NUMERICAL ANALYSIS OF RIGHT-HALF PLANE ZEROS FOR A SINGLE-LINK MANIPULATOR

A THESIS
Presented to
The Academic Faculty

by

Douglas Lynn Girvin

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Mechanical Engineering

Georgia Institute of Technology
March 1992

NUMERICAL ANALYSIS OF THE RIGHT-HALF PLANE ZEROS FOR A SINGLE-LINK MANIPULATOR

APPROVED:

Wayne Book

Wayne Book, Chairman

Nader Sadegh

Nader Sadegh

Massoud Tavakoli

Massoud Tavakoli

Date Approved by Chairperson 3 March 92

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. Wayne Book for his guidance and encouragement of my research project. His many insights and suggestions certainly improved the quality of this paper. I would also like to thank the other members of my reading committee, Dr. Sadegh and Dr. Tavakoli, for their time and suggestions. I would like to thank everyone in the research group for the enlightening discussions and their friendship.

I would like to especially thank my wife, Rhonda, whose love and support always keeps me going. I would also like to thank her for the time she spent improving the quality of this document. I would like to thank my parents, Larry and Louise, for their support during my collegiate studies which would not have been possible without them.

Finally, I would like to thank NASA for supporting this research through contract number NAG-1-623.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

The purpose of this research is to further develop an understanding of how nonminimum phase zero location is affected by structural link design. As the demand for light-weight robots that can operate in a large workspace increases, the structural flexibility of the links becomes more of an issue in controls problems. When the objective is to accurately position the tip while the robot is actuated at the base, the system is nonminimum phase. One important characteristic of nonminimum phase systems is system zeros in the right half of the Laplace plane. The ability to pick the location of these nonminimum phase zeros would give the designer a new freedom similar to pole placement.

The research targets a single-link manipulator operating in the horizontal plane and modeled as a Euler-Bernoulli beam with pinned-free end conditions. Using transfer matrix theory, one can consider link designs that have variable cross-sections along the length of the beam. A FORTRAN program was developed to determine the location of poles and zeros given the system model. The program was used to confirm previous research on nonminimum phase systems, and develop a relationship for designing linearly tapered links. The method allows the designer to choose the location of the first pole and zero and then defines the appropriate taper to match the desired locations. With the pole and zero location fixed, the designer can independently change the link's moment of

inertia about its axis of rotation by adjusting the height of the beam. These results can be applied to inverse dynamic algorithms currently under development at Georgia Tech.

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

As research for new applications for industrial robots proceeds, one major area of research is in robot flexibility. Traditionally, industrial robots are designed with stiff links, so the dynamics of the links can be assumed negligible in positioning the robot. In theory then as the robot moves, the links remain straight and do not bend. The tip position of the robot can be found geometrically from joint position at any given moment. In flexible robotics the links are no longer assumed to be rigid. As the robot moves, the links flex which can cause unwanted vibrations in the robot. These vibrations can cause error in positioning the tip of the robot.

Some of the applications motivating research in this field are assembly of space structures, inspection of large structures, and nuclear waste retrieval. When transporting things to outer space, weight is always a concern. Light-weight robots designed for space applications will be flexible and must be controlled as such. Large structures like airplanes and submarines require careful inspection to insure detection of flaws. The inspections can be laborious and repetitive which is ideal work for a robot. The large workspace dictates the links be as light as possible resulting in flexible links. An emerging area of research is remote handling of nuclear waste. Existing nuclear waste storage facilities are no longer safe and the waste needs to be removed and restored in

safer containers. The old containers are very large, while the access is usually quite small. Again, a light-weight slender robot with a large workspace is required. All of these applications are driving the research in the field of flexible robotics.

A common problem with flexible systems is how to control the system accurately to position the end-point. Rigid link robots are typically collocated systems; that is, the actuators and sensors are located at the same location (ie., a joint). With a flexible system this is not always the case. Most flexible systems are noncollocated. The system output (actuator torque) is generally located at the base of the system, while the output (tip position) is located at the end of the system. Noncollocated systems exhibit nonminimum phase behavior which results directly from the system zeros in the right-half of the s-plane (RHP zeros).

Controller design for collocated systems has been heavily researched and is well understood compared to controller design for noncollocated systems. In noncollocated systems, uncertainties from model inaccuracies and modal truncation present fundamental problems with system performance and stability [18]. The fundamental difference between collocated and noncollocated systems is the presence of these RHP zeros. To advance controller design for noncollocated systems, research needs to be conducted into the factors that affect the location of these RHP zeros. This research targets the relationship between RHP zeros and structural design.

1.2 Review of Related Research

Although research on RHP zeros is limited, there has been some notable research done in the past. Some of the research deals directly with the problems presented by

nonminimum phase systems, while other research examines different techniques to change the system characteristics from nonminimum phase to minimum phase.

In 1988, Nebot and Brubaker [13] experimented with a single-link flexible manipulator. The manipulator was constructed from thin plates connected by several bridges along their length. This provided flexibility in the horizontal plane, while maintaining stiffness in the vertical plane and torsional mode. They analytically determined the location of the first six zeros and determined three of them to be RHP zeros. They concluded these RHP zeros pose a formidable constraint in the controller design task.

In 1989, Spector and Flashner [19] investigated the sensitivity effects of structural models for noncollocated control systems. They considered a pinned-free beam with discrete end-point mass and inertia. They used transfer matrices to analyze the system. From the results they concluded the following. First, accurate dynamic modeling is critical in noncollocated control design. Poor modeling can result in interchanging the pole/zero order which produces phase errors resulting in closed-loop instability. Second, accurate modeling of zero location near the system bandwidth is critical in modeling noncollocated systems. Third, zeros are more sensitive to perturbations in system parameters and boundary conditions than modal frequencies. They suggest more research attention be given to modeling system zeros in noncollocated systems.

In 1990, Spector and Flashner [18] again studied modeling and design implications pertinent to noncollocated control. A similar system was used, a pinned-free beam without end-point mass, only the system was analyzed using wave number plane

theory. They also studied the effects of varying sensor/actuator separation distance. Most conclusions are identical to those drawn from the previous paper. In addition, they concluded all noncollocated systems are nonminimum phase above some finite frequency (the location of the lowest RHP zero dictates this frequency), and this frequency increases as sensor/actuator distance increases. Again they recommend more research into the modeling of zeros in noncollocated systems.

Also in 1990, Park and Asada [15],[14] investigated a minimum phase flexible arm with a torque actuation mechanism. Basically they used a cable mechanism to transfer the torque actuation point from the base to the tip of the arm. Since the sensor and actuator are located at the same point, the system is minimum phase. They concluded the inverse dynamics solution does not diverge because the RHP zeros are relocated to the LHP by the torque transmission mechanism. Also end-point feedback control can be stabilized for this system with simple a P-D controller. Unfortunately, implementation of the transmission device on multi-link systems could be difficult.

In 1991, Park, Asada, and Rai [1] expanded their previous work on a minimum phase flexible arm with a torque transmission device. In this research they integrate structure and control design using Finite Element Analysis (FEA) to design the shape of the arm while constraining pole and zero location. Essentially, they use the FEA program to generate a design that will increase the fundamental natural frequency and use the torque transmission device to eliminate the RHP zeros. A prototype of the new system had not been tested at that point, and the main contribution was a method to evaluate nonuniform beams for design applications.

1.3 Proposed Method of Approach

The underlying issue in noncollocated control is how to deal with the RHP zeros in the control algorithm. A major step in solving the problem is understanding what design parameters can be used to change the location of these RHP zeros. This research targets the relationship between RHP zero location and structural design. Specifically, how do changes in the shape of the structure (link) affect the location of these zeros?

Traditionally links are designed with uniform properties along the length because analytic solutions to this problem exist. A link with variable cross-section cannot be solved analytically, but with aid of a computer a numerical approximation can be found. The key to an accurate numerical solution is a good model of the system.

The research presented in this thesis models a single-link flexible rotary manipulator as a pinned-free beam. Transfer matrix theory was used to generate a beam with variable cross-section. FORTRAN code was written to generate the model and evaluate the system for the location of RHP zeros. The program was used to examine the relationship between link shape and RHP zero location. This relationship can be directly applied to controller design using the inverse dynamics approach researched here at Georgia Tech.

The research is presented as follows. Chapter 1 discusses the relevant research and the method of research. Chapter 2 presents some of the characteristics of nonminimum phase systems and some of the control methods currently used on these systems. Chapter 3 presents transfer matrix theory, describes modeling issues of concern to this research, and discusses computer implementation. Chapter 4 presents the

results pertinent to the relationship being studied. Finally, Chapter 5 presents the conclusions and addresses areas of future research. Appendices A, B, and C contain derivations necessary for implementation of the ZERO program. Appendix D contains pinned-pinned natural frequencies for selected designs. Appendix E contains the source code of the ZERO program.

CHAPTER 2

NONMINIMUM PHASE SYSTEMS

2.1 System Characteristics

As mentioned before, a system is considered nonminimum phase if there are system zeros or poles located in the right half of the Laplace plane. Figures 2.1a and 2.1b graphically express the difference between minimum phase and nonminimum phase systems.

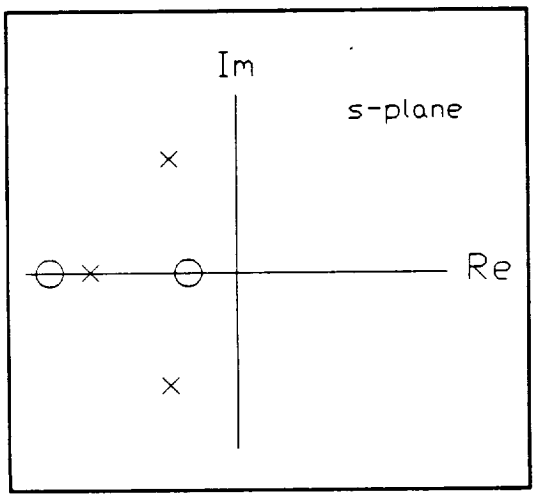


Figure 2.1a: Minimum Phase Pole/Zero Pattern

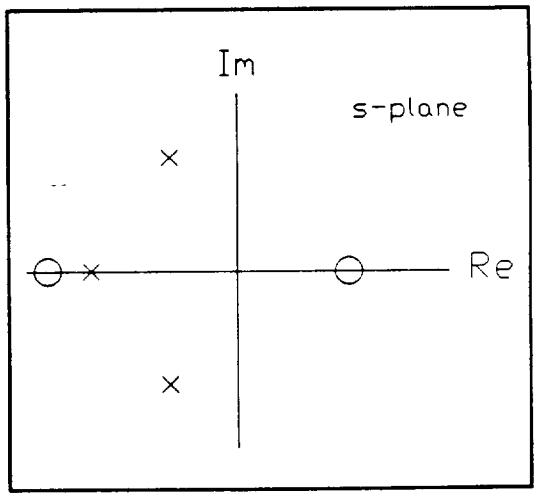


Figure 2.1b: Nonminimum Phase Pole/Zero Pattern

This is the case in the continuous-time domain. In the discrete-time domain (z-transform), the nonminimum phase zeros would lie outside the unit circle.

Often RHP zeros are called unstable zeros, but this is not good terminology. RHP zeros do not cause the plant to go unstable. Poles in the RHP will cause the system

response to exponentially increase resulting in instability, but zeros do not cause this. It is the controller design that can cause the zeros to have an effect on system stability. For example, when using an inverse dynamics algorithm, the RHP zeros will become unstable poles in the inverse system. Now the controller has unstable poles which can cause the entire system to go unstable.

A noticeable characteristic of a nonminimum phase system is the time response to a step input. Figure 2.2 shows the difference between minimum phase (MP) and nonminimum phase (NMP) response of the tip position for a single-link flexible manipulator.

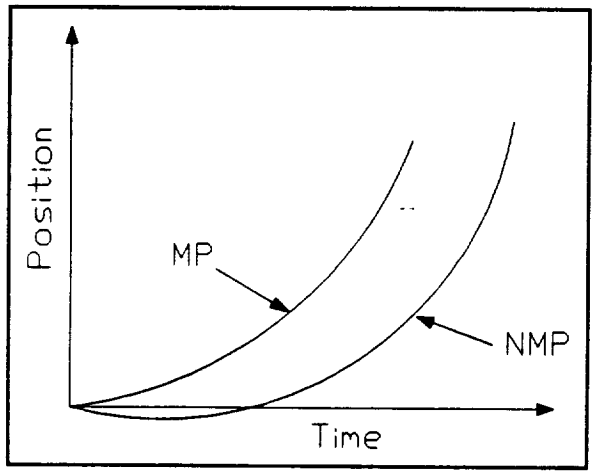


Figure 2.2: MP vs. NMP Time Response

Notice the tip of the NMP system initially starts to move in the direction opposite to the command. This type of response can be verified in Park and Asada's paper [14].

It has been stated that RHP zeros are indicative of a NMP system, but what physical phenomenon is responsible for NMP behavior? Spector and Flashner [18] concluded that NMP behavior is an inescapable result of the finite wave propagation

speed of elastic deformation in the structure. This wave propagation speed directly results in a time delay between system input and the corresponding system output. The time delay affects the system by reducing the phase margin. If the phase lag from the time delay exceeds the system phase margin (at the cutoff frequency), the system will be unstable.

These are some of the more prominent characteristics of nonminimum phase systems. Of interest in this research is the control of nonminimum phase systems and how to advance the research in this area. The following section describes some of the current techniques used to control nonminimum phase systems.

2.2 Control of Nonminimum Phase Systems

One method of controlling a nonminimum phase system studied by Misra [12] in 1989 is augmenting a nonminimum phase plant to make the overall system minimum phase. He used a "feedthrough" compensator so the augmented system was minimum phase. A feedthrough compensator was added so the poles of the compensated system move to the minimum phase zeros.

In 1987, Bayo [2] presented a structural finite element technique based on Bernoulli-Euler beam theory for open-loop control of flexible manipulators. The differential equations of motion are integrated in the frequency domain to determine the necessary torques for desired tip motion. The computed torque reproduced the desired trajectory without any overshoot, but closed-loop control was not investigated.

Another control algorithm investigated at Georgia Tech by Kwon and Book [5],[9],[8] used an inverse dynamic method to deal with a NMP flexible arm. The

method is similar to Bayo's, only integration was carried out in the time domain. The dynamic equations of motion for a flexible manipulator can be written as:

$$\begin{bmatrix} M_{rr} & M_{rf} \\ M_{rf}^T & M_{ff} \end{bmatrix} \begin{Bmatrix} \ddot{q}_r \\ \ddot{q}_f \end{Bmatrix} + \begin{bmatrix} D_{rr} & D_{rf} \\ D_{rf}^T & D_{ff} \end{bmatrix} \begin{Bmatrix} \dot{q}_r \\ \dot{q}_f \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_{ff} \end{bmatrix} \begin{Bmatrix} q_r \\ q_f \end{Bmatrix} = \begin{bmatrix} B_r \\ B_f \end{bmatrix} \tau \tag{2.1}$$

where,

q_r - Rigid body motion coordinate

q_f - Flexible motion coordinate

After some manipulation the inverse dynamics equations can be obtained in the following form:

$$\begin{aligned} \dot{X}_i &= [A_i]\, X_i + [B_i]\, \dot{q}_{ir} \\ \tau &= [C_i]\, X_i + [F_i]\, \dot{q}_{ir} \end{aligned} \tag{2.2}$$

where,

$$X_i = [q_f, \dot{q}_f]^T \qquad q_{ir} = [q_r, \dot{q}_r]^T$$

For the forward dynamic equations, the input is torque, and the outputs are all states. For the inverse dynamics system, the input is end-point desired trajectory, and the output is torque. The problem addressed is how to integrate these equations since the matrix $[A_i]$ has positive real poles. The RHP poles come from the RHP zeros in the original system. Their approach is to relax the solution range to include noncausal solutions

allowing a unique stable solution of the inverse dynamic equations.

To better understand the inverse dynamics solution, some terminology needs to be defined. According to Kwon [8], a causal system is one in which the system output (impulse response) occurs after the system input (impulse). An anticausal system has the output (backward impulse response) before an input is applied. A noncausal system is a combination of both a causal system and a anticausal system.

To illustrate these concepts Figure 2.3 shows the motion of a flexible arm moving from point A to point B.

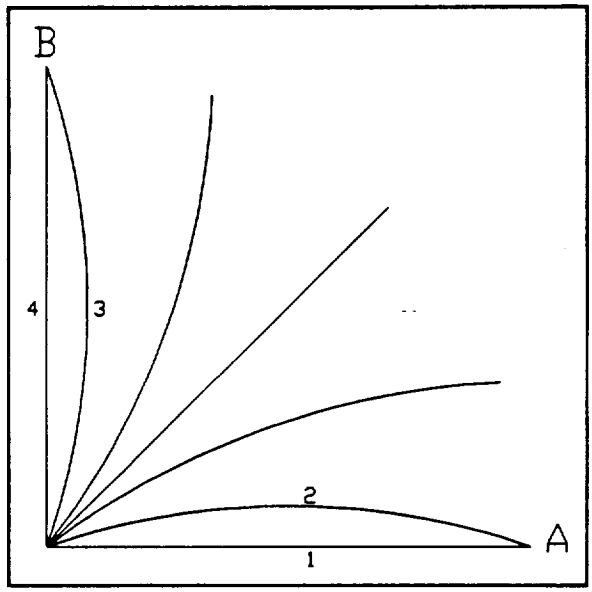


Figure 2.3: Flexible Link Motion

The two areas of interest on this curve are the start of motion and the end of motion. Motion starts as the arm moves from position 1 to 2, but the end-point does not move. The torque provided is applied to preshape the beam. This is the anticausal part of the inverse solution. The torque (output of the inverse system) occurs before the end-point (input to the inverse system) moves. When motion stops, the arm moves from position

3 to 4, again the end-point does not move. This represents the causal part of the inverse solution. The tip has stopped moving, but the torque continues to be applied. The torque applied between positions 4 and 5 is used to release the stored energy in the arm.

Since the motion can be divided into causal and anticausal parts, the solution to Equation 2.2 can be divided into both causal and anticausal parts. Of interest to this research is the anticausal solution. The poles of the anticausal system are unstable and a direct result of the RHP zeros from the forward dynamic system. The ability to place these RHP zeros would be equivalent to placing the poles of the inverse anticausal problem. This would give the designer some freedom in choosing the location of the anticausal poles, and allow the system to be designed for specific needs. One benefit could be minimizing the time of preshaping and the amount of energy provided by the actuator to preshape the beam before tip motion begins.

CHAPTER 3

TRANSFER MATRIX METHOD

3.1 Transfer Matrix Theory

Transfer matrices describe the interaction between two serially connected elements. These elements can be beams, springs, rotary joints, or many others. In 1979 Book, Majette, and Ma [6] and Book [4] (1974) used transfer matrices to develop an analysis package for flexible manipulators. They used transfer matrices to serially connect different types of elements to model the desired manipulator. Of interest in this paper is how to connect similar types of transfer matrices (beam elements) to model a beam with different cross-sectional area. Pestel and Leckie [16] provide an in depth discussion of transfer matrix derivations and applications.

Transfer matrices can be mathematically expressed by Equation 3.1. The state vector u_i is given by the state vector u_{i-1} multiplied by the transfer matrix B.

$$u_i = [B_i]u_{i-1} \tag{3.1}$$

When elements are connected serially, the states at the interface of two elements must be equal. By ordered multiplication of the transfer matrices, intermediate states can be eliminated to determine the transfer matrix for the overall system.

