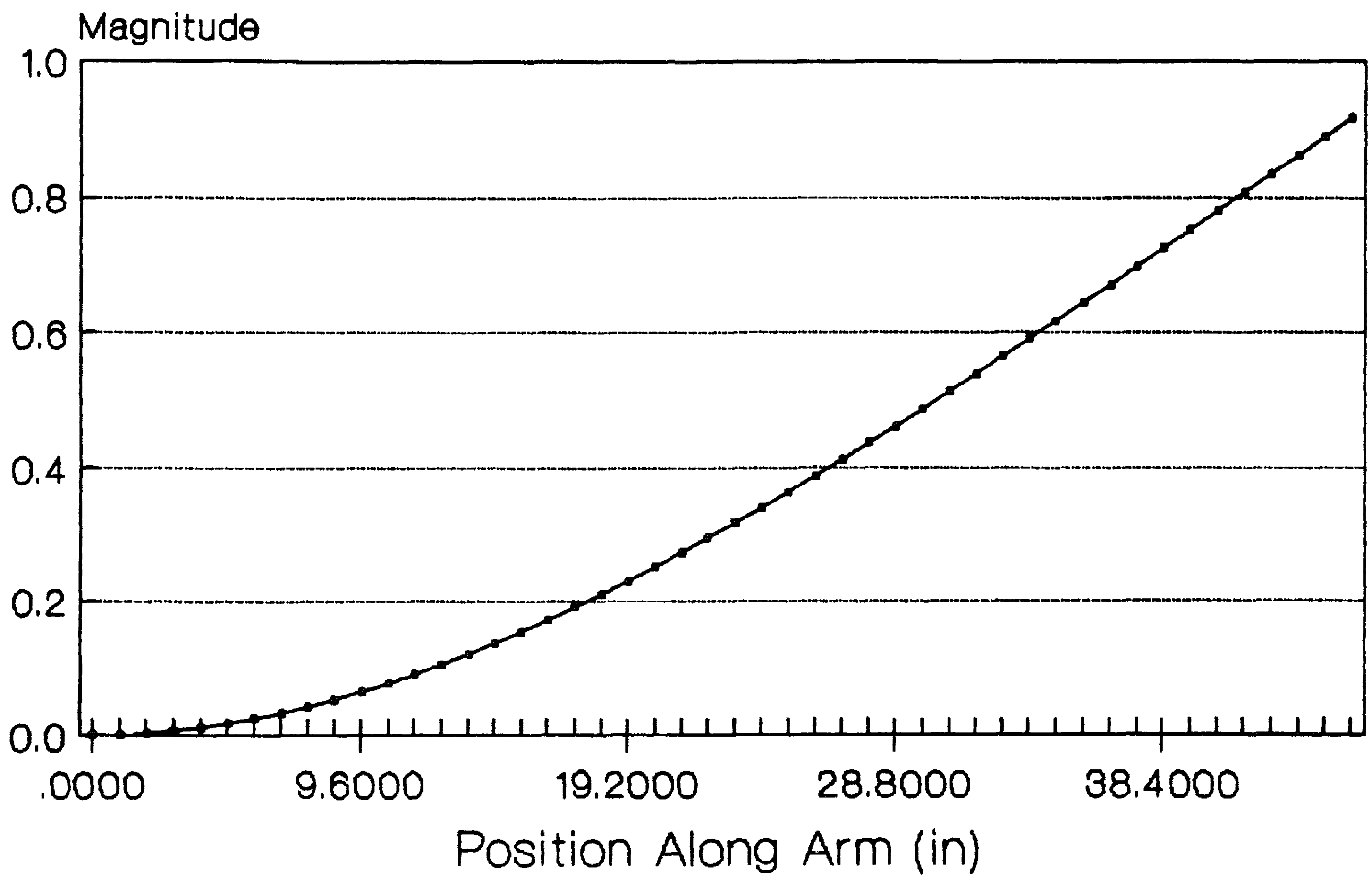
STRUCTURAL MODEL - EIGENFUNCTION CALCULATION

WN = 2.9000 RAD/SEC ALPHA = .7341

.0000	.0000
.9600	.0007
1.9200	.0028
2.8800	.0062
3.8400	.0108
4.8000	.0168
5.7600	.0239
6.7200	.0322
7.6800	.0417
8.6400	.0523
9.6000	.0639
10.5600	.0765
11.5200	.0901
12.4800	.1047
13.4400	.1201
14.4000	.1365
15.3600	.1536
16.3200	.1716
17.2800	.1903
18.2400	.2098
19.2000	.2299
20.1600	.2507
21.1200	.2720
22.0800	.2940
23.0400	.3165
24.0000	.3395
24.9600	.3630
25.9200	.3870
26.8800	.4113
27.8400	.4360
28.8000	.4611
29.7600	.4865
30.7200	.5123
31.6800	.5382
32.6400	.5644
33.6000	.5909
34.5600	.6175
35.5200	.6443
36.4800	.6712
37.4400	.6983
38.4000	.7255
39.3600	.7527
40.3200	.7801
41.2800	.8075
42.2400	.8349
43.2000	.8624
44.1600	.8899
45.1200	.9174
46.0800	.9449
47.0400	.9725
48.0000	1.0000

Stop - Program terminated.