The concept of state vector in transfer matrix theory is not to be confused with the state space form of modern control theory. The state equation in modern control

theory relates the states of the system as a function of time. In transfer matrix theory the state equation relates the states as a function of position. The independent variable in transfer matrix theory is frequency, not time. The elements of the matrix B depend on the system driving frequency; therefore, the states will change as the system frequency changes. The transfer matrix B essentially contains the transformed dynamic equations of motion that govern the element in analytic form. Therefore, analytical solution of the transfer matrix alone does not involve numerical approximations. This is desirable since numerical approximations introduce error into the solution.

3.2 Modeling of a Nonuniform Beam

A single-link manipulator as pictured in Figure 3.1 can be thought of as a beam with torque applied at one end and free at the other end. There are several steps to determine the RHP zeros and imaginary poles of this system. First, develop a model for the beam. Second, determine the appropriate boundary conditions. Third, determine the system input and output. Forth, solve for the system zeros. The following sections will discuss each of these steps in more detail.

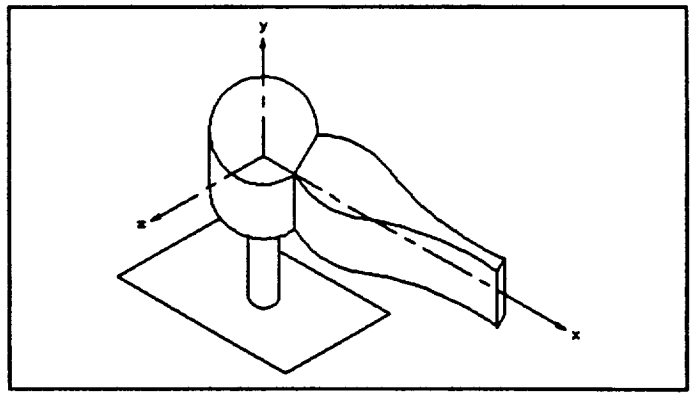


Figure 3.1: Single-Link, Flexible Manipulator

3.2.1 Element Approach to Modeling

A link with nonuniform cross-sections can be modeled as a series of discrete elements. While the shape of these elements is similar, the size can vary to allow for changes in cross-section. The appropriate element to model a flexible link is an Euler-Bernoulli beam element. The Euler-Bernoulli model neglects the effects of rotary inertia and shear deformation in the element. [11]. This assumption is generally valid for modeling beams whose length is roughly ten times the height. Flexible manipulators have long, slender links which are appropriately modeled under the Euler-Bernoulli assumption.

Transfer matrices are derived from the equation of motion for a given element. For a uniform Euler-Bernoulli beam element, the equation of motion transformed to the frequency domain has the form:

$$\frac{d^4 w(x,\omega)}{dx^4} = \frac{\mu \omega^2}{EI} w(x,\omega)$$

where,

μ	=	mass density per unit length
ω	=	frequency in radians/second
E	=	Young's modulus
I	=	Cross sectional area moment of inertia

Notice the equation is fourth order thus requiring four states to describe the solution in transfer matrix form. The state vector for the Euler-Bernoulli element is:

$$u = \begin{bmatrix} -w \\ \psi \\ M \\ V \end{bmatrix} = \begin{bmatrix} displacement \\ slope \\ moment \\ shear\ force \end{bmatrix} \tag{3.3}$$

The first two elements of the state vector are displacements (w and ψ) while the last two elements are forces (V and M). This arrangement of states is characteristic of transfer matrix theory. Figure 3.2 shows how these are defined for transfer matrix theory.

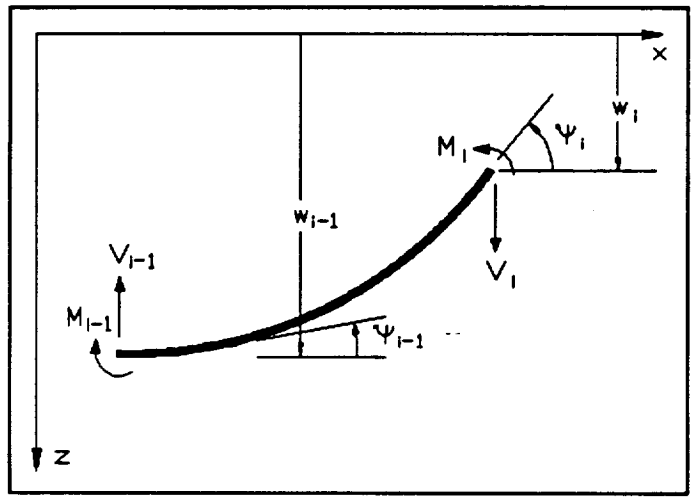


Figure 3.2: State Variables and Sign Conventions

An analytic solution to Equation 3.2 can be found when the element has uniform properties (ie. constant cross-section, mass density, and stiffness). Equation 3.4 gives the transfer matrix for a uniform Euler-Bernoulli element. Each element of Equation 3.4 is a function of frequency and must be reevaluated as the frequency of interest changes.

$$TM = \begin{bmatrix} C_0 & lC_1 & aC_2 & alC_3 \\ \frac{\beta^4 C_3}{l} & C_0 & \frac{aC_1}{l} & aC_2 \\ \frac{\beta^4 C_2}{a} & \frac{\beta^4 lC_3}{a} & C_0 & lC_1 \\ \frac{\beta^4 C_1}{al} & \frac{\beta^4 C_2}{a} & \frac{\beta^4 C_3}{l} & C_0 \end{bmatrix} \tag{3.4}$$

where,

$$C_0 = \frac{1}{2}(\cosh\beta + \cos\beta) \tag{3.5}$$

$$C_1 = \frac{1}{2\beta}(\sinh\beta + \sin\beta) \tag{3.6}$$

$$C_2 = \frac{1}{2\beta^2}(\cosh\beta - \cos\beta) \tag{3.7}$$

$$C_3 = \frac{1}{2\beta^3}(\sinh\beta - \sin\beta) \tag{3.8}$$

and

$$\beta^4 = \frac{\omega^2 l^4 \mu}{EI} \tag{3.9}$$

$$a = \frac{l^2}{EI} \tag{3.10}$$

With the transfer matrix for the fundamental beam elements, one can combine these elements serially to generate a model for the link. Figure 3.3 illustrates how a simple model can be constructed for a tapered beam. Although only two elements are considered here, more elements can be added to better approximate the shape of the link.

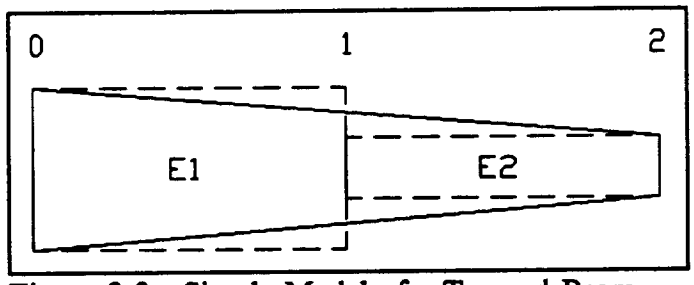


Figure 3.3: Simple Model of a Tapered Beam

Element A can be represented by the equation:

$$u_1 = [B_1]u_0 \tag{3.11}$$

Similarly for element B,

$$u_2 = [B_2]u_1 \tag{3.12}$$

Since the states at interface u_1 are the same for both elements, u_1 can be eliminated to obtain an overall transfer matrix for the beam:

$$u_2 = [B_2][B_1]u_0 \tag{3.13}$$

Eliminating one state simply illustrates the point that this multiplication can be carried out to eliminate all intermediate states in a model with more elements.

As previously mentioned, transfer matrices themselves are not numerical approximations. The transfer matrix for a Euler-Bernoulli beam contains the analytic solution for a uniform beam element. It is not an assumed modes solution. The approximation made in using transfer matrix theory involves the modeling of the beam and solution of the equations. To generate the model of a link with variable cross-

section, the size of the elements must vary. The interface of two different size elements will be discontinuous. In Figure 3.3, interface 1 is discontinuous between elements A and B. These discontinuities are the major approximation when using transfer matrices to model a beam. This approximation can be minimized by using more elements to model a nonuniform beam. As more elements are added to the model, the discontinuities between elements will decrease thus reducing the effects of this approximation on the results.

Transfer matrix theory is similar to Finite Element Analysis (FEA). In FEA, first the system must be discretized. Then an appropriate interpolation function must be selected to describe each element (ie. element stiffness). Next the system matrices must be assembled to produce a set of linear algebraic equations. Finally the linear equations are solved to get an approximate solution to the system under consideration.

Like FEA, when using transfer matrices the system must be first discretized into a finite number of elements. Unlike FEA though, there is no approximate interpolation function needed to describe each element. Each matrix contains the analytic equations describing the element. The two methods also differ in the method of solution of the numerical system of equations (for this application). As will be explained later, a root finder is used to determine the location of poles and zeros. An equation is extracted from the overall transfer matrix based on the desired input and output and the boundary conditions. The root finder then searches this equation to determine the location of poles and zeros. Although both are numerical methods to find an approximate solution to a continuous system, transfer matrix theory does reduce some approximations by using

exact solutions to the partial differential equations to describe the individual elements.

3.2.2 Boundary Conditions

The second step in finding the RHP zeros and imaginary poles of a system is applying the appropriate boundary conditions. As Figure 3.4 shows, there are several boundary conditions that can be applied to model a flexible link. The clamped-free condition corresponds to a rigid coordinate attached at the hub. The pinned-pinned condition corresponds to a rigid coordinate which passes through the end-point of the manipulator. In this research, the pinned-free boundary condition was chosen to model the flexible link. This corresponds to a rigid coordinate passing through the center of mass of the beam. This boundary condition was chosen because it naturally describes a flexible link and previous research by Spector and Flashner [18],[19] also used these boundary conditions to model the flexible link.

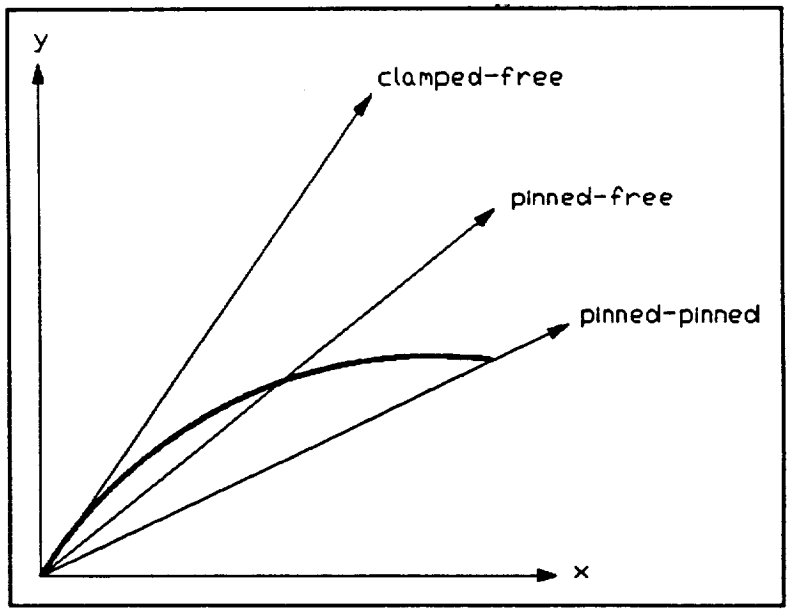


Figure 3.4: Boundary Conditions for a Flexible Link

A pinned-free boundary condition implies that:

At x=0 (base): $w=0 \quad M=0$ (pinned)

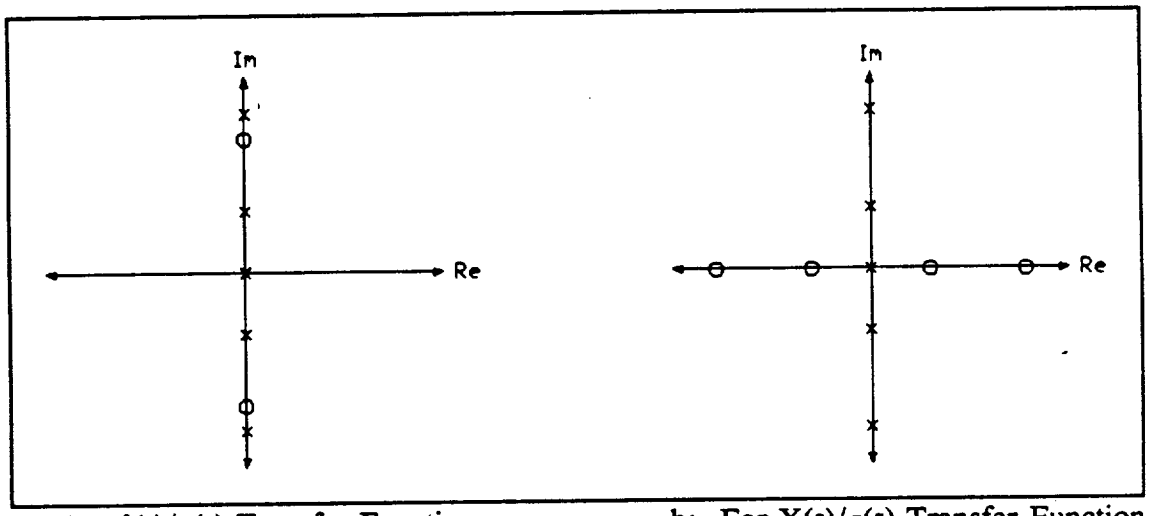
At x=L (tip): $V=0 \quad M=0$ (free)

These boundary conditions are applied to the overall transfer matrix for the system and the appropriate state variables are set to zero.

$$\begin{bmatrix} -w \\ \psi \\ 0 \\ 0 \end{bmatrix}_{x=L} = \begin{bmatrix} B_{11} & \cdots & B_{14} \\ \vdots & \ddots & \vdots \\ B_{41} & \cdots & B_{44} \end{bmatrix} \begin{bmatrix} 0 \\ \psi \\ 0 \\ V \end{bmatrix}_{x=0} \tag{3.14}$$

3.2.3 System Input and Output

When the system zeros are of interest, one must chose the system input and output. Unlike the natural frequency calculation which depends only on the boundary conditions, the location of system zeros will change as the input/output relationship changes. To illustrate this point, consider a single-link flexible manipulator modeled as a continuous system. Figure 3.5 shows the pole zero pattern of two different input/output relationships for the same system. Figure 3.5.a shows the transfer function between the joint angle, θ(s), and joint torque, τ(s), to be minimum phase. This is expected since these two are collocated. Figure 3.5.b shows the transfer function between tip position, X(s), and joint torque, T(s), to be nonminimum phase. The RHP zeros are a result of the noncollocated output relationship. Since this research targets the location of RHP zeros the system output is tip position, and the system input is joint



a: For $\theta(s)/\tau(s)$ Transfer Function b: For $X(s)/\tau(s)$ Transfer Function

Figure 3.5: Pole/Zero Patterns For Different Input/Output Relationships

torque. Considering the system input and output, the overall system transfer matrix will have the form:

$$\begin{bmatrix} -w \\ \psi \\ 0 \\ 0 \end{bmatrix}_{x=L} = \begin{bmatrix} B_{11} & \cdots & B_{14} \\ \vdots & \ddots & \vdots \\ B_{41} & \cdots & B_{44} \end{bmatrix} \begin{bmatrix} 0 \\ \psi \\ \tau \\ V \end{bmatrix}_{x=0} \tag{3.15}$$

In the above equation, w_L is the system output which corresponds to tip position, and τ is the system input corresponding to joint torque at the base of the manipulator.

3.2.4 Zero Function

With the system input and output chosen, Equation 3.15 can be simplified to determine the function that relates system output to system input. Appendix B contains a complete derivation of the zeros function for the system under consideration. The

equation used to determine zero location is:

$$w_L = -\left[\frac{B_{12}B_{44}B_{33} - B_{12}B_{34}B_{43} + B_{13}B_{34}B_{42} - B_{13}B_{44}B_{32} + B_{14}B_{43}B_{32} - B_{14}B_{33}B_{42}}{B_{34}B_{42} - B_{44}B_{32}}\right]\tau \tag{3.16}$$

Where B_{ij} are elements of the overall transfer matrix in Equation 3.15. When the function inside the brackets is zero (for a given frequency), the output will always be zero regardless of the input; therefore, the zeros of the bracketed term are the system zeros. As derived in Appendix A, this function is real.

$$f(\omega) = \frac{B_{12}B_{44}B_{33} - B_{12}B_{34}B_{43} + B_{13}B_{34}B_{42} - B_{13}B_{44}B_{32} + B_{14}B_{43}B_{32} - B_{14}B_{33}B_{42}}{B_{34}B_{42} - B_{44}B_{32}} \tag{3.17}$$

To search for RHP zeros, one must consider what type of frequency to input into Equation (3.17). Using the relationship which defines the Laplace variable, s

$$s = j\omega \tag{3.18}$$

one can easily determine ω should have the form:

$$\omega = 0 - jb \qquad \text{where} \quad 0 \le b \le \infty \tag{3.19}$$

Purely imaginary negative values of ω will result in purely real positive values of s.

Thus searching Equation 3.17 with frequencies of the form of Equation 3.19 one can find the location of the RHP zeros.

3.2.5 Natural Frequency Function

Although the location of RHP zeros is of primary concern in this research, knowledge of pole location will help in analysis of the results. Since the system damping is ignored, the poles will lie on the imaginary axis of the s-plane in complex conjugate pairs. The location of these poles can be determined by simply searching the positive imaginary axis of the s-plane. Considering the applied boundary conditions, one can extract two homogeneous equations from Equation 3.14 to get the homogeneous system:

$$\begin{Bmatrix} 0 \\ 0 \end{Bmatrix} = \begin{bmatrix} B_{32} & B_{34} \\ B_{42} & B_{44} \end{bmatrix} \begin{Bmatrix} \Psi \\ V \end{Bmatrix} \tag{3.20}$$

The poles (eigenvalues) of the system are those values of ω which make the determinant of the sub-transfer matrix in Equation 3.20 equal to zero (see reference [6] for a detailed explanation). For a two by two matrix this determinant is simply:

$$g(\omega) = B_{32}B_{44} - B_{34}B_{42} \tag{3.21}$$

Referring to Equation 3.18, one finds that Equation 3.21 is the denominator of the input/output transfer function which is to be expected. To find the values of the purely complex poles, one must search Equation 3.21 for its roots. According to the definition

of s, ω must have the form:

$$\omega = b + j0 \tag{3.22}$$

Searching over a range of values for b will give the poles in that range. With the zero and natural frequency functions determined, the problem remains to implement a computer solution to find the RHP zeros and imaginary poles.

3.3 Computer Implementation

Like Finite Element Analysis, the solution to pole/zero location of a flexible link using transfer matrix theory is computationally intensive. As the number of elements in the model increases, so does the number of computations. With the availability of computers today, the problem is fairly easy to solve if the proper algorithm can be implemented. Previous research by Book and others [6],[10] used transfer matrices to model systems and this provided some insight on how to realize a computer solution using transfer matrices, especially the DSAP [6] package. The program structures are purposely similar to aid in combining the programs for future research.

The code is written in FORTRAN (FORmula TRANslation) language as this language is well suited for solving scientific and engineering problems. A Digital VaxStation II was used to run the FORTRAN software. Vax FORTRAN is very compatible with FORTRAN 77, a popular ANSI standard version. The previous research by Book and others was also written in FORTRAN.