FIG. B-1: STRUCTURAL MODEL
Eigenfunction Calculation



Principal Mode

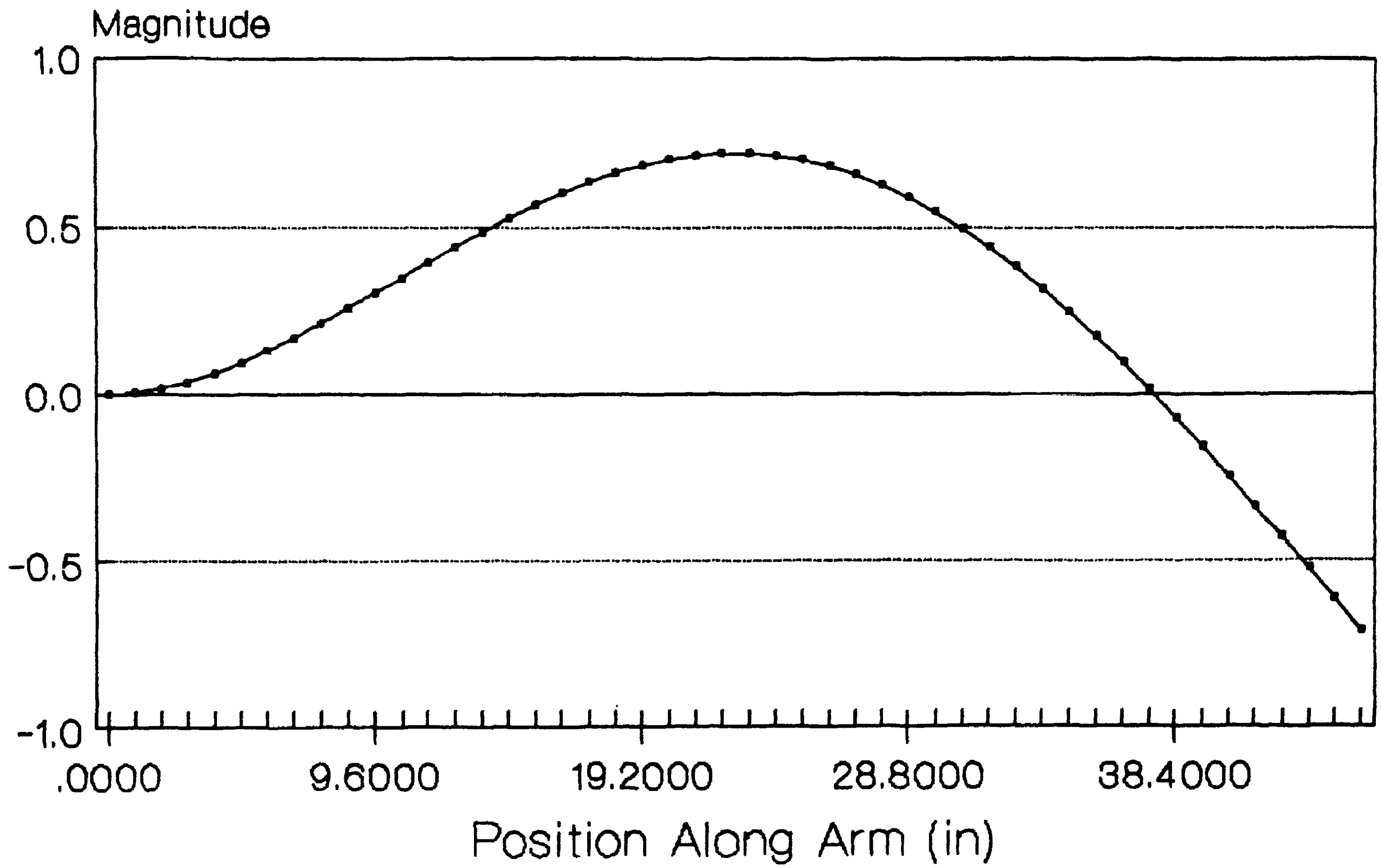
STRUCTURAL MODEL - EIGENFUNCTION CALCULATION

WN = 18.1731 RAD/SEC ALPHA = 1.0185

.0000	.0000
.9600	.0043
1.9200	.0165
2.8800	.0359
3.8400	.0615
4.8000	.0926
5.7600	.1284
6.7200	.1679
7.6800	.2104
8.6400	.2550
9.6000	.3011
10.5600	.3477
11.5200	.3943
12.4800	.4400
13.4400	.4842
14.4000	.5262
15.3600	.5654
16.3200	.6012
17.2800	.6332
18.2400	.6608
19.2000	.6835
20.1600	.7011
21.1200	.7132
22.0800	.7194
23.0400	.7197
24.0000	.7137
24.9600	.7015
25.9200	.6830
26.8800	.6581
27.8400	.6269
28.8000	.5896
29.7600	.5462
30.7200	.4970
31.6800	.4422
32.6400	.3822
33.6000	.3171
34.5600	.2474
35.5200	.1735
36.4800	.0957
37.4400	.0144
38.4000	-.0700
39.3600	-.1570
40.3200	-.2463
41.2800	-.3374
42.2400	-.4300
43.2000	-.5237
44.1600	-.6183
45.1200	-.7134
46.0800	-.8088
47.0400	-.9044
48.0000	-1.0000

Stop - Program terminated.

FIG. B-2: STRUCTURAL MODEL
Eigenfunction Calculation



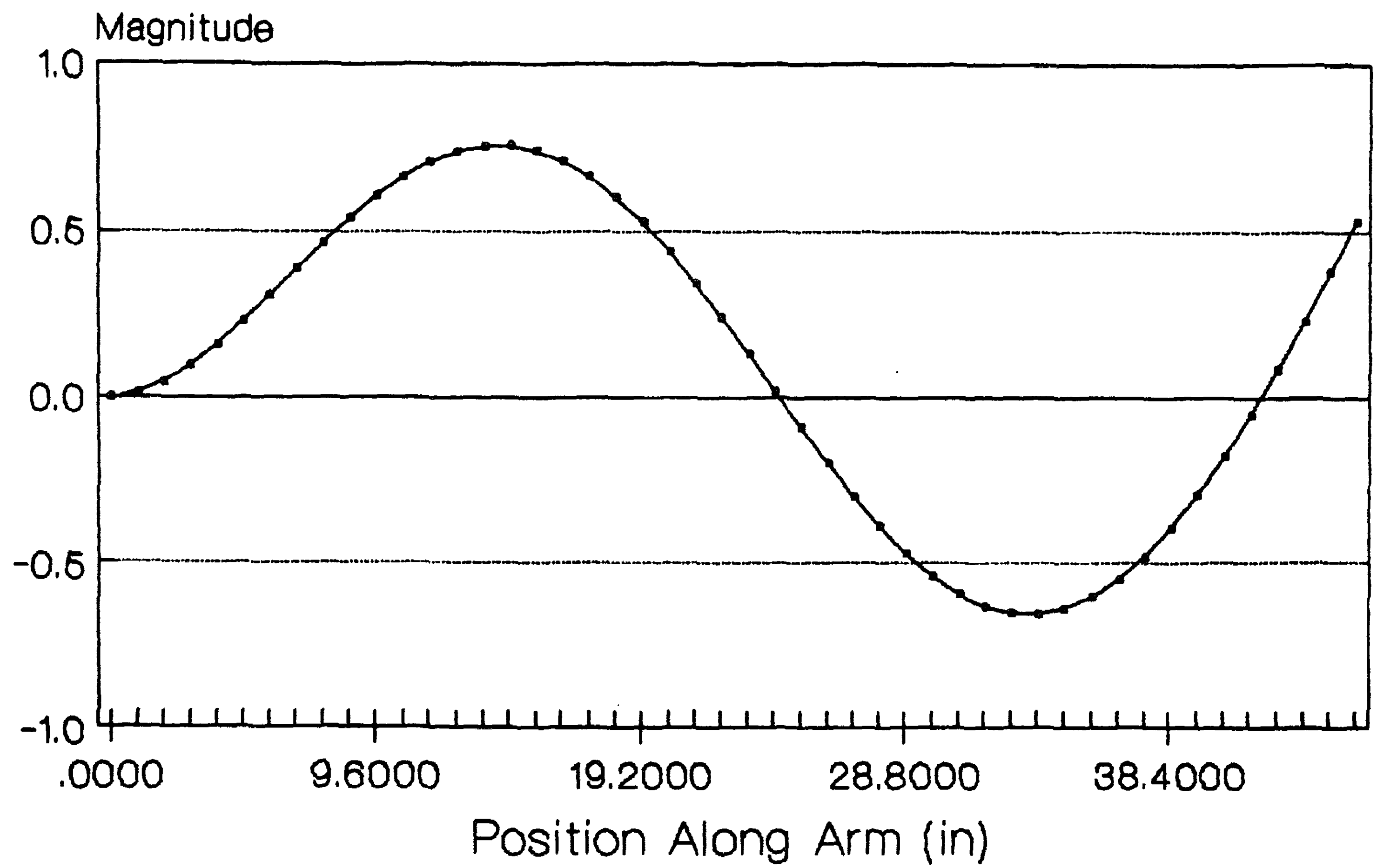
2nd Mode

STRUCTURAL MODEL – EIGENFUNCTION CALCULATION

WN = 50.8903 RAD/SEC		ALPHA = .9992
.0000	.0000	
.9600	.0117	
1.9200	.0442	
2.8800	.0936	
3.8400	.1562	
4.8000	.2280	
5.7600	.3055	
6.7200	.3852	
7.6800	.4635	
8.6400	.5376	
9.6000	.6044	
10.5600	.6615	
11.5200	.7067	
12.4800	.7384	
13.4400	.7551	
14.4000	.7560	
15.3600	.7408	
16.3200	.7095	
17.2800	.6625	
18.2400	.6008	
19.2000	.5257	
20.1600	.4390	
21.1200	.3427	
22.0800	.2390	
23.0400	.1304	
24.0000	.0196	
24.9600	-.0907	
25.9200	-.1978	
26.8800	-.2990	
27.8400	-.3918	
28.8000	-.4737	
29.7600	-.5427	
30.7200	-.5969	
31.6800	-.6347	
32.6400	-.6551	
33.6000	-.6573	
34.5600	-.6408	
35.5200	-.6057	
36.4800	-.5524	
37.4400	-.4817	
38.4000	-.3947	
39.3600	-.2928	
40.3200	-.1776	
41.2800	-.0510	
42.2400	.0850	
43.2000	.2287	
44.1600	.3779	
45.1200	.5310	
46.0800	.6865	
47.0400	.8431	
48.0000	1.0000	

Stop - Program terminated.

FIG. B-3: STRUCTURAL MODEL
Eigenfunction Calculation



3rd Mode

```
C                          JOHN G. DANIS
C                          MASTER'S THESIS
C                          ADVISOR: DR. F. D. CHICHESTER
C
C     STRUCTURAL MODEL: SIMPLE SIMULATION
C
C     THIS PROGRAM IS A SIMPLE SIMULATION OF A BASIC STRUCTURAL
C     MODEL OF A FLEXIBLE ROBOT ARM
C
      DIMENSION ARG(5,40),PHI(5,40),W(40,40),EIGNVAL(5),WN(5),
     &          ALPHA(5),AN(5),BN(5),MN(5),SF(5)
C
      REAL *8 LEN,INER,INTGD1,INTGD2,MN,ARG,PHI,DELT,W,INTGD3,SF
C
      WRITE(*,3)
    3 FORMAT(' ','STRUCTURAL MODEL - SIMPLE SIMULATION')
C
C     DEFINE ARM CHARACTERISTICS:
C     ASSUME AN ALUMINUM LINK, 2 IN. DIA., 48 IN. LONG
      PIE=3.1415927
      LEN=48.0
C     MASS PER UNIT LENGTH:
      U=1.1409E-3
      E=1.0E7
      INER=4.12E-4
C
C     DEFINE EIGENVALUES, MODE FREQUENCIES, AND CONSTANTS
C     INITIAL TRANSVERSE DISPLACEMENT AT TIP OF LINK:
      WO=1.0
C     INITIAL VELOCITY AT TIP OF LINK:
      WOD=0.0
C     N IS THE NUMBER OF INCLUDED MODES
      N=3
      EIGNVAL(1)=1.875
      EIGNVAL(2)=4.694
      EIGNVAL(3)=7.855
      DO 8 I=1,N
         WN(I)=(EIGNVAL(I)**2/LEN**2)*SQRT(E*INER/U)
         ALPHA(I)=(COSH(EIGNVAL(I))+COS(EIGNVAL(I)))/(SINH(EIGNVAL(I))
     &            +SIN(EIGNVAL(I)))