Program ZERO is the main module which handles control of the other subroutines. Figure 3.6 presents a flow chart for the ZERO program showing how the different subroutines are employed, and Table 3.1 describes the function of each program module. Each subroutines was designed to accomplish a specific task simplifying the programming job. The following is a quick overview of the program structure.

Once the model and computation parameters are input, the main search interval is divided into subintervals. Each subinterval is sent to the root finder, ZFALSE, individually and ZFALSE checks for roots. ZFALSE calls function F to evaluate the zero or pole function for a given frequency. To evaluate the function, F needs the overall transfer matrix for the given frequency. Function F calls subroutine BUILD to generate the overall transfer matrix. Subroutine BUILD first calls subroutine BEAM4 to generate the transfer matrix for the i^{th} element. Once BEAM4 returns the element transfer matrix, BUILD calls subroutine MUL to multiply the updated overall transfer matrix with the new element transfer matrix. BUILD repeatedly calls BEAM4 and MUL until the all intermediate states are eliminated. Once BUILD returns the overall transfer matrix to F, F evaluates the appropriate function and returns the value to ZFALSE. This entire process is repeated each time ZFALSE needs the function value for a particular frequency. After SUBDIV outputs the results from ZFALSE, it continues to send down the next subinterval until the entire frequency range has been searched. Appendix E contains the source code of the main program and all subroutines for reference if needed. The following sections will discuss each unit in more detail.

Table 3.1: Program Modules and Their Functions

Module	Function
ZERO	1. Input system model
SUBDIV	1. Input computation parameters 2. Input search interval 3. Divide search interval into subintervals
ZFALSE	1. Regula-falsi root finder
F	1. Generate complex frequency a. [0,-b] for zeros b. [+b,0] for poles 2. Evaluate function
BUILD	1. Generate overall transfer matrix
BEAM4	1. Generate element transfer matrix 2. Extract real matrix
MUL	1. Multiply two square matrices

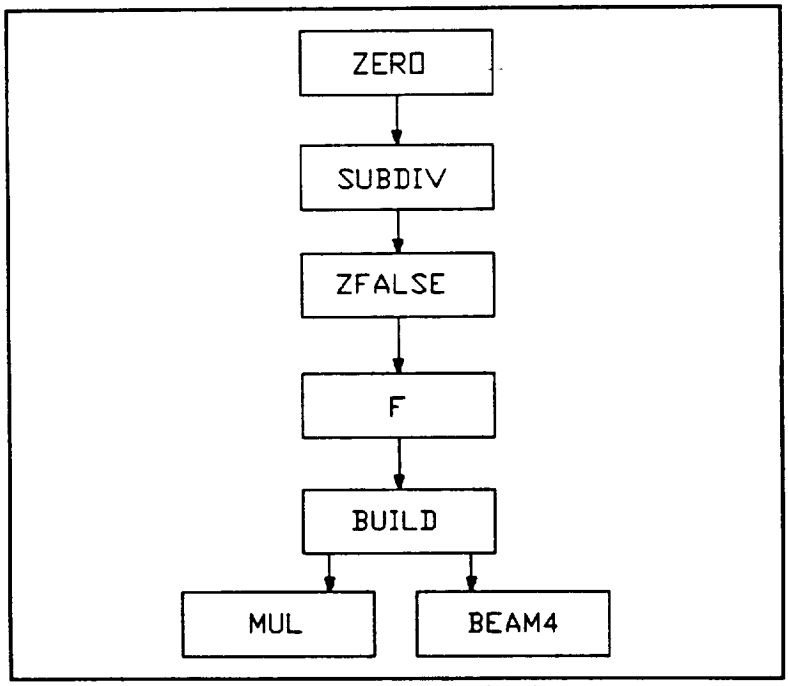


Figure 3.6: ZERO Program Organization

3.3.1 Main Program

The main program module ZERO handled control of the subroutines and user interface for entering the model. The model is entered as elements with each element needing 5 parameters to describe it. The first element corresponds to the base element while the last element corresponded to the tip element. For each element the five parameters are stored in a two dimensional array, EP, in the following order:

$EP(1,E_i)$ = Element length (in)

$EP(2,E_i)$ = Element mass per unit length (in^2-lb-sec^2)

$EP(3,E_i)$ = Element area moment of inertia (in^4)

$EP(4,E_i)$ = Element modulus of elasticity (psi)

$EP(5,E_i)$ = Element damping factor

E_i in the second index of array EP, corresponds to the i^{th} element of the model. The units in parentheses are only one choice; others can be used as long as they are consistent. The model can be entered into the program from the keyboard or through an input file. Input files must have the extension ".inp" to be recognized by the program. It should be noted that the damping factor must be zero for this program. It is included as a dummy parameter to maintain similarities with the program DSAP. Once the model is entered in, ZERO passes control on to the subroutine SUBDIV.

3.3.2 Subroutines

With the model entered, subroutine SUBDIV handles input of the computation parameters for the root finder, input of the frequency range, and output of the results.

The computation parameters eps, nsig, and itmax are discussed under ZFALSE. Along with the frequency range, the user inputs the number of subdivisions. SUBDIV sends one subdivision at a time down to ZFALSE to check for roots. The results of the search in each subinterval are printed to the screen. The possibilities are: a) a root was found, b) no root was found, or c) the program did not converge. In addition to screen output, SUBDIV generates two output files. The output file with extension ".out", contains the location of all computed zeros and poles. This allows the user to get a hardcopy of the results. The output file with extension ".dat" contains the values of the function being searched (zero function or natural frequency function) at each interval. This allows the user to examine the function values for more information if needed.

Since the damping factor is zero, the poles and zeros are a priori known not to have both real and complex parts. The nonminimum phase zeros will lie on the positive real axis, and the poles will lie on the imaginary axis in complex conjugate pairs. Although the two do not lie on the same axis, a two-dimensional search can be avoided by performing two one-dimensional searches along both axes. This is accomplished by changing the form of the frequency (ie. purely complex or real). SUBDIV prompts the user first for the frequency range and number of subdivisions for the zeros search. Once this is completed, SUBDIV prompts the user for the frequency range and number of subdivisions for the poles search. Both searches use the same computation parameters.

Subroutine ZFALSE determines whether or not a root lies within the specified subinterval. It first checks the values passed down defining the subinterval. If these

have the same sign, then it assumes no root lies in the subinterval and passes the appropriate flag back to SUBDIV. If the signs are opposite, then it begins to iterate in on the suspected root. The estimation of the root is determined by the Regula-Falsi method as illustrated in Figure 3.7. Using the two values that define the subinterval, x_R and x_L, it linearly interpolates for the first estimate of the root and determines the function value at the new estimate. The estimate is judged to be a root if it passes one of two criteria. Criteria 1 tests the magnitude of the function at the estimated root, x_{est}. IF $x_{est} \leq$ eps then x_{est} is a root; therefore, eps is the numerical value of "zero" input by the user. Criteria 2 tests the number of significant digits which do not change from one estimation to the next. If $|x_{est}-x_{est,old}| \leq 10^{-(nsig)}$ then x_{est} is a root; therefore, nsig stores the number of significant digits of the root. If neither of these tests are passed before the maximum number of iterations, itmax, is reached, then ZFALSE returns with an appropriate message.

ZFALSE calls the subprogram F to evaluate the function at the given frequency. The frequency passed to F is a real number and based on the type of search (zero or pole), F will generate the proper complex frequency. It next calls BUILD to assemble the overall transfer matrix for that frequency. With the overall transfer matrix, F evaluates the proper function and returns the value to ZFALSE. F knows which function (zero or pole) to use based on a flag set in SUBDIV.

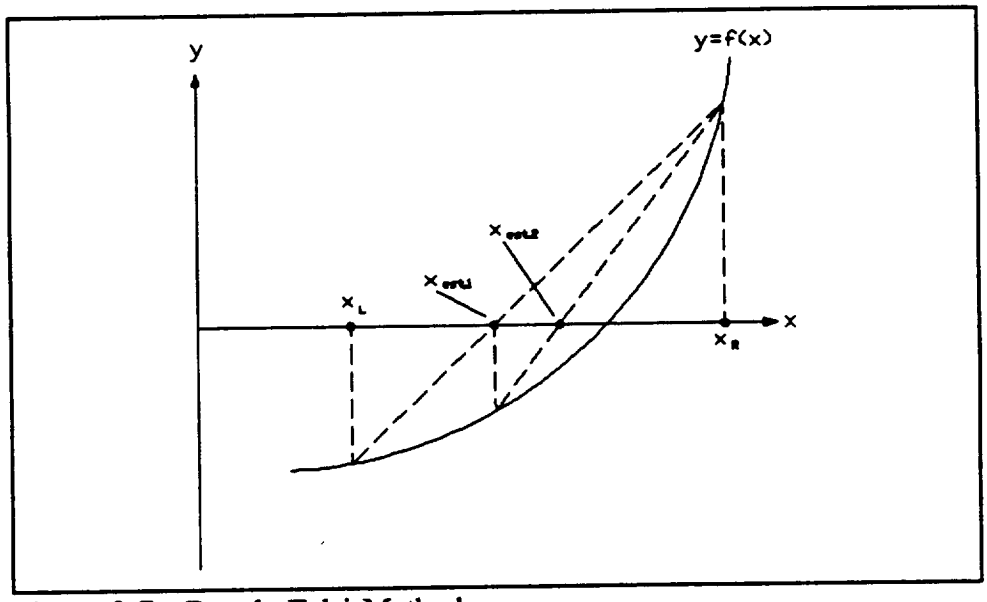


Figure 3.7: Regula-Falsi Method

The subroutine BUILD generates the overall transfer matrix for a given frequency. For each element in the model, BUILD calls subroutine BEAM4 to generate the transfer matrix for that element. When BEAM4 returns with the element transfer matrix, BUILD calls subroutine MUL to update the overall transfer matrix by premultiplying the current overall transfer matrix by the new element transfer matrix. BUILD repeatedly calls BEAM4 and MUL until the overall transfer matrix is complete.

As mentioned, subroutine BEAM4 generates the element transfer matrix for a given frequency. It implements Equation 3.4 given the element parameters passed down from program ZERO in an array. Since the frequency will be complex, all computations to generate the element transfer matrix are carried out using complex calculations. As Appendix A shows, the resulting transfer matrix will have all imaginary parts identically

zero. Before BEAM4 returns the element transfer matrix, it extracts the real part of each term to generate a real element transfer matrix. This real element transfer matrix is passed back to BUILD. By extracting the real elements in BEAM4, all other subroutines can avoid having to do complex calculations. With this review of the program structure, the following section presents a sample run of the program ZERO.

3.3.3 Sample Run of Program ZERO

The sample run includes the screen output and keyboard input as presented to the user. Also included is the input file which contains the element parameters for the model. The first output file (extension: .out) contains a summary of pole and zero location, while the second output file (extension: .dat) contains the pole and zero function values at each subinterval. The file selected has nominal properties with A=0.75 inches and B=0.25 inches.

```
THIS PROGRAM DETERMINES THE LOCATION OF ZEROS
AND POLES FOR A BEAM USING TRANSFER MATRIX THEORY.

WOULD YOU LIKE TO ENTER THE MODEL INFORMATION
MANUALLY OR THRU AN INPUT FILE?
1 FOR MANUAL, 2 FOR FILE, 3 FOR INPUT DESCRIPTION
          3


MODELING PARAMETERS:

   NE-  NUMBER OF ELEMENTS IN THE MODEL
   L-   LENGTH OF ELEMENTS
   MPL- MASS PER UNIT LENTGH OF ELEMENT
   AMI- AREA MOMENT OF INERTIA OF ELEMENT
   E-   YOUNGS MODULUS OF ELEMENT
   DF-  DAMPING FACTOR OF ELEMENT(MUST BE ZERO FOR
        THIS PROGRAM)

TYPE: 1 FOR KEYBOARD INPUT, 2 FOR FILE INPUT
          2


THE INPUT FILE MUST HAVE EXTENSION ".INP"
AND LINES 1-5 ARE RESERVED FOR COMMENT

WHAT IS THE FILE NAME, WITHOUT EXTENSION, WITHIN APOSTROPHES
[.TAPER]TAPB3

WOULD YOU LIKE DEFINITIONS OF THE COMPUTATION PARAMETERS?
1 FOR YES, 2 FOR NO
          1

COMPUTATION PARAMETERS IN ORDER OF INPUT:

 EPSILON= FIRST CONVERGENCE CRITERION. A TRIAL
          ROOT, X, IS ACCEPTED IF ABS[F(X)]<EPS

 NSIG= SECOND CONVERGENCE CRITERION. A TRIAL
       ROOT, X, IS ACCEPTED IF IT AGREES WITH
       THE PREVIOUS TRIAL VALUE TO NSIG SIGNI-
       FICANT DIGITS.

 ITMAX= THE MAXIMUM NUMBER OF ITERATIONS PER
        SUBINTERVAL

 LOW= THE LOWER BOUND OF THE SEARCH INTERVAL

 HIGH= THE UPPER BOUND OF THE SEARCH INTERVAL

 NDIV= THE NUMBER OF SUBDIVISIONS IN THE MAIN
       SEARCH INTERVAL [LOW,HIGH]


INPUT EPS,NSIG,ITMAX
 1.0000000000000000E-20          10          50

INPUT LOW,HIGH,NDIV FOR ZEROS
  1.000000000000000       400.0000000000000             40


                          ZEROS
     SEARCH INTERVAL                  RESULT
     ---------------                  ------

     1.00 TO     10.98                NO ZERO
```

10.98	TO	20.95	ZERO AT	13.644
20.95	TO	30.93	NO ZERO	
30.93	TO	40.90	NO ZERO	
40.90	TO	50.88	NO ZERO	
50.88	TO	60.85	ZERO AT	56.616
60.85	TO	70.83	NO ZERO	
70.83	TO	80.80	NO ZERO	
80.80	TO	90.78	NO ZERO	
90.78	TO	100.75	NO ZERO	
100.75	TO	110.72	NO ZERO	
110.72	TO	120.70	NO ZERO	
120.70	TO	130.68	NO ZERO	
130.68	TO	140.65	ZERO AT	133.016
140.65	TO	150.63	NO ZERO	
150.63	TO	160.60	NO ZERO	
160.60	TO	170.58	NO ZERO	
170.58	TO	180.55	NO ZERO	
180.55	TO	190.53	NO ZERO	
190.53	TO	200.50	NO ZERO	
200.50	TO	210.48	NO ZERO	
210.48	TO	220.45	NO ZERO	
220.45	TO	230.43	NO ZERO	
230.43	TO	240.40	NO ZERO	
240.40	TO	250.38	ZERO AT	243.353
250.38	TO	260.35	NO ZERO	
260.35	TO	270.33	NO ZERO	
270.33	TO	280.30	NO ZERO	
280.30	TO	290.28	NO ZERO	
290.28	TO	300.25	NO ZERO	
300.25	TO	310.23	NO ZERO	
310.23	TO	320.20	NO ZERO	
320.20	TO	330.18	NO ZERO	
330.18	TO	340.15	NO ZERO	
340.15	TO	350.13	NO ZERO	
350.13	TO	360.10	NO ZERO	
360.10	TO	370.08	NO ZERO	
370.08	TO	380.05	NO ZERO	
380.05	TO	390.03	ZERO AT	388.194
390.03	TO	400.00	NO ZERO	

INPUT LOW,HIGH,NDIV FOR POLES
1.000000000000000 300.0000000000000 40

POLES

SEARCH INTERVAL			RESULT	
1.00	TO	8.48	NO POLE	
8.48	TO	15.95	POLE AT	15.890
15.95	TO	23.43	NO POLE	
23.43	TO	30.90	NO POLE	
30.90	TO	38.38	NO POLE	
38.38	TO	45.85	NO POLE	
45.85	TO	53.32	POLE AT	46.034
53.32	TO	60.80	NO POLE	
60.80	TO	68.27	NO POLE	
68.27	TO	75.75	NO POLE	
75.75	TO	83.22	NO POLE	
83.22	TO	90.70	NO POLE	
90.70	TO	98.17	POLE AT	92.904
98.17	TO	105.65	NO POLE	
105.65	TO	113.12	NO POLE	
113.12	TO	120.60	NO POLE	
120.60	TO	128.07	NO POLE	

128.07	TO	135.55	NO POLE	
135.55	TO	143.02	NO POLE	
143.02	TO	150.50	NO POLE	
150.50	TO	157.98	POLE AT	156.637
157.98	TO	165.45	NO POLE	
165.45	TO	172.93	NO POLE	
172.93	TO	180.40	NO POLE	
180.40	TO	187.88	NO POLE	
187.88	TO	195.35	NO POLE	
195.35	TO	202.83	NO POLE	
202.83	TO	210.30	NO POLE	
210.30	TO	217.78	NO POLE	
217.78	TO	225.25	NO POLE	
225.25	TO	232.73	NO POLE	
232.73	TO	240.20	POLE AT	237.385
240.20	TO	247.68	NO POLE	
247.68	TO	255.15	NO POLE	
255.15	TO	262.63	NO POLE	
262.63	TO	270.10	NO POLE	
270.10	TO	277.58	NO POLE	
277.58	TO	285.05	NO POLE	
285.05	TO	292.53	NO POLE	
292.53	TO	300.00	NO POLE	

INPUT FILE: TAPB3

NE				
E L	MPL	I	E	DF
10				
0.222222D+01	0.716250D-01	0.351563D-01	0.100000D+08	0.0D+00
0.444444D+01	0.663194D-01	0.279082D-01	0.100000D+08	0.0D+00
0.444444D+01	0.610139D-01	0.217318D-01	0.100000D+08	0.0D+00
0.444444D+01	0.557083D-01	0.165413D-01	0.100000D+08	0.0D+00
0.444444D+01	0.504028D-01	0.122510D-01	0.100000D+08	0.0D+00
0.444444D+01	0.450972D-01	0.877522D-02	0.100000D+08	0.0D+00
0.444444D+01	0.397917D-01	0.602816D-02	0.100000D+08	0.0D+00
0.444444D+01	0.344861D-01	0.392411D-02	0.100000D+08	0.0D+00
0.444444D+01	0.291806D-01	0.237733D-02	0.100000D+08	0.0D+00
0.222222D+01	0.238750D-01	0.130208D-02	0.100000D+08	0.0D+00

```
CALCULATION PARAMETERS:   [.TAPER]TAPB3

  EPS=        0.100E-19
  NSIG=        10
  ITMAX=       50
  NE=         10

      RESULT
      ------
ZERO AT      13.644
ZERO AT      56.616
ZERO AT     133.016
ZERO AT     243.353
ZERO AT     388.195

POLE AT      15.890
POLE AT      46.034
POLE AT      92.904
POLE AT     156.637
POLE AT     237.385
```

FILE: [.TAPER]TAPB3

ZEROS

XLL	XRR	FXL	FXR
1.00 TO	10.98	0.5192E-01	0.9159E-04
10.98 TO	20.95	ZERO: F=	0.3349E-14
20.95 TO	30.93	-0.3742E-04	-0.1753E-04
30.93 TO	40.90	-0.1753E-04	-0.5520E-05
40.90 TO	50.88	-0.5520E-05	-0.9922E-06
50.88 TO	60.85	ZERO: F=	-0.3918E-18
60.85 TO	70.83	0.3547E-06	0.5707E-06
70.83 TO	80.80	0.5707E-06	0.4579E-06
80.80 TO	90.78	0.4579E-06	0.2963E-06
90.78 TO	100.75	0.2963E-06	0.1667E-06
100.75 TO	110.72	0.1667E-06	0.8095E-07
110.72 TO	120.70	0.8095E-07	0.3046E-07
120.70 TO	130.68	0.3046E-07	0.3853E-08
130.68 TO	140.65	ZERO: F=	0.2324E-16
140.65 TO	150.63	-0.8202E-08	-0.1213E-07
150.63 TO	160.60	-0.1213E-07	-0.1197E-07
160.60 TO	170.58	-0.1197E-07	-0.1007E-07
170.58 TO	180.55	-0.1007E-07	-0.7687E-08
180.55 TO	190.53	-0.7687E-08	-0.5450E-08
190.53 TO	200.50	-0.5450E-08	-0.3601E-08
200.50 TO	210.48	-0.3601E-08	-0.2194E-08
210.48 TO	220.45	-0.2194E-08	-0.1189E-08
220.45 TO	230.43	-0.1189E-08	-0.5134E-09
230.43 TO	240.40	-0.5134E-09	-0.8836E-10
240.40 TO	250.38	ZERO: F=	0.0000E+00
250.38 TO	260.35	0.1563E-09	0.2776E-09
260.35 TO	270.33	0.2776E-09	0.3193E-09
270.33 TO	280.30	0.3193E-09	0.3130E-09
280.30 TO	290.28	0.3130E-09	0.2806E-09
290.28 TO	300.25	0.2806E-09	0.2366E-09
300.25 TO	310.23	0.2366E-09	0.1900E-09
310.23 TO	320.20	0.1900E-09	0.1462E-09
320.20 TO	330.18	0.1462E-09	0.1078E-09
330.18 TO	340.15	0.1078E-09	0.7580E-10
340.15 TO	350.13	0.7580E-10	0.5028E-10
350.13 TO	360.10	0.5028E-10	0.3067E-10
360.10 TO	370.08	0.3067E-10	0.1617E-10
370.08 TO	380.05	0.1617E-10	0.5879E-11
380.05 TO	390.03	ZERO: F=	0.0000E+00
390.03 TO	400.00	-0.1058E-11	-0.5422E-11

POLES

XLL	XRR	FXL	FXR
1.00 TO	8.48	-0.7589E+03	-0.3732E+05
8.48 TO	15.95	POLE: F=	0.2910E-10
15.95 TO	23.43	0.1226E+04	0.3255E+06
23.43 TO	30.90	0.3255E+06	0.9139E+06
30.90 TO	38.38	0.9139E+06	0.1214E+07
38.38 TO	45.85	0.1214E+07	0.5942E+05
45.85 TO	53.32	POLE: F=	0.0000E+00
53.32 TO	60.80	-0.4001E+07	-0.1189E+08
60.80 TO	68.27	-0.1189E+08	-0.2276E+08
68.27 TO	75.75	-0.2276E+08	-0.3262E+08
75.75 TO	83.22	-0.3262E+08	-0.3333E+08
83.22 TO	90.70	-0.3333E+08	-0.1246E+08
90.70 TO	98.17	POLE: F=	0.1526E-04
98.17 TO	105.65	0.4507E+08	0.1533E+09
105.65 TO	113.12	0.1533E+09	0.3189E+09
113.12 TO	120.60	0.3189E+09	0.5329E+09

120.60	TO	128.07	0.5329E+09	0.7610E+09
128.07	TO	135.55	0.7610E+09	0.9336E+09
135.55	TO	143.02	0.9336E+09	0.9399E+09
143.02	TO	150.50	0.9399E+09	0.6267E+09
150.50	TO	157.98	POLE: F=	0.0000E+00
157.98	TO	165.45	-0.1928E+09	-0.1717E+10
165.45	TO	172.93	-0.1717E+10	-0.4118E+10
172.93	TO	180.40	-0.4118E+10	-0.7485E+10
180.40	TO	187.88	-0.7485E+10	-0.1175E+11
187.88	TO	195.35	-0.1175E+11	-0.1662E+11
195.35	TO	202.83	-0.1662E+11	-0.2144E+11
202.83	TO	210.30	-0.2144E+11	-0.2516E+11
210.30	TO	217.78	-0.2516E+11	-0.2622E+11
217.78	TO	225.25	-0.2622E+11	-0.2258E+11
225.25	TO	232.73	-0.2258E+11	-0.1170E+11
232.73	TO	240.20	POLE: F=	0.0000E+00
240.20	TO	247.68	0.9266E+10	0.4325E+11
247.68	TO	255.15	0.4325E+11	0.9285E+11
255.15	TO	262.63	0.9285E+11	0.1598E+12
262.63	TO	270.10	0.1598E+12	0.2443E+12
270.10	TO	277.58	0.2443E+12	0.3440E+12
277.58	TO	285.05	0.3440E+12	0.4531E+12
285.05	TO	292.53	0.4531E+12	0.5617E+12
292.53	TO	300.00	0.5617E+12	0.6540E+12

CHAPTER 4

RESULTS

The results of the zero and pole locations found from program ZERO are presented in this chapter as a collection of studies. Each study investigates a different aspect of the relationship between RHP zero location and structural link design. As mentioned previously, pole location is often of interest to the designer, so this information is presented for each study. Unless otherwise specified, several dimensions remain the same from one study to the next (referred to as nominal dimensions). The overall length of the beams is 40 inches, and the height (which remains constant over length) is 1 inch. The material properties are selected to be those of aluminum: modulus of elasticity, E, is 10E6 psi, and the density is 9.55E-2 lb/in^3.

4.1 Validity of Results

Before examining the relationship between RHP zeros and link design, the validity of the computer algorithm to determine zero/pole location must first be checked. Since analytic solutions exist for the location of poles for a uniform beam, the results from ZERO were compared to the analytic solution to determine the accuracy of the program. The vibrations text by Rao [17] contains the analytic solution for pole location of a pinned-free beam under lateral vibration. The poles were determined from the following

equation:

$$\omega_n = (\beta_n l)^2 \left(\frac{EI}{\rho A l^4} \right)^{\frac{1}{2}} \tag{4.1}$$

For a pinned-free beam, the values of $\beta_n l$ are:

$$\begin{aligned} \beta_1 l &= 3.926602 \\ \beta_2 l &= 7.068583 \\ \beta_3 l &= 10.210176 \end{aligned}$$

For a uniform beam with width=0.5 inches and nominal properties as given above, the pole locations are presented in Table 4.1 along with the results from program ZERO.

Table 4.1: ZERO Program vs. Analytical Solution

Pole	Program ZERO	Analytical Solution
1	14.23	14.23
2	46.12	46.12
3	96.23	96.23

The results generated from program ZERO show excellent correspondence to the analytic values. However analytic calculation of zeros is not as simple of a task since the boundary conditions are no longer homogeneous, and texts lack tabulated results. The

same method was used to calculate poles and zeros, only a different function was used.

It must be noted that the results presented in this chapter will not include the two poles lying at the origin. These poles are a result of the rigid body mode of the system. Keep in mind the location of the poles will be presented in tabular form as a real number, but they actually are located on the s-plane in complex conjugate pairs along the imaginary axis. The zeros are also presented in tabular form as real numbers, and they lie on the real axis as reflected pairs about the imaginary axis. This means for every RHP zero found, there was a corresponding LHP zero equal in magnitude but opposite in sign. The symmetry of the s-plane results from ignoring the damping of the structure in the Euler-Bernoulli model and was confirmed by Spector and Flashner [18].

4.2 Effects of Discretization

When modeling a continuous system with a discrete model, one should check to make sure the discretization of the model does not affect the results. This was easily confirmed by studying a uniform beam. Using transfer matrices, a uniform beam can be modeled with one element or several elements. Table 4.2 shows the results of program ZERO for a uniform beam modeled with 1 element and 20 elements. The beam had nominal properties with W=0.5 inches. Notice the results were identical for both the poles and zeros as one would expect. For a tapered beam, the number of elements will be more critical because increasing the elements will decrease the discontinuities at each element interface. For nonuniform designs, the poles and zeros should converge to the actual values as the number of elements increases. This will be confirmed later in the chapter.

Table 4.2: Effects of Discretization

Zero Pole	NE=1	NE=20
1	10.33 14.23	10.33 14.23
2	55.80 46.12	55.80 46.12
3	137.8 96.23	137.8 96.23
4	256.2 164.6	256.2 164.6
5	411.1 251.1	411.1 251.1

4.3 Modeling of a Tapered Beam

Another point to consider in the computer implementation of the RHP zeros problem was how well does the model represent the actual system. Although the model was limited to uniform elements, there were any number of combinations one can find to represent the system. This study examined two different methods for modeling a linearly tapered beam. As shown in Figure 4.1 the link was tapered along the length in the width dimension while the height was held constant. The taper was described by two dimensions: the width at the base, A, and the width at the tip, B. The degree of taper, R=A/B, was used to compare different designs.

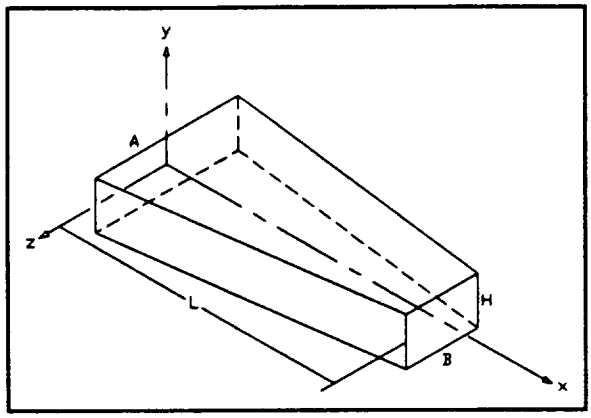


Figure 4.1: Tapered Link Diagram

Using Method 1 to model the tapered link, the beam was divided into elements of equal length. For a three element model with length L, each element will have length L/3. The height of each element was the same, while the width of each element changed linearly as a function of x. Figure 4.2 presents modeling Method 1.

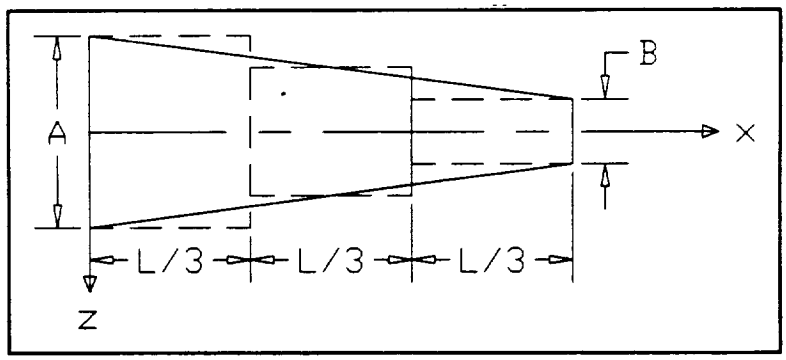


Figure 4.2: Modeling Method 1

Using Method 2 to model the tapered link, the beam was divided into elements so the first and last element have length one-half of the intermediate elements. For a three element model with length L, the first and last elements will have length L/4 and the middle element will have length L/2. Again the height of each element was the

same, while the width of each element changed linearly as a function of x. Figure 4.3 presents modeling Method 2.

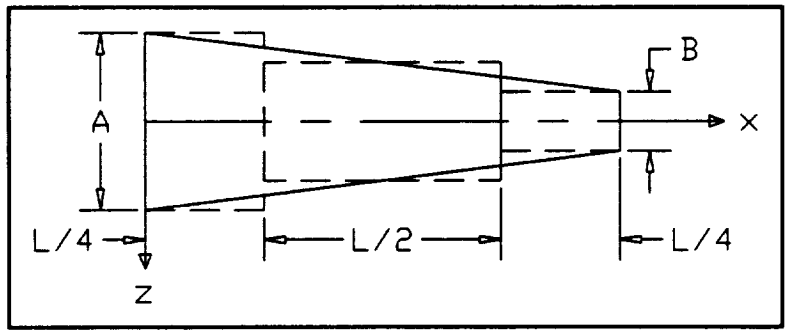


Figure 4.3: Modeling Method 2

Figures 4.2 and 4.3 illustrate the main difference between the two modeling methods. Method 2 compensated the elements at each end for meeting the specified end widths A and B. In both methods the width of intermediate elements was determined by the width of the tapered beam at the midpoint of each element. Since the end elements meet the specified A and B, the tapered link will not pass through the midpoint of these two elements. Method 2 compensates for this exception by making the end element lengths one half the length of the other elements.

To compare these two different modeling methods for a linearly tapered beam, a beam with nominal dimensions and A=0.75 inches and B=0.25 inches was studied. This corresponds to R=3. The number of elements was increased with each method until the zeros and poles converged. Table 4.3 presents the results from Method 1 where all elements were of equal length, and Table 4.4 presents the results from Method 2 where the end elements were half the length of all other elements. Although only two

methods are considered in this research, there are many different ways to discretize a nonuniform link.

The two methods were evaluated based on an error function. When the tapered beam was modeled with 80 elements, both methods converged to nearly identical values for the poles and zeros. These values, when NE=80, were taken to be the "correct' values and other cases were compared to this case. The error, e, was defined for the zeros as:

$$e = \left| \frac{z_{80,i} - z_{NE,i}}{z_{80,i}} \right| \tag{4.2}$$

where i refers to the i^{th} zero

A similar definition was used for the poles. The value of e at the top of each column represents the maximum of all individual errors in each column. As the tables show, Method 2 provided better results for the same number of elements. In each table, one column was shaded to distinguish it as the number of elements needed to get the error under 1%. For Method 2, this column corresponded to NE=10 as opposed to NE=20 for Method 1. Thus, compensating the end elements did provide a better model of a linearly tapered beam, and this method was used in the following studies unless specified otherwise. Both tables also show that the convergence of poles and zeros as the number of the number of elements was increased.

Table 4.3: Results From Method 1

Zero Pole	NE=3 ($e \leq 20\%$)	NE=5 ($e \leq 8.6\%$)	NE=10 ($e \leq 1.9\%$)	NE=20 ($e \leq 0.3\%$)	NE=40 ($e \leq 0.1\%$)	NE=80
1	13.91 13.64	13.99 15.95	13.73 16.05	13.69 15.96	13.68 15.92	13.68 15.91
2	45.57 38.08	55.99 43.24	57.28 46.19	56.91 46.21	56.84 46.14	56.83 46.11
3	121.2 88.16	122.0 85.31	134.2 92.52	133.8 93.20	133.6 93.13	133.5 93.09
4	210.8 137.5	223.2 147.5	242.9 154.9	244.7 157.0	244.2 157.0	244.1 157.0
5	357.8 219.5	383.8 234.7	382.1 233.3	389.4 237.7	388.9 237.9	388.7 237.8

Table 4.4: Results From Method 2

Zero Pole	NE=3 ($e \leq 16\%$)	NE=5 ($e \leq 4.0\%$)	NE=10 ($e \leq 0.4\%$)	NE=20 ($e \leq 0.1\%$)	NE=40 ($e \leq 0.0\%$)	NE=80
1	13.09 15.57	13.49 15.82	13.64 15.89	13.67 15.90	13.68 15.91	13.68 15.91
2	53.77 38.66	56.12 45.82	56.62 46.03	56.78 46.09	56.82 46.10	56.83 46.10
3	120.4 85.88	135.1 93.17	133.0 92.90	133.4 93.03	133.5 93.06	133.5 93.06
4	233.6 154.4	234.7 148.3	243.4 156.6	243.9 156.9	244.1 156.9	244.1 156.9
5	360.6 220.3	384.6 231.1	388.2 237.4	388.3 237.7	388.6 237.7	388.6 237.8

4.4 Linear Taper Designs

When comparing different link designs to evaluate pole/zero location as a function of link shape, it was necessary to keep some parameter constant to aid in the evaluation. For a single-link manipulator rotating in the horizontal plane, the link's mass moment of inertia about its axis of rotation, I_y, was of importance. This parameter directly affected the dynamic equations of motion and was an important design parameter in terms of motor selection. In the following studies, several link designs were evaluated for a given value of I_y. Appendix C shows the derivation of a tapered link's moment of inertia about its axis of rotation in terms of the links parameters: L, A, B, H, and ρ. The final result was:

$$I_y = \frac{\rho H}{48}(A^3 + A^2B + AB^2 + B^3 + 4AL^2 + 12BL^2) \qquad (4.3)$$

For a given tapered link design, one can use Equation 4.3 to determine I_y. Knowing I_y, one can change the value of A and solve Equation 4.3 for B. Since the equation was cubic in B, the commercial package *Mathematica* was used to solve for B. Following this method, a group of tapered link designs were generated all with the same I_y.

The first study investigated several tapered link designs with nominal dimensions and all designs having I_y=764.05 in-lb-sec^2. Table 4.5 presents the raw data for each of these designs. Even with I_y held constant, it was still difficult to interpret the data. To aid in developing a relationship between zero location and link shape, the zeros were normalized with respect to the first pole for each design. The first pole is an important

parameter in control system design, and normalizing the zeros with respect to the first pole aided in the interpretation of the results. Table 4.6 presents the normalized data for those designs with I_y=764.05 in-lb-sec^2. The second study presents data for several link designs with nominal dimensions and I_y=1528.1 in-lb-sec^2. Table 4.7 shows the raw data for these link designs and Table 4.8 shows the normalized data for these designs. Figures 4.4 and 4.5 show pole/zero maps for selected values of R for I_y=764.05 and I_y=1528.1 respectively.

Several patterns were evident by examining the raw data. First as a general rule, both the poles and zeros increased (moved away from the origin) as the taper on the beam increased. Increasing the taper effectively moved more of the link mass closer to the base. Increasing the value of the poles is often desirable to push them out of the system bandwidth and increase system response time. The ordering of poles and zeros was the second pattern recognized. In a minimum phase system, the poles and zeros will both lie on the imaginary axis in complex conjugate pairs and in an alternating order. This means, along the imaginary axis, the poles and zero are found in the order p_1, z_1, p_2,z_2, etc. or vice versa. Previous research [18] has found this alternating order of poles and zeros does not hold for nonminimum phase systems. Referring to Table 4.5, notice the order of the magnitude of poles and zeros was: z_1,p_1,p_2,z_2,p_3,z_3,p_4,p_5,z_4.... p_2 jumped in front of z_2, and the same occurred for p_5. This reordering of poles and zeros can be critical as accurate knowledge of the pole/zero order is important for control system design.