         WRITE(*,6) I,WN(I),I,ALPHA(I)
    6    FORMAT('0','WN(',I1,') = ',F8.4,' RAD/SEC',7X,'ALPHA(',I1,
     &          ') = ',F8.4)
    8 CONTINUE
C
C     NUMINT IS THE NUMBER OF POINTS FOR THE INTEGRATION
      NUMINT=21
      DX=LEN/(FLOAT(NUMINT) - 1.0)
      DO 15 I=1,N
         SUM1=0.0
         SUM2=0.0
         SUM3=0.0
         DO 10 J=1,NUMINT
            X1=(FLOAT(J) - 1.0)*DX
            ARG(I,J) = EIGNVAL(I)*X1/LEN
            PHI(I,J) = COSH(ARG(I,J)) - COS(ARG(I,J)) - ALPHA(I)*
     &                 (SINH(ARG(I,J)) - SIN(ARG(I,J)))
            INTGD1=(U*PHI(I,J)**2)*DX
            SUM1=SUM1+INTGD1
            INTGD2=WO*U*PHI(I,J)*DX
            SUM2=SUM2+INTGD2
            INTGD3=WOD*U*PHI(I,J)*DX
            SUM3=SUM3+INTGD3
   10    CONTINUE
         SF(I) = ABS(1.0/PHI(I.NUMINT))
```

```
         DO 12 J=1,NUMINT
            ARG(I,J) = SF(I)*ARG(I,J)
            PHI(I,J) = SF(I)*PHI(I,J)
   12    CONTINUE
C        GENERALIZED MASS:
         MN(I) = SF(I)*SUM1
         AN(I) = SF(I)*SUM2/MN(I)
         BN(I) = SF(I)*SUM3/(MN(I)*WN(I))
         WRITE(*,14) I,MN(I),I,AN(I),I,BN(I)
   14    FORMAT(' ','MN(',I1,') = ',F8.4,5X,'AN(',I1,') = ',F8.4,
     &          5X,'BN(',I1,') = ',F8.4)
   15 CONTINUE
C
      T=0.0
      DELT=0.10
      NTIME=21
      DO 80 I=1,NTIME
         DO 70 J=1,NUMINT
            W(I,J)=0.0
            DO 60 K=1,N
               W(I,J)=W(I,J)+PHI(K,J)*(AN(K)*COS(WN(K)*T)+BN(K)*
     &                  SIN(WN(K)*T))
   60       CONTINUE
   70    CONTINUE
         T=T+DELT
   80 CONTINUE
C
C     GENERATE DATA TO PLOT MOTION OF ARM TIP
      WRITE(*,90)
   90 FORMAT('0','MOTION OF ARM TIP WITH RESPECT TO TIME')
      DO 100 I=1,NTIME
         T = (FLOAT(I) - 1.0)*DELT
         WRITE(*,95) T,W(I,NUMINT)
   95    FORMAT(' ',F8.4,7X,F8.4)
  100 CONTINUE
C
C     GENERATE DATA TO PLOT ARM SHAPES
      WRITE(*,110)
  110 FORMAT('0','SHAPE OF ARM AT T = 0.0 SEC')
      DO 120 J=1,NUMINT
         X = (FLOAT(J) - 1.0)*DX
         WRITE(*,115) X,W(1,J)
  115    FORMAT(' ',F8.4,7X,F8.4)
  120 CONTINUE
      WRITE(*,130)
  130 FORMAT('0','SHAPE OF ARM AT T = 2.0 SEC')
      DO 140 J=1,NUMINT
         X = (FLOAT(J) - 1.0)*DX
         WRITE(*,135) X,W(NTIME,J)
  135    FORMAT(' ',F8.4,7X,F8.4)
  140 CONTINUE
      STOP
      END
```