Table 4.5: Tapered Beams With I_y=764.05

Zero Pole	A=.375 B=.375	A=.4 B=.367	A=.5 B=.333	A=.6 B=.3	A=.7 B=.267	A=.8 B=.233	A=.9 B=.2	A=1 B=.167
1	7.745 10.68	8.153 11.04	9.762 12.46	11.34 13.84	12.90 15.21	14.44 16.60	15.98 18.03	17.50 19.52
2	41.85 34.59	43.15 35.48	47.38 38.80	51.37 41.87	55.05 44.73	58.45 47.41	61.60 49.94	64.51 52.36
3	103.4 72.18	105.9 73.88	115.0 80.17	123.1 85.75	130.2 90.75	136.4 95.19	141.7 99.14	146.2 102.6
4	192.2 123.4	196.6 126.2	212.7 136.5	226.6 145.5	238.6 153.4	248.7 160.1	257.1 165.9	263.6 170.6
5	308.4 188.3	315.3 192.6	340.5 208.0	362.0 221.2	380.3 232.6	395.5 242.3	407.8 250.3	416.9 256.5

Table 4.6: Normalized Data For I_y=764.05

Zero	R=1.00	R=1.09	R=1.50	R=2.00	R=2.62	R=3.43	R=4.50	R=5.99
1	0.7252	0.7385	0.7835	0.8194	0.8481	0.8699	0.8863	0.8965
2	3.919	3.909	3.803	3.712	3.619	3.521	3.417	3.305
3	9.682	9.592	9.230	8.895	8.560	8.217	7.859	7.490
4	18.00	17.81	17.07	16.37	15.69	14.98	14.26	13.50
5	28.88	28.56	27.33	26.16	25.00	23.83	22.62	21.36

Table 4.7: Tapered Beams With $I_y = 1528.1$

Zero Pole	A=.75 B=.75	A=.8 B=.733	A=.9 B=.7	A=1.0 B=.667	A=1.1 B=.633	A=1.2 B=.600
1	15.49 21.35	16.31 22.08	17.92 23.51	19.52 24.92	21.11 26.30	22.68 27.68
2	83.71 69.16	86.03 70.95	90.50 74.35	94.76 77.60	98.83 80.73	102.7 83.74
3	206.7 144.4	211.7 147.7	221.2 154.2	230.1 160.3	238.4 166.1	246.2 171.5
4	384.4 246.8	393.2 252.5	409.9 263.1	425.4 273.1	439.9 282.4	453.2 291.0
5	616.7 376.7	630.6 385.1	656.8 401.1	681.0 415.9	703.3 429.6	724.0 442.4

Table 4.8: Normalized Data For $I_y = 1528.1$

Zero	R=1.00	R=1.09	R=1.29	R=1.50	R=1.74	R=2.00
1	0.7256	0.7385	0.7623	0.7836	0.8026	0.8195
2	3.921	3.896	3.849	3.803	3.757	3.712
3	9.682	9.588	9.407	9.233	9.603	8.894
4	18.00	17.81	17.44	17.08	16.72	16.38
5	28.89	28.56	27.93	27.33	26.74	26.16

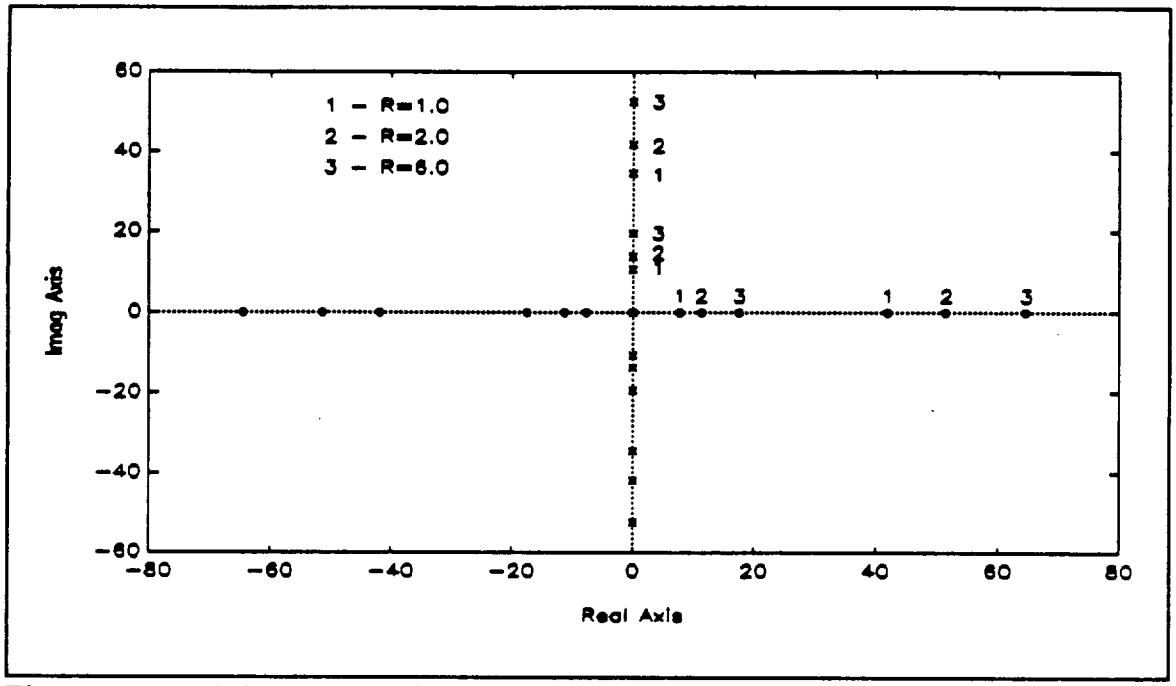


Figure 4.4: Pole/Zero Map of Selected Designs For I_y=764.05

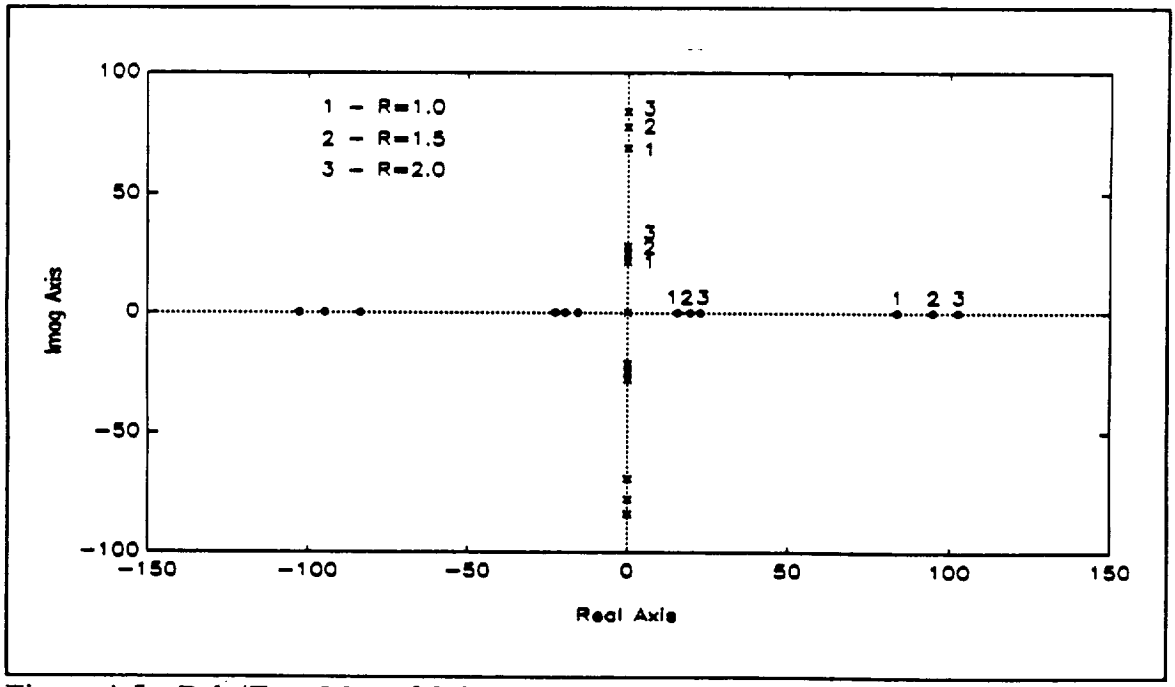


Figure 4.5: Pole/Zero Map of Selected Designs For I_y=1528.1

4.4.1 Designs With Constant I_y

Important information was learned from examining the relationship between the taper ratio, R, and the values of the normalized zeros. The first normalized zero was of most importance, and Figure 4.6 shows this relationship for the data from the first study fitted with a third order polynomial. Figure 4.7 presents the same data for the second study. The significance was not the actual relationship, but the fact that the two relationships were nearly the same for both cases. Figure 4.8 better illustrates this point showing both polynomial fits on the same graph. Even though the coefficients were different for each polynomial fit, the curves were nearly identical.

This illustrates an important relationship in the design of tapered links. For a given ratio R, the normalized zero will always remain the same. The designer can choose the location of the first pole and zero, determine the normalized zero, and then using Figure 4.8 find the appropriate taper ratio R. Of course there are constraints on this process. A ratio less than one corresponds to a taper with B greater than A, which is usually undesirable. At the other end, R is limited by the value of H. If A is larger than the value of H, the link will be wider at the base than it is tall, and the assumption that the link is stiff in the vertical plane will no longer be valid. Although the designer can choose the pole/zero relationship, the values of normalized zeros are limited to approximately 0.72-0.82 (according to Figure 4.8).

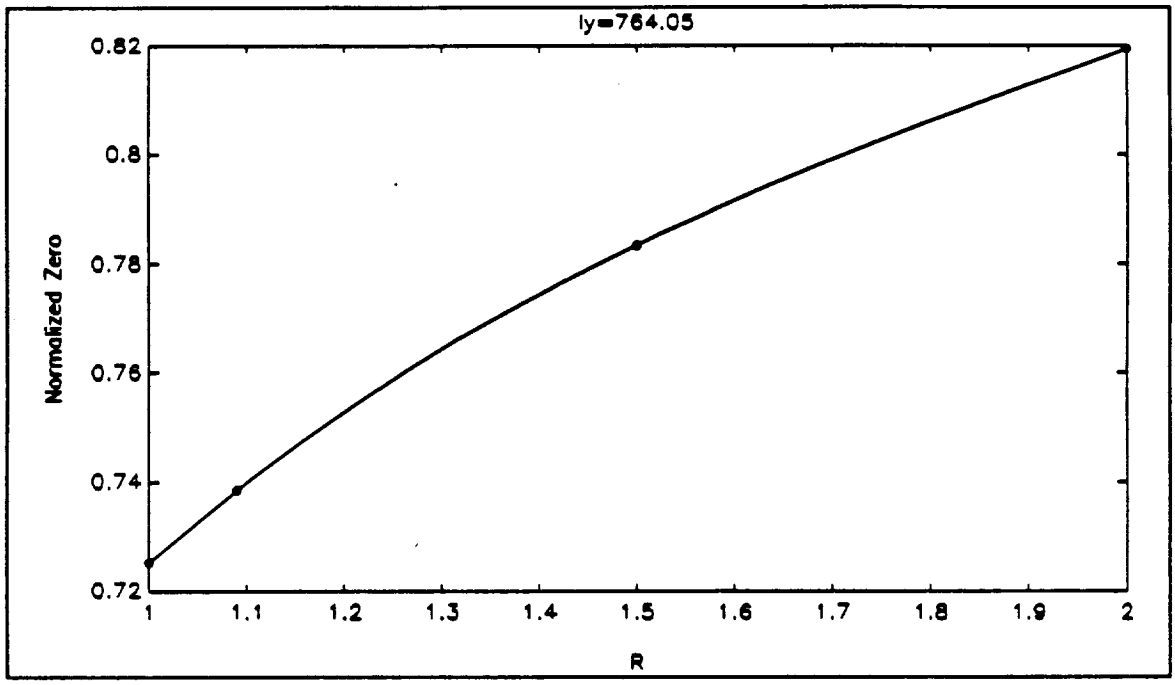


Figure 4.6: First Normalized Zero vs. R For I_y=764.05
$NZ=0.0343R^3-0.1993R^2+0.4518R+0.4384$

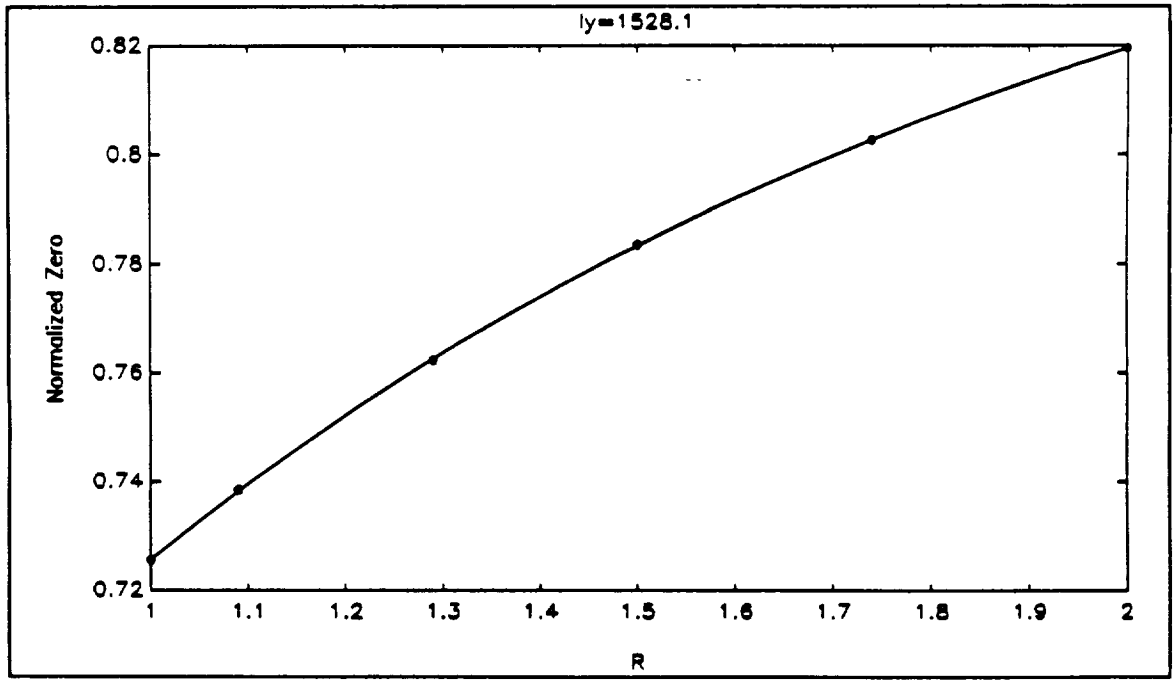


Figure 4.7: First Normalized Zero vs. R For I_y=1528.1
$NZ=0.0168R^3-0.1191R^2+0.3331R+0.4948$

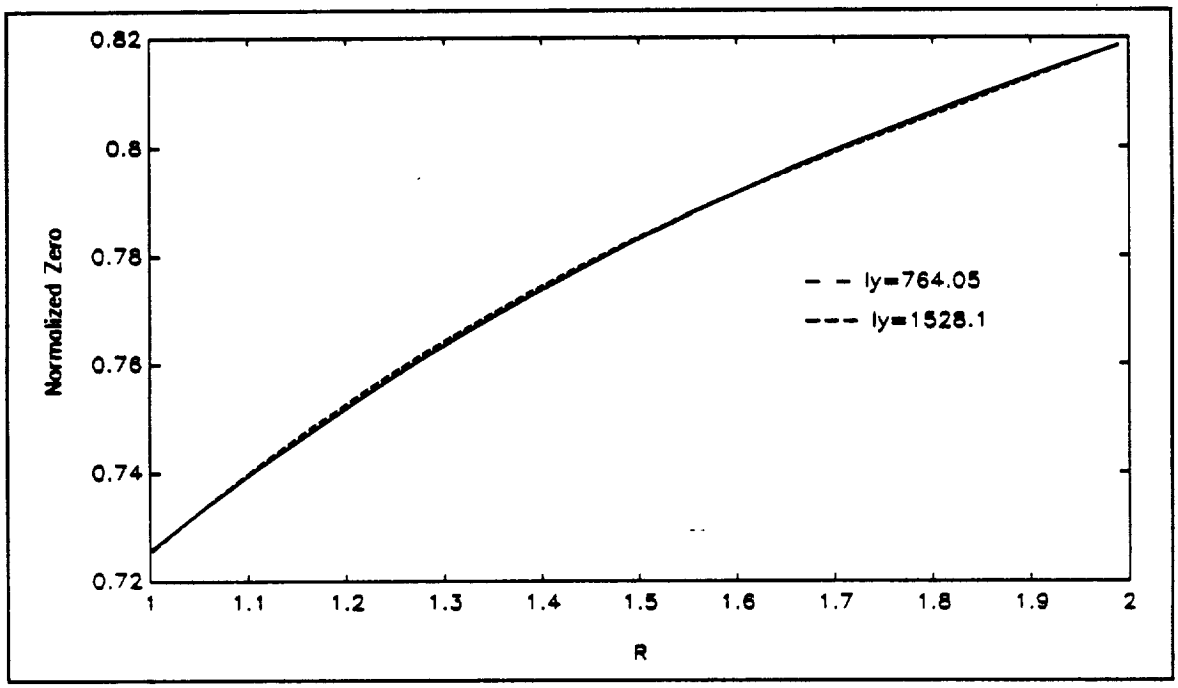


Figure 4.8: Comparison of Polynomial Curve Fits

A simple verification of the above relationship is the uniform beam which has no taper. According to the stated relationship, the normalized first zero should be the same for all uniform beams. Table 4.9 presents the results for several uniform beam designs. All cases had nominal dimensions. The normalized zero in all cases was 0.726 which confirmed the normalized zero will not change as long as R is constant.

Table 4.9: ZERO Results For Uniform Beam Designs

Zero Pole	W=0.25"	W=0.5"	W=0.75"
1	5.163 7.116	10.33 14.23	15.49 21.35
NZ	0.726	0.726	0.726
2	27.90 23.06	55.80 46.12	83.71 69.19
3	68.90 48.12	137.8 96.23	206.7 144.3
4	128.1 82.28	256.2 164.6	384.4 246.8
5	205.6 125.6	411.1 251.1	616.7 376.7

4.4.2 Designs With Constant Poles and Zeros

The previous study demonstrated how the designer can choose the pole/zero relationship and then determine the appropriate taper design from the ZERO results. This study presents the designer with another freedom. Once the taper is chosen, the designer can change the link to independently adjust the value of I_y. Table 4.10 presents

the results of a study performed on designs with L=40 inches, and all designs have the same taper. The height of the link was changed to adjust the value of I_y.

Table 4.10: Variable Height Designs

Zero Pole	H=1.0"	H=1.5"	H=2.0"
1	11.34 13.84	11.34 13.84	11.34 13.84
2	51.37 41.87	51.37 41.87	51.37 41.87
3	123.1 85.75	123.1 85.75	123.1 85.75
4	226.6 145.5	226.6 145.5	226.6 145.5
5	362.0 221.2	362.0 221.2	362.0 221.2
I_y	764.05	1146.1	1528.1

One should notice that the pole and zero locations of all designs in Table 4.10 were the same, yet the value of I_y changed with adjustments in link height. Since the adjustment of H is out of the plane of motion, it had no effect on the location of poles and zeros. Combining this with the results from the previous study, the designer can effectively choose the location of poles and zeros and independently adjust the links moment of inertia about its axis of rotation to meet the needs of the particular system.

CHAPTER V

CONCLUSIONS

5.1 Summary and Contributions

Program ZERO was developed as a tool to locate the poles and zeros of a single-link manipulator modeled as a pinned-free Euler-Bernoulli beam. The program used transfer matrix theory to allow for variable cross-sections granting the designer new freedom in analysis of nonuniform link designs. The results were shown to be very accurate when system pole location was compared to analytic solutions for uniform beams. Several results from previous studies were confirmed with this research.

First, the reordering of poles and zeros was confirmed for nonminimum phase systems. Accurate knowledge of pole/zero order is critical for proper control system design. In conjunction with this, Tables 4.3 and 4.4 show that even for very few elements in the model, the program still predicts the proper order of poles and zeros.