STRUCTURAL MODEL - SIMPLE SIMULATION

WN(1) =	2.8996 RAD/SEC	ALPHA(1) =	.7341		
WN(2) =	18.1731 RAD/SEC	ALPHA(2) =	1.0185		
WN(3) =	50.8903 RAD/SEC	ALPHA(3) =	.9992		
MN(1) =	.0302	AN(1) =	.7563	BN(1) =	.0000
MN(2) =	.0303	AN(2) =	.3448	BN(2) =	.0000
MN(3) =	.0305	AN(3) =	.2765	BN(3) =	.0000

MOTION OF ARM TIP WITH RESPECT TO TIME

.0000	.6879
.1000	.9105
.2000	.7347
.3000	.0053
.4000	.1295
.5000	.6796
.6000	.0802
.7000	-.8095
.8000	-.6544
.9000	-.4447
1.0000	-.7789
1.1000	-.6663
1.2000	-.4274
1.3000	-.9055
1.4000	-.9348
1.5000	.0772
1.6000	.4518
1.7000	-.1031
1.8000	.0304
1.9000	.6737
2.0000	.6822

SHAPE OF ARM AT T = 0.0 SEC

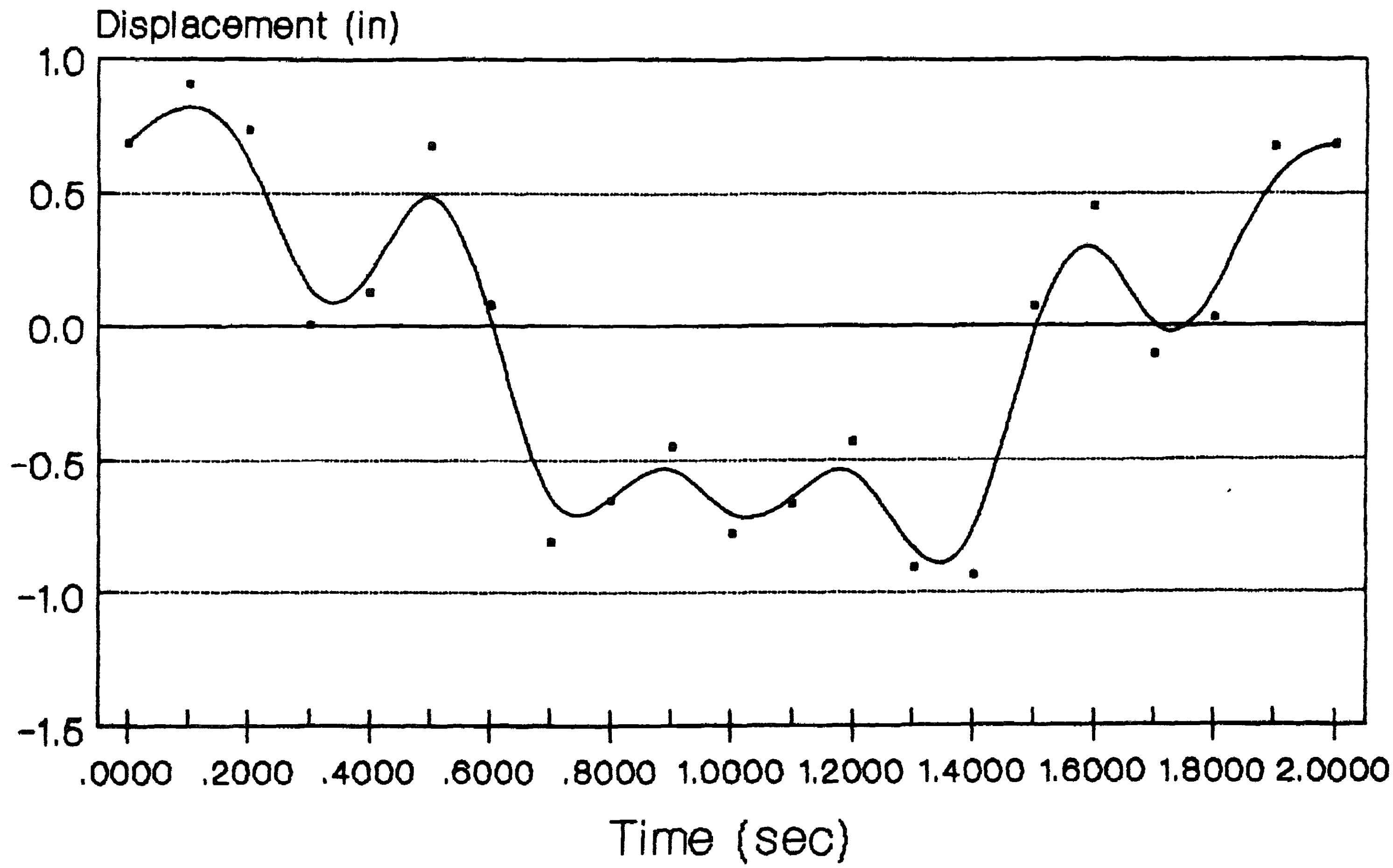
.0000	.0000
2.4000	.0305
4.8000	.1077
7.2000	.2104
9.6000	.3192
12.0000	.4177
14.4000	.4936
16.8000	.5399
19.2000	.5549
21.6000	.5418
24.0000	.5083
26.4000	.4643
28.8000	.4211
31.2000	.3885
33.6000	.3745
36.0000	.3833
38.4000	.4154
40.8000	.4678
43.2000	.5348
45.6000	.6100
48.0000	.6879

SHAPE OF ARM AT T = 2.0 SEC

.0000	.0000
2.4000	.0106
4.8000	.0380
7.2000	.0758
9.6000	.1179
12.0000	.1594
14.4000	.1966

```
 16.8000          .2272
 19.2000          .2507
 21.6000          .2683
 24.0000          .2822
 26.4000          .2954
 28.8000          .3113
 31.2000          .3327
 33.6000          .3618
 36.0000          .3997
 38.4000          .4460
 40.8000          .4996
 43.2000          .5583
 45.6000          .6198
 48.0000          .6822
Stop - Program terminated.
```

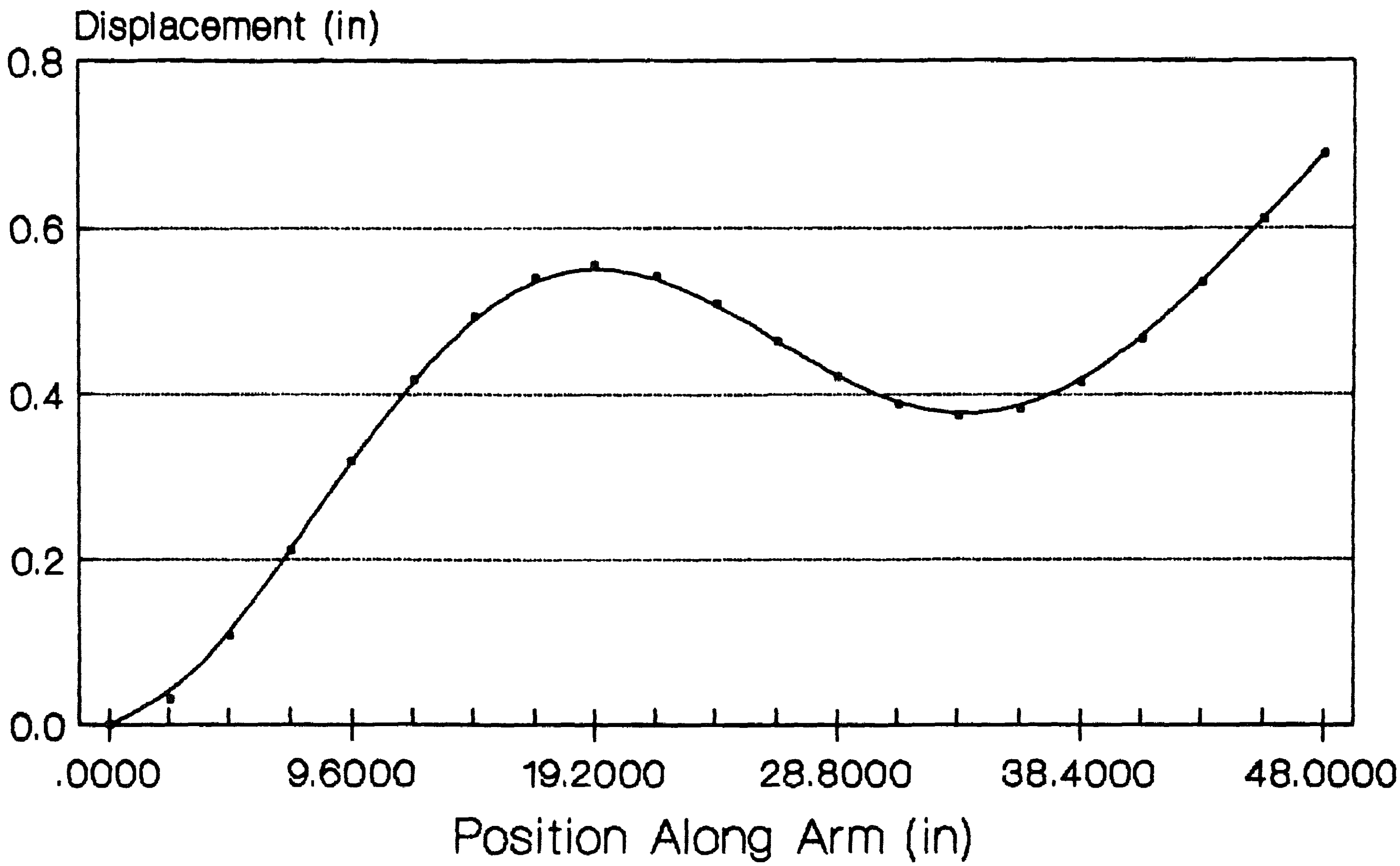
FIG. B-4: STRUCTURAL MODEL
Motion at arm tip