Second, the studies presented suggested the nonminimum phase characteristics could not be eliminated by changing the structural design of the link. The system will be nonminimum phase above a finite frequency dictated by the location of the first nonminimum phase zero. It may be possible that this frequency is out of the operating range and not of concern to the designer.

The major contributions of this research are the development of the ZERO program to determine zero and pole location for a single-link nonuniform flexible manipulator, and formulation of a design procedure to place the first pole and zero and independently change the value of the link's moment of inertia about its axis of rotation to meet the needs of the system.

Program ZERO was set up specifically for pinned-free boundary conditions of the model and determines pole and zero location based on a frequency range entered by the user. Linearly tapered beams were studied in this research, but any type of nonuniform beam can be analyzed by program ZERO. Slight modifications would also allow for different boundary conditions.

The design procedure for tapered beams allows the designer to choose the first pole and zero subject to certain physical constraints. These physical constraints only allow for approximately 25% variation in R according to Table 4.6. This zero to pole ratio defines a particular taper ratio according to the collected data. Keeping the ratio the same, the size of the taper can be changed to get the proper magnitude of the pole and zero. With the pole and zero placed, the height of the beam can be changed to adjust the link's moment of inertia about its axis of rotation. This procedure can be used to design tapered links to meet the particular requirements of the system.

5.2 Future Work

Program ZERO was designed to model a single-link manipulator modeled with pinned-free boundary conditions. This is a simplified model, but it was necessary to

show transfer matrices yield good results for this case before progressing to more complicated problems. Now that transfer matrices have proven useful to solve for zero location, future work exists to extend the results of this research.

First, the program could be modified so the user could input the desired boundary conditions which best represent the system. This could include hub inertia or end-point mass. Second, the program could be extended to multi-link designs to predict pole and zero location for different configurations. Transfer matrices have been derived for rotary joints and many other elements. The DSAP package developed by Book, et. al. [A] handles multi-link models and would be a good reference. Finally, the results for tapered link designs could be applied to the inverse dynamic algorithm developed by Kwon and Book [N]. This method requires mode shapes for the assumed modes and uses pinned-pinned boundary conditions. To help with this transition, Appendix D gives the natural frequencies for some tapered designs modeled with pinned-pinned boundary conditions. These results can be used to generate the modes shapes necessary for the inverse dynamic algorithm.

APPENDIX A

PROOF OF REAL TRANSFER MATRIX

The subroutine ZFALSE can only find the roots of a real equation. The following proof shows a purely complex frequency will result in a transfer matrix with only real elements. The damping factor is assumed to be zero. The transfer matrix for a Bernoulli-Euler beam has the form:

$$TM = \begin{bmatrix} C_0 & lC_1 & aC_2 & alC_3 \\ \frac{\beta^4 C_3}{l} & C_0 & \frac{aC_1}{l} & aC_2 \\ \frac{\beta^4 C_2}{a} & \frac{\beta^4 lC_3}{a} & C_0 & lC_1 \\ \frac{\beta^4 C_1}{al} & \frac{\beta^4 C_2}{a} & \frac{\beta^4 C_3}{l} & C_0 \end{bmatrix} \tag{A.1}$$

where,

$$C_0 = \frac{1}{2}(\cosh\beta + \cos\beta) \tag{A.2}$$

$$C_1 = \frac{1}{2\beta}(\sinh\beta + \sin\beta) \tag{A.3}$$

$$C_2 = \frac{1}{2\beta^2}(\cosh\beta - \cos\beta) \tag{A.4}$$

$$C_3 = \frac{1}{2\beta^3}(\sinh\beta - \sin\beta) \tag{A.5}$$

and

$$\beta^4 = \frac{\omega^2 l^4 \mu}{EI} \quad (A.6) \qquad a = \frac{l^2}{EI} \quad (A.7)$$

The following symbols are defined as

μ	=	mass density per unit length
ω	=	frequency in radians/second
E	=	Young's modulus
I	=	Cross sectional area moment of inertia
l	=	length of the beam

As explained in Chapter 3, to search for real positive values of s, the real part of ω should be zero and the imaginary part of ω should be negative. Using the notation [x,y] to denote a complex number with real part x and imaginary part y, a purely imaginary frequency, ω, can be defined as:

$$\omega = [0, -p] \qquad where \quad 0 \le p \le \infty \quad (A.8)$$

To simplify the proof let:

$$a = 1 \quad (A.9) \qquad \frac{l^4 \mu}{EI} = 1 \quad (A.10)$$

Now to find β take the square root of β^4 two times. For a detailed discussion of finding the square root of a complex number see Churchill and Brown [7]. The principle roots are:

$$\beta^4 = [0,-p]^2 = [-p^2,0] \tag{A.11}$$

$$\beta^2 = \sqrt{\beta^4} = [0,-p] \tag{A.12}$$

$$\beta = \sqrt{\beta^2} = [0.707\sqrt{p},-0.707\sqrt{p}] = [b,-b] \tag{A.13}$$

$$\textit{where } b = 0.707\sqrt{p}$$

Expand cosh β in terms of $\beta=[b,-b]$ to get:

$$\begin{aligned}\cosh\beta &= \tfrac{1}{2}(e^{\beta} + e^{-\beta}) \\ &= \tfrac{1}{2}(e^{[b,-b]} + e^{-[b,-b]}) \\ &= \tfrac{1}{2}(e^{b}e^{-ib} + e^{-b}e^{ib}) \\ &= \tfrac{1}{2}(e^{b}[\cos(-b),\sin(-b)] + e^{-b}[\cos b,\sin b])\end{aligned} \tag{A.14}$$

For any angle b:

$$\cos b = \cos(-b) \quad (A.15) \qquad\qquad \sin b = -\sin(-b) \tag{A.16}$$

Using these relations in Equation (A.14), cosh β simplifies to:

$$\cosh\beta = \tfrac{1}{2}[\,(e^{b} + e^{-b})\cos b\ ,\ -(e^{b} - e^{-b})\sin b\,] \tag{A.17}$$

Similarly cos β reduces to:

$$\begin{aligned}\cos\beta &= \tfrac{1}{2}(e^{i\beta} + e^{-i\beta}) \\ &= \tfrac{1}{2}[\,(e^{b} + e^{-b})\cos b\ ,\ (e^{b} - e^{-b})\sin b\,]\end{aligned} \tag{A.18}$$

Similarly sinh β reduces to:

$$\begin{aligned}\sinh\beta &= \tfrac{1}{2}(e^{\beta} - e^{-\beta}) \\ &= \tfrac{1}{2}\left[\ (e^{b} - e^{-b})\cos b\ ,\ -(e^{b} + e^{-b})\sin b\ \right]\end{aligned} \tag{A.19}$$

Similarly sin β reduces to:

$$\begin{aligned}\sin\beta &= \tfrac{1}{2i}(e^{i\beta} - e^{-i\beta}) \\ &= \tfrac{1}{2}\left[\ (e^{b} + e^{-b})\sin b\ ,\ -(e^{b} - e^{-b})\cos b\ \right]\end{aligned} \tag{A.20}$$

With these expansions, one can substitute into the expressions for C_0, C_1, C_2, and C_3 to show the imaginary parts of these functions are zero. Substitute (A.17) and (A.18) into (A.2) and solve for C_0 to get:

$$\begin{aligned}C_0 &= \tfrac{1}{2}\left\{\tfrac{1}{2}\left[\ (e^{b} + e^{-b})\cos b\ ,\ -(e^{b} - e^{-b})\sin b\ \right] + \tfrac{1}{2}\left[\ (e^{b} + e^{-b})\cos b\ ,\ (e^{b} - e^{-b})\sin b\ \right]\right\} \\ &= \tfrac{1}{2}\left[\ (e^{b} + e^{-b})\cos b\ ,\ 0\ \right]\end{aligned} \tag{A.21}$$

Substitute (A.17) and (A.18) into (A.3) and solve for C_1 to get:

$$\begin{aligned}C_1 &= \frac{\tfrac{1}{2}\left\{\tfrac{1}{2}\left[(e^{b} - e^{-b})\cos b, -(e^{b} + e^{-b})\sin b\right] + \tfrac{1}{2}\left[(e^{b} + e^{-b})\sin b, -(e^{b} - e^{-b})\cos b\right]\right\}}{[b, -b]} \\ &= \frac{\tfrac{1}{4}\left[\{(e^{b} - e^{-b})\cos b + (e^{b} + e^{-b})\sin b\}\ ,\ \{-(e^{b} - e^{-b})\cos b - (e^{b} + e^{-b})\sin b\}\right]}{[b, -b]}\end{aligned} \tag{A.22}$$

Let,

$$b_1 = (e^b - e^{-b})\cos b + (e^b + e^{-b})\sin b \tag{A.23}$$

Now, Equation (A.22) is simply

$$C_1 = \frac{1}{4}\frac{[b_1,-b_1]}{[b,-b]} = \frac{1}{4}\left[\frac{bb_1+bb_1}{2b^2},\frac{-bb_1+bb_1}{2b^2}\right] \tag{A.24}$$

Substituting back in for b_1, C_1 reduces to:

$$C_1 = \frac{1}{4b}[\ (e^b - e^{-b})\cos b + (e^b + e^{-b})\sin b\ ,\ 0\ ] \tag{A.25}$$

Substitute Equations (A.19) and (A.20) into Equation (A.4) and solve for C_2 to get:

$$C_2 = \frac{\frac{1}{2}\left\{\frac{1}{2}[(e^b + e^{-b})\cos b, -(e^b - e^{-b})\sin b] - \frac{1}{2}[(e^b + e^{-b})\cos b, (e^b - e^{-b})\sin b]\right\}}{[0,2b^2]}$$
$$= \frac{1}{4b^2}[\ -(e^b - e^{-b})\sin b\ ,\ 0\ ] \tag{A.26}$$

Substitute Equations (A.19) and (A.20) into Equation (A.5) and solve for C_3 to get:

$$C_3 = \frac{\frac{1}{2}\left\{\frac{1}{2}[(e^b - e^{-b})\cos b, -(e^b + e^{-b})\sin b] - \frac{1}{2}[(e^b + e^{-b})\sin b, -(e^b - e^{-b})\cos b]\right\}}{[2b^3,2b^3]}$$
$$= \frac{\frac{1}{4}[\{(e^b - e^{-b})\cos b - (e^b + e^{-b})\sin b\},\{(e^b - e^{-b})\cos b - (e^b + e^{-b})\sin b\}]}{[2b^3,2b^3]} \tag{A.27}$$

Let

$$b_2 = (e^b - e^{-b})\cos b - (e^b + e^{-b})\sin b \tag{A.28}$$

Now, Equation (A.30) simplifies to:

$$C_3 = \frac{\frac{1}{4}[b_2, b_2]}{[2b^3, 2b^3]} = \frac{1}{4}[\frac{b_2}{2b^3}, 0] \qquad \text{(A.29)}$$

Substituting back for b_3, C_3 becomes:

$$C_3 = \frac{1}{8b^3}[\,(e^b - e^{-b})\cos b - (e^b + e^{-b})\sin b\,,\ 0\,] \qquad \text{(A.30)}$$

All elements of the matrix in Equation (A.1) are shown to be real elements. Equations (A.21), (A.25), (A.26), and (A.30) show the imaginary parts of C_0, C_1, C_2, C_3 are zero, respectively. Equation (A.11) shows the imaginary part of β^4 is zero. As long as the damping factor is zero, Equation (A.7) will always be real.

APPENDIX B

DERIVATION OF ZEROS FUNCTION

The zeros of a system are defined as the frequencies that result in zero system output for an arbitrary system input. To determine the system zeros, one must know a) the system input, b) the system output, and c) the relationship between the input and the output. This can be expressed in an equation of the form:

$$INPUT = \begin{pmatrix} TRANSFER \\ FUNCTION \end{pmatrix} * OUTPUT \tag{B.1}$$

For an arbitrary input to the system, the only way to guarantee zero output is for the transfer function to be zero at the given frequency.

As presented in Chapter 2, transfer matrix theory is very similar to Finite Element Analysis in that the beam is modeled as a system of contiguous elements each having its own transfer matrix. These element transfer matrices can be multiplied together to generate the overall transfer matrix for the beam. Now the equation for the beam can be expressed as:

$$\begin{bmatrix} -w \\ \psi \\ M \\ V \end{bmatrix}_{x=L} = \begin{bmatrix} B_{11} & \cdots & B_{14} \\ \vdots & \ddots & \vdots \\ B_{41} & \cdots & B_{44} \end{bmatrix} \begin{bmatrix} -w \\ \psi \\ M \\ V \end{bmatrix}_{x=0} \tag{B.2}$$

Figure A.1 shows the system under consideration, a flexible beam. The input to the system is the torque applied at the base. The output of the system is the position of the end-point of the beam. The boundary conditions corresponding to this system are:

At x=L: $M=0,\ V=0$ (free)

At x=0: $W=0,\ M=0$ (pinned)

Substituting the boundary conditions and the system input (w_L) and output (τ) into Equation (B.2), the equation for the beam becomes:

$$\begin{bmatrix} -w \\ \psi \\ 0 \\ 0 \end{bmatrix}_{x=L} = \begin{bmatrix} B_{11} & \cdots & B_{14} \\ \vdots & \ddots & \vdots \\ B_{41} & \cdots & B_{44} \end{bmatrix} \begin{bmatrix} 0 \\ \psi \\ \tau \\ V \end{bmatrix}_{x=0} \tag{B.3}$$

Equation (B.3) can be expanded to find the relationship between input and output. The four equations are:

$$-w_L = B_{12}\psi_0 + B_{13}\tau + B_{14}V_0 \tag{B.4}$$

$$\psi_L = B_{22}\psi_0 + B_{23}\tau + B_{24}V_0 \tag{B.5}$$

$$0 = B_{32}\psi_0 + B_{33}\tau + B_{34}V_0 \tag{B.6}$$

$$0 = B_{42}\psi_0 + B_{43}\tau + B_{44}V_0 \tag{B.7}$$

Since ψ_L is not of interest, Equations (B.4), (B.6), and (B.7) can be solved for the relationship between w_L and τ:

$$w_L = -\left[\frac{B_{12}B_{44}B_{33} - B_{12}B_{34}B_{43} + B_{13}B_{34}B_{42} - B_{13}B_{44}B_{32} + B_{14}B_{43}B_{32} - B_{14}B_{33}B_{42}}{B_{34}B_{42} - B_{44}B_{32}}\right]\tau \tag{B.8}$$

The above equation describes the relationship between the input and output for the system under consideration. When the term inside the brackets goes to zero, the system is said to have a zero at the corresponding frequency.

APPENDIX C

INERTIA OF A TAPERED LINK ABOUT ITS BASE

The inertia of a tapered link about its axis of rotation is an important parameter in controls since it directly effects the equations of motion. Figure C.1 shows a sketch of a tapered link with the appropriately defined coordinate axes. For a link rotating about the y axis in the horizontal plane (xz), the mass moment of inertia of interest is I_y.

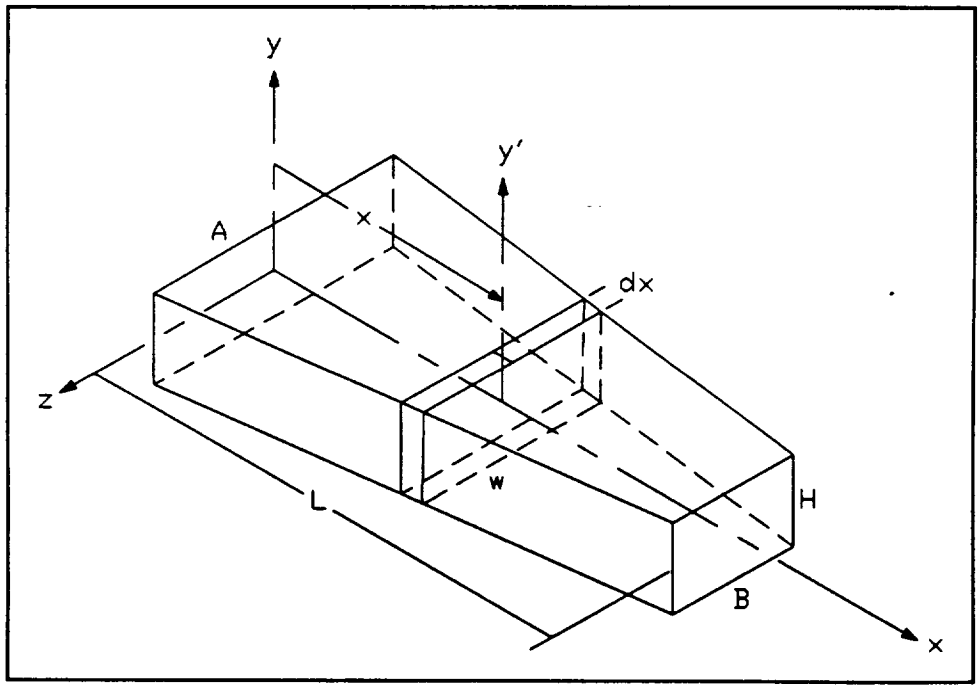


Figure C.1: Tapered Beam with Differential Element

Following the method outlined in Beer and Johnson [V], the differential mass of the element is:

$$dm = \rho H w dx \tag{C.1}$$

where ρ is the density of the material

For a linear taper the following relationship can be derived to express w as a function of A, B, and L:

$$w(x) = A + \left(\frac{B-A}{L}\right)x \tag{C.2}$$

The differential inertia about axis y' is:

$$dIy' = \frac{1}{12} w^2 dm \tag{C.3}$$

Using the parallel axis theorem one can determine dIy:

$$\begin{aligned} dIy &= dIy' + x^2 dm \\ &= \frac{1}{12} w^2 dm + x^2 dm \end{aligned} \tag{C.4}$$

Iy can be found by integrating the above expression for dIy over the length of the beam.

$$Iy = \int_0^L dIy \tag{C.5}$$

Making substitutions, the above equations simplify to:

$$dm = \rho H\frac{(AL - Ax + Bx)}{L} \tag{C.6}$$

$$dIy' = \rho H\frac{(AL - Ax + Bx)^3}{12L^3} \tag{C.7}$$

$$dIy = \rho H\frac{(AL-Ax+Bx)(A^2L^2-2A^2Lx+2ABLx+A^2x^2-2ABx^2+B^2x^2+12L^2x^2)}{12L^3} \tag{C.8}$$

$$Iy = \frac{\rho H}{48}(A^3 + A^2B + AB^2 + B^3 + 4AL^2 + 12BL^2) \tag{C.9}$$

Equation C.9 can be used to determine different tapered link shapes that will have the same value of inertia about the axis of rotation. This is helpful in evaluating the different link shapes. Once A, B, H, L, and ρ are selected for the initial link, Equation C.9 is used to evaluate the inertia, Iy. Assuming H, L, and ρ remain the same, B can be determined for various values of A. This involves solving a cubic equation in B, which is well suited to a program like *Mathematica*[1]

[1]*Mathematica* by Stephen Wolfram, a commercial program for doing mathematical computations, 1988.

APPENDIX D

MODE SHAPES FOR PINNED-PINNED BOUNDARY CONDITIONS

To implement the results of this research in the inverse dynamic control algorithm developed by Kwon and Book [8], the mode shapes must be determined for pinned-pinned boundary conditions. The natural frequencies for a tapered link are easily determined with the ZERO program by altering the code. Pinned-pinned boundary conditions change the frequency determinant which changes the search function in program ZERO. To help aid in this implementation, this appendix presents the state matrix and modes shapes for the first two natural frequencies of a given tapered design.