includes first three modes

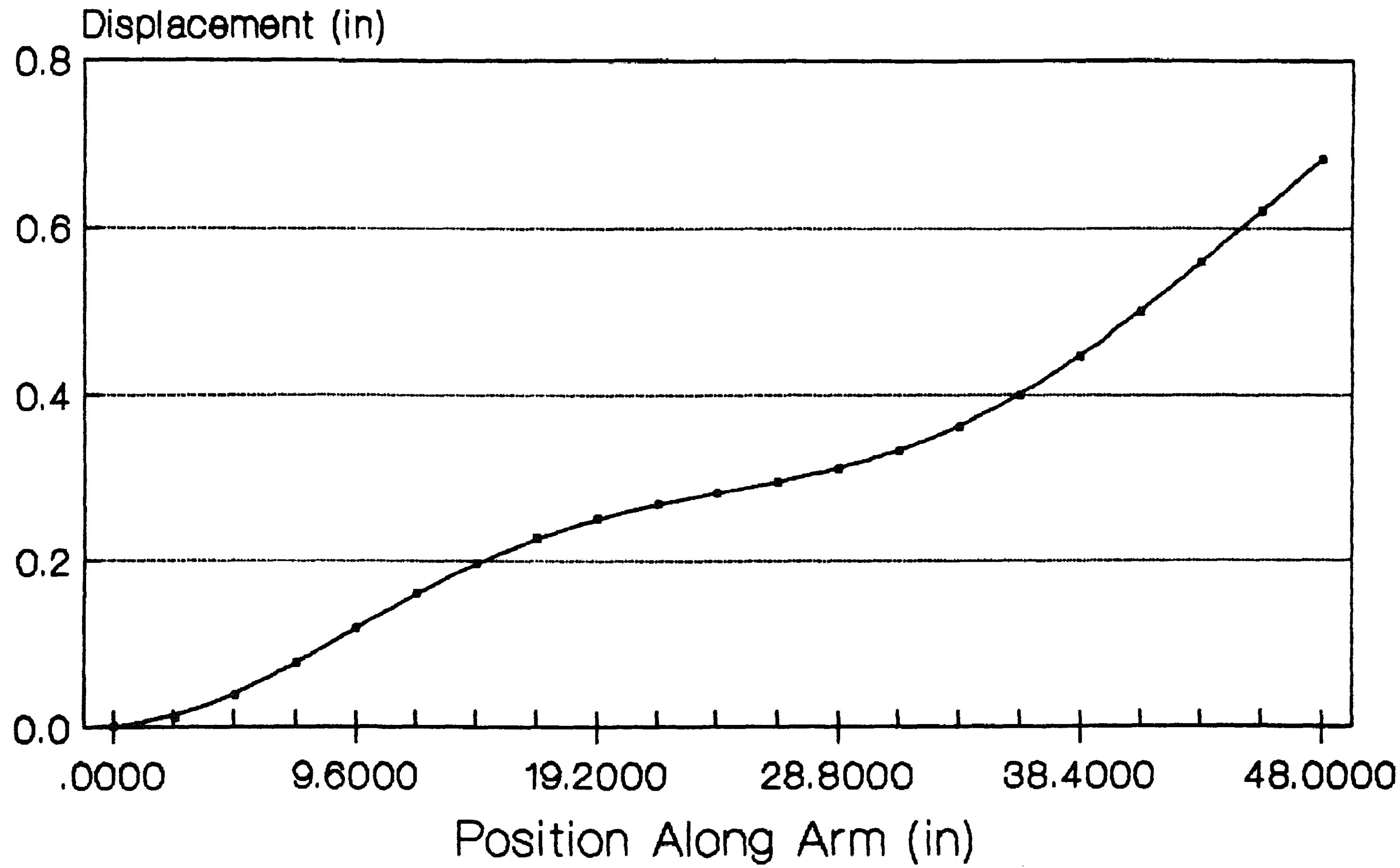
FIG. B-5: STRUCTURAL MODEL

Arm shape (t = 0 sec)



includes first three modes

FIG. B-6: STRUCTURAL MODEL
Arm shape (t = 2 sec)



includes first three modes

REFERENCES

1. Book, Wayne J., "Recursive Lagrangian Dynamics of Flexible Manipulator Arms", Intl. Jnl. of Robotics Research, vol. 3, #3, Fall 1984, pp. 87-101

2. Book, Wayne J., Majette, M., "Controller Design for Flexible Distributed Parameter Mechanical Arms Via Combined State Space and Frequency Domain Techniques", Transactions of the ASME, Jnl. of Dynamic Systems, Measurement, and Control, vol. 105, Dec. 1983, pp. 245 - 254

3. Davis, J. H., Hirschorn, R. M., "Tracking Control of a Flexible Robot Link", IEEE Transactions on Automatic Control, vol. 33, n. 3, March 1988, pp. 238 - 248

4. Den Hartog, J. P., Mechanical Vibrations, McGraw-Hill Book Co., New York, 1956, pp. 148 - 162

5. Flugge, W., Handbook of Engineering Mechanics, McGraw-Hill Book Co., New York, 1962, pp. 35-1 - 35-22, 61-1 - 61-18

6. Hastings, G. G., Book, W., "A Linear Dynamic Model for Flexible Robot Manipulators", IEEE Control Systems Magazine, vol. 7, n. 1, Feb. 1987, pp. 61 - 64

7. Hemati, N., Leu, M. C., "Automated Symbolic Derivation of Dynamic Equations of Motion for Robotic Manipulators", Transactions of the ASME, Jnl. of Dynamic Systems, Measurement, and Control, vol. 108, n. 3, Sep. 1986, pp. 172 - 179

8. Judd, R. P., Falkenburg, D., "Dynamics of Nonrigid Articulated Robot Linkages", IEEE Transactions on Automatic Control, vol. AC-30, n. 5, May 1985, pp. 499 - 502

9. Koplik, J., Leu, M. C., "Computer Generation of Robot Dynamic Equations and Related Issues", Jnl. of Robotic Systems, vol. 3, n. 3, 1986, pp. 301 - 319