The tapered design was chosen from Table 4.5 and has A=0.6 in. and B=0.3 in. (R=2.0). As described earlier, the beam has L=40 in., H=1 in., and properties of aluminum. With the natural frequencies determined from program ZERO and the model parameter input file, *MATLAB*[1] was used to generate the mode shapes. For a discussion of mode shape generation using transfer matrices see Majette [10].

---

[1] *386-MATLAB*, a high-performance interactive software package for scientific and engineering numeric computation, The Mathworks, Inc., 1990.

The state matrix consists of the state vectors at each interface for the given natural frequency. The chosen design has twenty elements; therefore, the state matrix will have twenty-one columns and four rows. The state matrix is given for both the first and second natural frequencies. Recall from Chapter 3 that the state vector is described by Equation D.1. Figures D.1 and D.2 present the mode shapes for the first and second natural frequencies respectively.

$$u = \begin{bmatrix} -w \\ \psi \\ M \\ V \end{bmatrix} = \begin{bmatrix} displacement \\ slope \\ moment \\ shear\ force \end{bmatrix} \tag{D.1}$$

State matrix for first natural frequency (ω_1=7.886 rad/sec):

```
Columns 1-4:
   0.0000000e+00  -3.8934437e-04  -1.1406488e-03  -1.8071610e-03
  -3.7090242e-04  -3.6783076e-04  -3.4168377e-04  -2.8756676e-04
   0.0000000e+00   1.0483872e+00   3.0463353e+00   4.7608846e+00
   1.0000000e+00   9.8791457e-01   8.9500392e-01   7.2075561e-01

Columns 5-8:
  -2.3301876e-03  -2.6553124e-03  -2.7384772e-03  -2.5521281e-03
  -2.0611699e-04  -1.0076284e-04   2.2124724e-05   1.5332873e-04
   6.0383554e+00   6.7680119e+00   6.8915262e+00   6.4086547e+00
   4.8287558e-01   2.0427987e-01  -8.8434909e-02  -3.6704511e-01

Columns 9-12:
  -2.0908440e-03  -1.3757614e-03  -4.5703719e-04   5.8642232e-04
   2.8118602e-04   3.9233089e-04   4.7280928e-04   5.0954752e-04
   5.3781616e+00   3.9133899e+00   2.1723473e+00   3.4274484e-01
  -6.0413512e-01  -7.7571394e-01  -8.6376466e-01  -8.5844303e-01

Columns 13-16
   1.6506650e-03   2.6143541e-03   3.3506190e-03   3.7429093e-03
   4.9212125e-04   4.1471370e-04   2.7809584e-04   9.1400251e-05
  -1.3769667e+00  -2.7989053e+00  -3.7672233e+00  -4.1771564e+00
  -7.5963004e-01  -5.7756198e-01  -3.3231267e-01  -5.1995553e-02

Columns 17-20:
   3.6632194e-03   3.0754690e-03   2.0538451e-03   7.1951226e-04
  -1.6516173e-04  -3.8567645e-04  -5.7100923e-04  -6.7667652e-04
  -3.9894727e+00  -3.2418047e+00  -2.0443862e+00  -5.5675806e-01
   2.2929341e-01   4.7394149e-01   6.5190731e-01   7.4634877e-01

Column 21:
  -9.0368192e-17
  -6.8420890e-04
   2.3670925e-01
   7.5752083e-01
```

State matrix for second natural frequency (ω_2=32.06 rad/sec):

```
Columns 1-4:
   0.0000000e+00  -3.8934437e-04  -1.1406488e-03  -1.8071610e-03
  -3.7090242e-04  -3.6783076e-04  -3.4168377e-04  -2.8756676e-04
   0.0000000e+00   1.0483872e+00   3.0463353e+00   4.7608846e+00
   1.0000000e+00   9.8791457e-01   8.9500392e-01   7.2075561e-01


Columns 5-8:
  -2.3301876e-03  -2.6553124e-03  -2.7384772e-03  -2.5521281e-03
  -2.0611699e-04  -1.0076284e-04   2.2124724e-05   1.5332873e-04
   6.0383554e+00   6.7680119e+00   6.8915262e+00   6.4086547e+00
   4.8287558e-01   2.0427987e-01  -8.8434909e-02  -3.6704511e-01


Columns 9-12:
  -2.0908440e-03  -1.3757614e-03  -4.5703719e-04   5.8642232e-04
   2.8118602e-04   3.9233089e-04   4.7280928e-04   5.0954752e-04
   5.3781616e+00   3.9133899e+00   2.1723473e+00   3.4274484e-01
  -6.0413512e-01  -7.7571394e-01  -8.6376466e-01  -8.5844303e-01


Columns 13-16:
   1.6506650e-03   2.6143541e-03   3.3506190e-03   3.7429093e-03
   4.9212125e-04   4.1471370e-04   2.7809584e-04   9.1400251e-05
  -1.3769667e+00  -2.7989053e+00  -3.7672233e+00  -4.1771564e+00
  -7.5963004e-01  -5.7756198e-01  -3.3231267e-01  -5.1995553e-02


Columns 17-20:
   3.6632194e-03   3.0754690e-03   2.0538451e-03   7.1951226e-04
  -1.6516173e-04  -3.8567645e-04  -5.7100923e-04  -6.7667652e-04
  -3.9894727e+00  -3.2418047e+00  -2.0443862e+00  -5.5675806e-01
   2.2929341e-01   4.7394149e-01   6.5190731e-01   7.4634877e-01


Column 21:
  -9.0368192e-17
  -6.8420890e-04
   2.3670925e-01
   7.5752083e-01
```

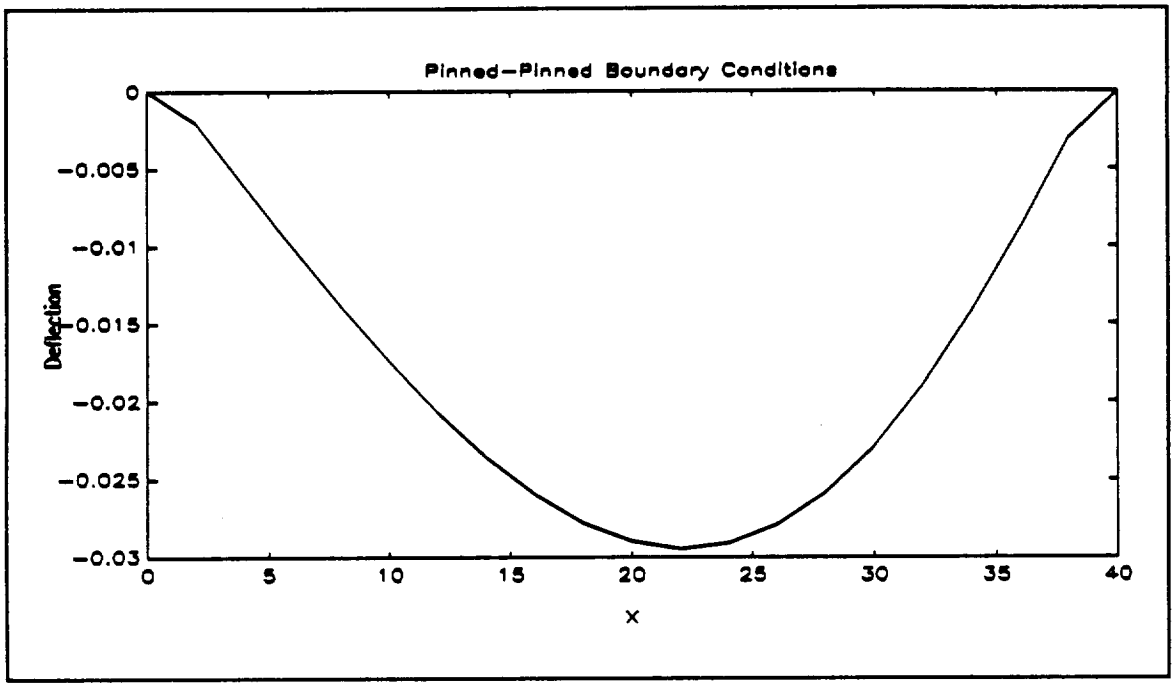


Figure D.1: First Mode Shape For Tapered Link

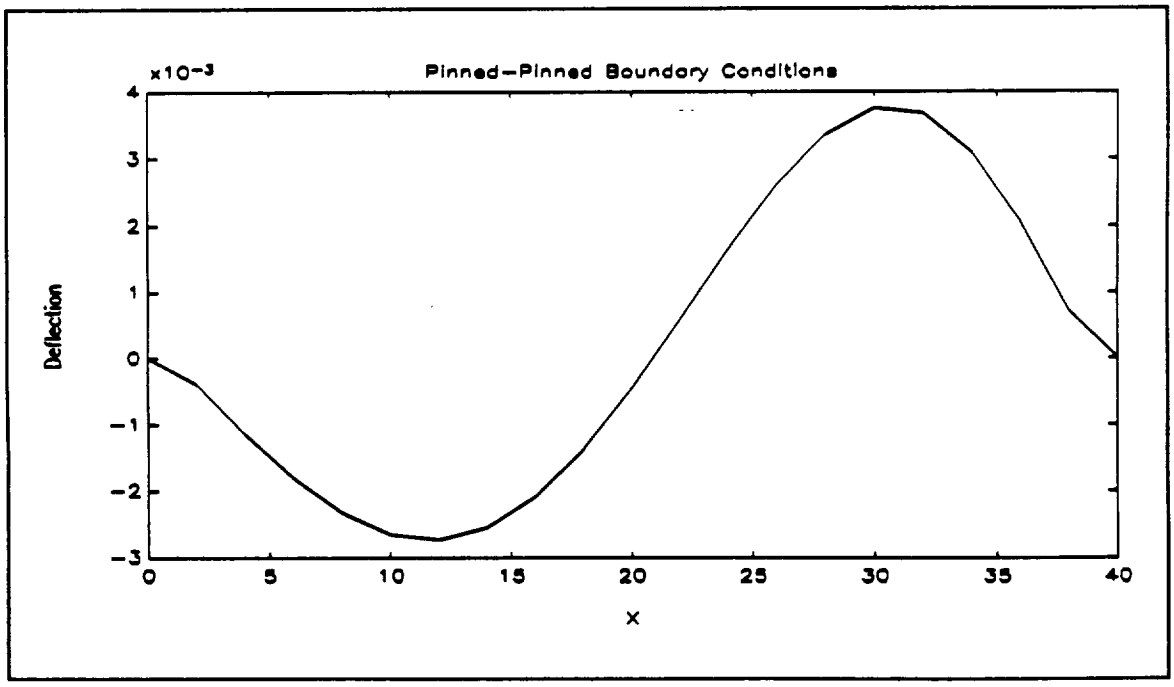


Figure D.2: Second Mode Shape For Tapered Link

APPENDIX E

PROGRAM SOURCE CODE

```
      PROGRAM ZERO

C  THIS PROGRAM IS DESIGNED TO IDENTIFY THE LOCATION OF SYSTEM ZEROS
C  AND POLES FOR THE NONCOLOCATED CONTROL STRUCTURE OF A SINGLE LINK
C  MANIPULATOR.  IT USES THE THEORY OF TRANSFER MATRIX ALGEBRA TO
C  GENERATE THE MODEL FOR THE BEAM.  THE PROGRAM IS DESIGNED TO HANDLE
C  A BEAM WITH VARIABLE CROSS-SECTIONAL AREA ALONG THE LONGITUDINAL AXIS
C  THIS MODEL ASSUMES NO DAMPING, AND THEREFORE THE ZEROS WILL LIE
C  ALONG THE REAL AXIS IN REFLECTED PAIRS ABOUT THE IMAGINARY AXIS AND
C  POLES WILL LIE ALONG THE IMAGINARY AXIS IN COMPLEX CONJUGATE PAIRS.
C  ALL CALCULATIONS ARE PERFORMED IN DOUBLE PRECISION.
C
C  VARIABLES:
C    NE- NUMBER OF ELEMENTS IN THE MODEL (I)
C    EP- ELEMENT PARAMETER MATRIX (R)
C    L- ELEMENT LENGTH, EP[?,1] (R)
C    MPL- MASS PER UNIT LENGTH, EP[?,2] (R)
C    AMI- AREA MOMENT OF INERTIA, EP[?,3] (R)
C    E- YOUNG'S MODULUS, EP[?,4] (R)
C    DF- DAMPING FACTOR, EP[?,5] (R)
C    FINPUT- STORES INPUT FILE NAME (C)
C    LIMIT- MAXIMUM NUMBER OF ELEMENTS (INTEGER PARAMETER)

C  **NOTE- DF MUST BE ZERO FOR THIS PROGRAM**

C  WRITTEN BY DOUG GIRVIN, 1991

      DOUBLE PRECISION EP,L,MPL,AMI,E,DF
      CHARACTER FILE*20,FINP*24,DUM*1
      PARAMETER (LIMIT=100)
      DIMENSION EP(LIMIT,5)
      COMMON NE,ITYPE,EP
C  INPUT FROM KEYBOARD OR FILE?
      WRITE(6,100)
      READ(5,*) N
      IF (N.EQ.1) GO TO 5
      IF (N.EQ.2) GO TO 10
C  DESCRIPTION OF INPUT VARIABLES
      WRITE(6,105)
      READ(5,*)N
      IF (N.EQ.1) GO TO 5
      IF (N.EQ.2) GO TO 10
C  MANUAL INPUT FROM KEYBOARD
5     WRITE(6,110)
      READ(5,*) NE
      DO 20 I=1,NE
        WRITE(6,115) I
        READ(5,*) EP(I,1),EP(I,2),EP(I,3),EP(I,4),EP(I,5)
20    CONTINUE
      GO TO 15
C  INPUT FROM A TEXT FILE
10    WRITE(6,120)
      READ(5,*) FILE
      FINP= FILE // '.INP'
      OPEN(12,FILE=FINP,STATUS='OLD')
      READ(12,130)
      READ(12,*) NE
      DO 25 I=1,NE
        READ(12,*) EP(I,1),EP(I,2),EP(I,3),EP(I,4),EP(I,5)
25    CONTINUE
C  CALL SUBDIVISION SUBROUTINE TO INPUT SEARCH INTERVALS
15    CALL SUBDIV(FILE)
      IF(N.EQ.2) CLOSE(12)
      WRITE(6,125)
100   FORMAT(///T2,'THIS PROGRAM DETERMINES THE LOCATION OF ZEROS',/,
     $          T2,'AND POLES FOR A BEAM USING TRANSFER MATRIX THEORY.',//,
```

```
     $          T2,'WOULD YOU LIKE TO ENTER THE MODEL INFORMATION',/,
     $          T2,'MANUALLY OR THRU AN INPUT FILE?',/,
     $          T2,'1 FOR MANUAL, 2 FOR FILE, 3 FOR INPUT DESCRIPTION')
105  FORMAT(//T2,'MODELING PARAMETERS:',//,
     $                '    NE-  NUMBER OF ELEMENTS IN THE MODEL',/,
     $                '    L-   LENGTH OF ELEMENTS',/,
     $                '    MPL- MASS PER UNIT LENTGH OF ELEMENT',/,
     $                '    AMI- AREA MOMENT OF INERTIA OF ELEMENT',/,
     $                '    E-   YOUNGS MODULUS OF ELEMENT',/,
     $                '    DF-  DAMPING FACTOR OF ELEMENT(MUST BE ZERO FOR',/,
     $                '         THIS PROGRAM)',//,
     $                ' TYPE: 1 FOR KEYBOARD INPUT, 2 FOR FILE INPUT')
110  FORMAT(/T2,'INPUT THE NUMBER OF ELEMENTS IN THE MODEL- NE')
115  FORMAT(/T2,'INPUT L,MPL,AMI,E,DF FOR ELEMENT ',I3)
120  FORMAT(//T2,'THE INPUT FILE MUST HAVE EXTENSION ".INP"',/
     $               T2,'AND LINES 1-5 ARE RESERVED FOR COMMENT'//
     $               T2,'WHAT IS THE FILE NAME, WITHOUT EXTENSION,'
     $               ' WITHIN APOSTROPHES')
125  FORMAT(//T2,'THE SCREEN OUTPUT CAN BE FOUND IN THE FILE WITH '
     $               /T2,'EXTENSION ".OUT" AND THE VALUES OF THE FUNCTION '
     $               /T2,'USED TO DETERMINE A ZERO CAN BE FOUND IN THE '
     $               /T2,'FILE WITH EXTENSION ".DAT"',//)
130  FORMAT(////)
     END
```

```
      SUBROUTINE SUBDIV(FILE)
C
C  THIS SUBROUTINE HANDLES USER INPUT OF THE COMPUTATION PARAMETERS
C  AND SEARCH INTERVALS FOR THE POLES AND ZEROS.  ONCE ALL INFORMATION
C  IS ENTERED, THE MAIN SEARCH INTERVAL IS DIVIDED INTO SUBINTERVALS TO
C  BE SENT TO ZFALSE.  THE RESULTS RETURNED FROM ZFALSE ARE PRINTED TO
C  THE SCREEN AND OUTPUT FILES FOR LATER EVALUATION.
C
C  VARIALBES:
C    FOUT- OUTPUT FILE FOR LOCATION OF POLES AND ZEROS (C)
C    RAW- OUTPUT FILE FOR ZERO (POLE) FUNCTION VALUES
C         AT EACH INTERVAL                                  (C)
C    EPS- FIRST CONVERGENCE CRITERIA FOR ZFALSE (R)
C    NSIG- SECOND CONVERGENCE CRITERIA FOR ZFALSE (I)
C    ITMAX- MAXIMUM ITERATIONS PER SUBINTERVAL (I)
C    LOW- LOWER LIMIT OF MAIN SEARCH INTERVAL (R)
C    HIGH- UPPER LIMIT OF MAIN SEARCH INTERVAL (R)
C    NDIV- NUMBER OF SUBINTERVALS (I)
C    DELT- LENGTH OF EACH SUBINTERVAL (R)
C    XL- LOWER LIMIT OF SUBINTERVAL PASSED TO ZFALSE (R)
C    XR- UPPER LIMIT OF SUBINTERVAL PASSED TO ZFALSE (R)
C    OMG- POLE OR ZERO FOUND BY ZFALSE (R)
C
C  WRITTEN BY DOUG GIRVIN, 1991
C
      DOUBLE PRECISION EPS,DELT,HIGH,LOW,XR,XL,OMG,XLOLD
      CHARACTER FILE*20,FOUT*24,RAW*24
      COMMON NE,ITYPE,EP
      FOUT= FILE // '.OUT'
      RAW= FILE // '.DAT'
      OPEN(10,FILE=FOUT,STATUS='NEW')
      OPEN(11,FILE=RAW,STATUS='NEW')
      WRITE(11,140) FILE
      WRITE(6,125)
      READ(5,*) ICP
      IF(ICP.EQ.2) GO TO 5
      WRITE(6,130)
5     WRITE(6,100)
      READ(5,*) EPS,NSIG,ITMAX
      WRITE(6,101)
C  ZERO CALCULATION
      READ(5,*) LOW,HIGH,NDIV
      ITYPE=1
      WRITE(10,135) FILE,EPS,NSIG,ITMAX,NE
      WRITE(6,120)
      WRITE(10,119)
      GOTO 15
C  POLE CALCULATION
10    WRITE(6,102)
      READ(5,*) LOW,HIGH,NDIV
      ITYPE=2
      WRITE(6,121)
      WRITE(10,*)
      WRITE(11,141)
15    DELT=(HIGH-LOW)/NDIV
      XL=LOW
      DO 20 I=1,NDIV
        XR=XL+DELT
        CALL ZFALSE(EPS,NSIG,XL,XR,OMG,ITMAX,IER)
C  MAXIMUM ITERATIONS REACHED WITHOUT CONVERGENCE
        IF(IER.EQ.130) THEN
          WRITE(6,105) XL,XR
        END IF
C  NO POLE (OR ZERO) IN SUBINTERVAL
        IF(IER.EQ.129) THEN
          IF(ITYPE.EQ.1) THEN
```

```
          WRITE(6,110) XL,XR
        ELSE
          WRITE(6,111) XL,XR
        END IF
      END IF
C  POLE (OR ZERO) WAS FOUND IN SUBINTERVAL
      IF(IER.LT.129) THEN
        IF(ITYPE.EQ.1) THEN
          WRITE(6,115) XL,XR,OMG
          WRITE(10,117)OMG
        ELSE
          WRITE(6,116) XL,XR,OMG
          WRITE(10,118)OMG
        END IF
      END IF
      XL=XR
20    CONTINUE
      IF(ITYPE.EQ.1) GOTO 10
100   FORMAT(1X,'INPUT EPS,NSIG,ITMAX')
101   FORMAT(1X,'INPUT LOW,HIGH,NDIV FOR ZEROS')
102   FORMAT(//,1X,'INPUT LOW,HIGH,NDIV FOR POLES')
105   FORMAT(T2,F9.2,' TO ',F9.2,T40,'MAX ITERATIONS REACHED WITHOUT'
     $                ' CONVERGENCE')
110   FORMAT(T2,F9.2,' TO ',F9.2,T40,'NO ZERO')
111   FORMAT(T2,F9.2,' TO ',F9.2,T40,'NO POLE')
115   FORMAT(T2,F9.2,' TO ',F9.2,T40,'ZERO AT ',F10.3)
116   FORMAT(T2,F9.2,' TO ',F9.2,T40,'POLE AT ',F10.3)
117   FORMAT(T2,'ZERO AT ',F10.3)
118   FORMAT(T2,'POLE AT ',F10.3)
119   FORMAT(//,T7,'RESULT'/,T7,'------')
120   FORMAT(//,T30,'ZEROS',/,T7,'SEARCH INTERVAL',T40,'RESULT'/,
     $               T7,'---------------',T40,'------'/)
121   FORMAT(//,T30,'POLES',/,T7,'SEARCH INTERVAL',T40,'RESULT'/,
     $               T7,'---------------',T40,'------'/)
125   FORMAT(1X,'WOULD YOU LIKE DEFINITIONS OF THE COMPUTATION'
     $               ' PARAMETERS?'/' 1 FOR YES, 2 FOR NO')
130         FORMAT(1X,'COMPUTATION PARAMETERS IN ORDER OF INPUT:'//
     $               '  EPSILON= FIRST CONVERGENCE CRITERION. A TRIAL'/
     $               '           ROOT, X, IS ACCEPTED IF ABS[F(X)]<EPS'//
     $               '  NSIG= SECOND CONVERGENCE CRITERION. A TRIAL'/
     $               '        ROOT, X, IS ACCEPTED IF IT AGREES WITH'/
     $               '        THE PREVIOUS TRIAL VALUE TO NSIG SIGNI-'/
     $               '        FICANT DIGITS.'//
     $               '  ITMAX= THE MAXIMUM NUMBER OF ITERATIONS PER'/
     $               '         SUBINTERVAL'//
     $               '  LOW= THE LOWER BOUND OF THE SEARCH INTERVAL'//
     $               '  HIGH= THE UPPER BOUND OF THE SEARCH INTERVAL'//
     $               '  NDIV= THE NUMBER OF SUBDIVISIONS IN THE MAIN'/
     $               '        SEARCH INTERVAL [LOW,HIGH]'//)
135   FORMAT(1X,'CALCULATION PARAMETERS:',2X,A,//,T4,'EPS=',T14,E10.3
     $              ,/,T4,'NSIG=',T14,I4,/,T4,'ITMAX=',T14,I4,/,T4,'NE=',
     $              T14,I3)
140   FORMAT(T2,'FILE:',2X,A//,T30,'ZEROS',/
     $              T8,'XLL',T21,'XRR',T36,'FXL',T53,'FXR',/)
141   FORMAT(//,T30,'POLES',/
     $              T8,'XLL',T21,'XRR',T36,'FXL',T53,'FXR',/)
      CLOSE(10)
      CLOSE(11)
      RETURN
      END
```

```
      SUBROUTINE ZFALSE(EPS,NSIG,XL,XR,XAPP,ITMAX,IER)
      DOUBLE PRECISION F2,EPS,XL,XR,XAPP,ZERO,TEN,HALF,XLL,XRR
      DOUBLE PRECISION EPSP,FXL,FPREV,FXR,FXAPP
      PARAMETER (ZERO=0.0D0,TEN=10.0D0,HALF=0.5D0)
      COMMON NE,ITYPE,EP
      IER = 0
      IC = 0
      XLL = DMIN1(XL,XR)
      XRR = DMAX1(XL,XR)
      IF(XL .NE. XLL)IER = 35
      EPSP = TEN**(-NSIG)
      FXL = F(XLL)
      FPREV = FXL
      FXR = F(XRR)
      IF (FXL*FXR) 15,10,5
C                                     TERMINAL ERROR
    5 IER=129
      WRITE(11,555)XLL,XRR,FXL,FXR
      GO TO 40
C                                     FXL OR FXR = 0
   10 XAPP = XRR
      IF (FXL .EQ. ZERO) XAPP = XLL
      WRITE(11,556)XLL,XRR,FXL,FXR
      GO TO 40
C                                     COMPUTE APPROXIMATE ROOT
   15 XAPP = XLL+FXL*(XRR-XLL)/(FXL-FXR)
      FXAPP = F(XAPP)
      IF (DABS(FXAPP) .GT. EPS) GO TO 20
      IF (ITYPE.EQ.1) THEN
        WRITE(11,600) XL,XR,FXAPP
      ELSE
        WRITE(11,601) XL,XR,FXAPP
      ENDIF
      GO TO 40
C                                     DETERMINE WHETHER APPROXIMATE ROOT
C                                     LIES BETWEEN XAPP AND XLL OR XAPP
C                                     AND XRR
   20 IF (FXAPP*FXL .GT. ZERO) GO TO 25
      XRR = XAPP
      FXR = FXAPP
      IF (FPREV*FXR .GT. ZERO) FXL = HALF*FXL
      FPREV = FXR
      GO TO 30
   25 XLL = XAPP
      FXL = FXAPP
      IF (FPREV*FXL .GT. ZERO) FXR = HALF*FXR
      FPREV = FXL
   30 IF (XRR-XLL .GT. EPSP*DABS(XRR)) GO TO 35
      IF (ITYPE.EQ.1) THEN
        WRITE(11,600) XL,XR,FXAPP
      ELSE
        WRITE(11,601) XL,XR,FXAPP
      ENDIF
      GO TO 40
C                                     CONTINUE FOR ITMAX ITERATIONS
   35 IC = IC+1
      IF (IC .LE. ITMAX) GO TO 15
      IER = 130
   40 IF (IER .NE. 0) GO TO 9000
      GO TO 9005
 9000 CONTINUE
  555 FORMAT(2X,F10.2,' TO',F10.2,6X,E11.4,6X,E11.4)
  556 FORMAT(2X,F10.2,' TO',F10.2,6X,E11.4,6X,E11.4,' *')
  600 FORMAT(2X,F10.2,' TO',F10.2,8X,'ZERO: F= ',E11.4)
  601 FORMAT(2X,F10.2,' TO',F10.2,8X,'POLE: F= ',E11.4)
 9005 RETURN
      END
```

```
      DOUBLE PRECISION FUNCTION F(X)

C  THIS FUNCTION DETERMINES THE VALUE OF F BASED ON THE TRANSFER
C  MATRIX GENERATED BY THE BUILD SUBROUTINE.  F CAN BE A FUNCTON TO
C  DETERMINE ZERO LOCATION OR POLE LOCATION DEPENDING ON THE FLAG
C  ITYPE.

C  VARIABLES
C    ITYPE- CALC TYPE: 1=ZERO, 2=POLE (I)
C    F- VALUE OF FUNCION (R)
C    OMG- TRIAL FREQUENCY (C)
C    TM- OVERALL TRANSFER MATRIX OF MODEL (R)

C  WRITTEN BY DOUG GIRVIN, 1991

      DOUBLE COMPLEX OMG
      DOUBLE PRECISION TM,C1,C2,C3,C4,C5,C6,NUM,DEN,X,EP
      DIMENSION TM(4,4)
      COMMON NE,ITYPE,EP
      IF(ITYPE.EQ.1)THEN
        OMG=DCMPLX(0.0D0,-X)
        CALL BUILD(OMG,TM)
C  CALCULATE FUNCTION THAT EVALUATES ZEROS
        C1=TM(1,2)*TM(3,3)*TM(4,4)
        C2=TM(1,2)*TM(3,4)*TM(4,3)
        C3=TM(1,3)*TM(3,4)*TM(4,2)
        C4=TM(1,3)*TM(3,2)*TM(4,4)
        C5=TM(1,4)*TM(3,2)*TM(4,3)
        C6=TM(1,4)*TM(3,3)*TM(4,2)
        NUM=C1-C2+C3-C4+C5-C6
        DEN=(TM(3,4)*TM(4,2))-(TM(3,2)*TM(4,4))
        F=-NUM/DEN
      ELSE
        OMG=DCMPLX(X,0.0D0)
        CALL BUILD(OMG,TM)
C  CALCULATE FUNCTION THAT EVALUATES POLES
        F=(TM(3,2)*TM(4,4))-(TM(3,4)*TM(4,2))
      END IF
      RETURN
      END
```

```
      SUBROUTINE BUILD(OMG,TM)

C  THIS SUBROUTINE GENERATES THE OVERALL TRANSFER MATRIX FOR THE
C  MODELED BEAM BY MULTIPLYING THE ELEMENT TRANSFER MATRICES GENERATED
C  BY THE BEAM4 SUBROUTINE.  MULTIPLICATION IS PERFORMED BY THE MUL
C  SUBROUTINE.  ALL CALCULATIONS ARE PERFORMED IN DOUBLE PRECISION
C  AND DOUBLE COMPLEX FOR HIGHER ACCURACY.  THIS SUBROUTINE  USES THE
C  REAL FORM OF THE B MATRIX.

C  VARIBLES:
C    IE- ELEMENT NUMBER (I)
C    NE- TOTAL NUMBER OF ELEMENTS IN MODEL (I)
C    OMG- TRIAL FREQUENCY (C)
C    B- ELEMENT TRANSFER MATRIX (R)
C    TM- OVERALL TRANSFER MATRIX OF MODEL (R)

C  WRITTEN BY DOUG GIRVIN, 1991

      DOUBLE COMPLEX OMG
      DOUBLE PRECISION B,TM,EP
      COMMON NE,ITYPE,EP
      DIMENSION B(4,4), TM(4,4), EP(100,5)
      DO 20 IE=1,NE
        CALL BEAM4(IE,OMG,B)
        IF (IE.EQ.1) THEN
                  DO 10 I=1,4
           DO 10 J=1,4
10        TM(I,J)=B(I,J)
          GOTO 20
        ELSE
          CALL MUL(B,TM,4)
        ENDIF
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BEAM4(IE,OMG,B)

C  THIS SUBROUTINE GENERATES THE TRANSFER MATRIX FOR A EULER-BERNOULLI
C  BEAM ELEMENT GIVEN THE ELEMENT NUMBER AND THE TRIAL FREQUENCY.
C  ALL CALCULATIONS ARE PERFORMED IN DOUBLE PRECISION AND DOUBLE
C  COMPLEX FOR HIGHER ACCURACY.  THIS SUBROUTINE RETURNS THE B MATRIX
C  AS A REAL MATRIX.

C  VARIABLES:
C    IE- ELEMENT NUMBER (I)
C    OMG- TRIAL FREQUENCY (C)
C    BC- ELEMENT TRANFER MATRIX (C)
C    B- ELEMENT TRANSFER MATRIX (R)
C    EP- ELEMENT PARAMETER MATRIX (R)

C  WRITTEN BY DOUG GIRVIN, 1991

      DOUBLE PRECISION MPL,L,REI,CEI,EP,B
      DOUBLE COMPLEX BC,OMG,OMG2,EI,B4,B2,B1,AR,CCS,CSN,CEP,CEN
      DOUBLE COMPLEX CCSH,CSNH,C0,C1,C2,C3
      COMMON NE,ITYPE,EP
      DIMENSION B(4,4),BC(4,4),EP(100,5)
C  PRELIMINARY CALCULATIONS
      OMG2= OMG*OMG
      L= EP(IE,1)
      REI= EP(IE,4)*EP(IE,3)
      CEI= EP(IE,3)*EP(IE,5)
      EI= DCMPLX(REI,CEI)
      MPL= EP(IE,2)
      B4= MPL*OMG2*L**4/EI
C  VAIABLES NEEDED TO CALCULATE BC MATRIX
      B2= CDSQRT(B4)
      B1= CDSQRT(B2)
      AR= L*L/EI
      CCS= CDCOS(B1)
      CSN= CDSIN(B1)
      CEP= CDEXP(B1)
      CEN= CDEXP(-B1)
      CCSH= 0.5D0*(CEP+CEN)
      CSNH= 0.5D0*(CEP-CEN)
      C0= 0.5D0*(CCSH+CCS)
      C1= 0.5D0*(CSNH+CSN)/B1
      C2= 0.5D0*(CCSH-CCS)/B2
      C3= 0.5D0*(CSNH-CSN)/(B1*B2)
C CALCULATE UPPER HALF OF BC MATRIX
      BC(1,1)= C0
      BC(1,2)= L*C1
      BC(1,3)= AR*C2
      BC(1,4)= AR*L*C3
      BC(2,1)= B4*C3/L
      BC(2,2)= C0
      BC(2,3)= AR*C1/L
      BC(3,1)= B4*C2/AR
      BC(3,2)= B4*L*C3/AR
      BC(4,1)= B4*C1/(AR*L)
C CONVERT BC TO REAL B MATRIX
      DO 20 I=1,4
        DO 20 J=1,5-I
20    B(I,J)=DREAL(BC(I,J))
C GENERATE LOWER HALF OF B MATRIX (MIRROR IMAGE)
      DO 10 I=1,3
        IS= 5-I
        IU= 4-I
        DO 10 J=1,IU
          JS= 5-J
10    B(JS,IS)= B(I,J)
      RETURN
      END
```

```
      SUBROUTINE MUL(X,Y,N)

C  THIS SUBROUTINE MULTIPLIES TWO REAL MATRICES IN THE ORDER X*Y,
C  AND STORES THE RESULT IN Y (MATRIX X IS PRSERVED). THE MATRICES MUST
C  BE SQUARE AND HAVE DIMENSIONS N BY N.

C  VARIABLES:
C    N- SIZE OF MATRICES (I)
C    X- MATRIX (R)
C    Y- MATRIX (R)
C    T- TEMPORARILY STORES RESULT (R)

C  WRITTEN BY DOUG GIRVIN, 1991

      DOUBLE PRECISION X,Y,T
      DIMENSION X(4,4), Y(4,4), T(4,4)
      DO 10 I=1,N
        DO 10 J=1,N
          T(I,J)=0.0D0
          DO 10 K=1,N
10    T(I,J)=T(I,J)+X(I,K)*Y(K,J)
      DO 20 I=1,N
        DO 20 J=1,N
20    Y(I,J)=T(I,J)
      RETURN
      END
```

BIBLIOGRAPHY

[1] Asada, H., Park, J.-H., and Rai, S., "A Control-Configured Flexible Arm: Integrated Structure/Control Design," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, April, 1991, pp. 2356-2362.

[2] Bayo, E., "A Finite Element Approach to Control the End-Point Motion of a Single-Link Flexible Robot," *Journal of Robotic Systems*, Vol. 4, No. 1, 1987, pp.63-75.

[3] Beer, Ferdinand, and Johnson, Russell, Jr., *Vector Mechanics for Engineers, Statics and Dynamics*, Third Edition, McGraw-Hill, New York, 1977.

[4] Book, W. J., *Design and Control of Flexible Manipulator Arms*, Ph.D. Thesis, Massachusetts Institute of Technology, April, 1974.

[5] Book, W. J., and Kwon, D.-S., "Contact Control for Advanced Applications of Light Weight Arms," *Symposium on Control of Robots and Manufacturing*, Arlington, Texas, 1990.

[6] Book, W. J., Majette, M., and Ma, K., *The Distributed Systems Analysis Package (DSAP) and Its Application to Modeling Flexible Manipulators*, NASA Contract NAS 9-13809, Subcontract No. 551, School of Mechanical Engineering, Georgia Institute of Technology, 1979.

[7] Churchill, R. V., and Brown, J. W., *Complex Variables and Applications*, Fifth Edition, McGraw-Hill Publishing Company, New York, 1990.

[8] Kwon, D.-S., *An Inverse Dynamic Tracking Control for Bracing A Flexible Manipulator*, Ph.D. Dissertation, Georgia Tech, Woodruff School of Mechanical Engineering, June, 1991.

[9] Kwon, D.-S., and Book, W. J., "An Inverse Dynamics Method Yielding Flexible Manipulator State Trajectories," *Proceedings of the American Control Conference*, June, 1990, pp. 186-193.

[10] Majette, M., *Modal State Variable Control of a Linear Distributed Mechanical System Modeled with the Transfer Matrix Method*, Master's Thesis, Georgia Tech, Woodruff School of Mechanical Engineering, June, 1985.

[11] Meirovitch, L., *Elements of Vibrational Analysis*, McGraw-Hill, New York, 1986.

[12] Misra, Pradee, "On The Control of Non-Minimum Phase Systems," *Proceedings of the 1989 American Control Conference*, 1989, pp. 1295-1296.

[13] Nebot, E. M., Lee, G. K. F., and Brubaker, T. A., "Experiments on a Single Link Flexible Manipulator,", *Proceedings from the USA-Japan Symposium on Flexible Automation Crossing Bridges: Advances in Flexible Automation and Robotics*, 1988, pp. 391-398.

[14] Park, J.-H., and Asada, H., "Design and Analysis of Flexible Arms for Minimum-Phase Endpoint Control," *Proceedings of the American Control Conference*, 1990, pp. 1220-1225.

[15] Park, J.-H., and Asada, H., "Design and Control of Minimum-Phase Flexible Arms with Torque Transmission Mechanisms," *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, 1990, pp. 1790-1795.

[16] Pestel and Leckie, *Matrix Methods in Elastomechanics*, McGraw-Hill, New York, 1963.

[17] Rao, Singiresu S., *Mechanical Vibrations*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[18] Spector, V. A., and Flashner, H., "Modeling and Design Implications of Noncollocated Control in Flexible Systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 112, June, 1990, pp. 186-193.

[19] Spector, V. A., and Flashner, H., "Sensitivity of Structural Models for Noncollocated Control Systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 111, December, 1989, pp. 646-655.