10. Ower, J. C., Van de Vegte, J., "Classical Control Design for a Flexible Manipulator: Modeling and Control System Design", IEEE Journal of Robotics and Automation, vol. RA-3, n. 5, Oct. 1987, pp. 485 - 489

11. Paul, Richard P., Robot Manipulators, The MIT Press, Cambridge, Ma., 1981, pp. 13 - 20, 41 - 91

12. Sakawa, Y., Matsuno, F., Fukushima, S., "Modeling and Feedback Control of a Flexible Arm", Jnl. Robotic Systems, vol. 2, #4, 1985, pp. 453-472

13. Singh, S. N., Schy, A. A., "Control of Elastic Robotic Systems by Nonlinear Inversion and Modal Damping", Transactions of the ASME, Jnl. of Dynamic Systems,

Measurement, and Control, vol. 108, n. 3, Sep. 1986, pp. 180 - 189

14. Thomas, M., Tesar, D., "Dynamic Modeling of Serial Manipulator Arms", Transactions of the ASME, Jnl. of Dynamic Systems, Measurement, and Control, vol. 104, n. 3, Sep. 1982, pp. 218-228

15. Usoro, P., Nadira, R., Mahil, S., "A Finite Element/Lagrange Approach to Modeling Lightweight Flexible Manipulators", Transactions of the ASME, Jnl. of Dynamic Systems, Measurement, and Control, vol. 108, Sep. 1986, pp. 198 - 205

16. Walker, M. W., Orin, D. E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms", Transactions of the ASME, Jnl. of Dynamic Systems, Measurement, and Control, vol. 104, #3, Sep. 1982, pp. 205 - 211

17. Wang, T., Kohli, D., "Closed and Expanded Form of Manipulator Dynamics Using Lagrangian Approach", Transactions of the ASME, Jnl. of Mechanisms, Transmissions, and Automation in Design, vol. 107, pp. 223 - 225

18. Zalucky, A., Hardt, D. E., "Active Control of Robot Structure Deflections", Transactions of the ASME, Jnl. of Dynamic Systems, Measurement, and Control, vol. 106, Mar. 1984, pp. 63 - 69