AN ABSTRACT OF THE THESIS OF

Byeung Kyun Lee for the degree of Doctor of Philosophy in Mechanical Engineering presented on June 3, 1988.

Title: A Model Reference Adaptive System for Control of a Flexible Mechanical Manipulator

Abstract approved: Redacted for privacy

Charles E. Smith

For control of a flexible manipulator, the assumed-mode method is applied to the mathematical model of the model reference adaptive system. In the assumed-mode method, the flexible, continuous manipulator is described by a limited number of degrees of freedom. A modified model reference adaptive system is studied for direct application of the adaptive control scheme to the control of a flexible manipulator. Use of the assumed-mode method and the modified model reference adaptive system reduce difficulties in designing the controller of the flexible manipulator. A numerical simulation, using the above procedure, is developed to identify a flexible manipulator with unknown parameters and simulation results show the satisfactory convergence of the parameters.

A Model Reference Adaptive System for Control of a Flexible Mechanical Manipulator

by

Byeung Kyun Lee

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed June 3, 1988

Commencement June 1989

APPROVED:

Redacted for privacy

Professor of Mechanical Engineering in charge of major

Redacted for privacy

Head of Department of Mechanical Engineering

Redacted for privacy

Dean of Graduate School

Date thesis is presented June 3, 1988

Typed by B. McMechan for Byeung Kyun Lee

Acknowledgements

It is with great pleasure that I acknowledge the inspirational guidance and continuing advice of my advisor, Dr. Charles E. Smith, during the completion of this project. I would also like to express my sincere appreciation to Dr. Ronald B. Guenther and Dr. Timothy C. Kennedy for their encouragement and advice. My sincere gratitude is also extended to Mr. John L. Williams for his consistent encouragement and valuable help and to Dr. William McMechan for his readings and suggestions on my writings.

I am indebted to Dr. James R. Welty and Dr. Gordon M. Reistad, the former and the current Head of the Department of Mechanical Engineering for their continuous financial support and advice during my graduate studies at Oregon State University.

I must also acknowledge my parents for their unending support and prayers during my period of work in the United States, and my wife Mi-Sook and my children, Eungi and Sung-Yeup, for their understanding and patience during my long period of study.

Finally, I thank my Lord, through whom I have known the meaning of my life, for His everlasting love for me.

Table of Contents

Table of Contents (continued)

List of Figures

List of Figures (continued)

List of Figures (continued)

A MODEL REFERENCE ADAPTIVE SYSTEM FOR CONTROL OF A FLEXIBLE MECHANICAL MANIPULATOR

I. INTRODUCTION

Since the first use of the term "robot" in 1920, robotic technology has been applied to a number of manufacturing areas and at present has increasingly been applied to practical purposes. The study of robotics is an interdisciplinary field, including the technology of mechanical and electrical component design, motion analysis, controller and sensor design, and artificial intelligence. The various robotics applications require the type of continuous research and development effort associated with advanced performance problems.

Mechanical manipulators are a case in point. In many instances, mechanical manipulators are considered as rigid bodies, existing for the convenience of analyzing kinetic and dynamic motion and controller design. In order to satisfy the assumption of rigid body motion, the structure must possess reasonable stiffness, a requirement which increases the weight and therefore the sizes of the high powered actuators necessary for control. In the case of a long and thin manipulator, the deflection of the manipulator causes a decrease in robotic accuracy.

In this investigation, the flexible manipulator has been studied as a physical model for the consideration of an increase in accuracy, a decrease in weight, and a reduction in actuator sizes. Motion analysis of a flexible manipulator involves completion of a number of difficult tasks, including system modeling, feedback sensing techniques, and control strategies associated with distributed parameters. Several studies of the flexible manipulator have examined these tasks [4,5,6, 13,35,45] and have derived a number of advantages from the development of robotic flexible manipulators: an increase in accuracy, higher speeds, smaller actuators, lower energy consumption, lower overall cost, safer operation due to reduced inertia, less bulky design, enhanced back-driveability due to the elimination of gearing, lower overall mass to be transported, and lowered mounting strength and rigidity requirements. Among these advantages, this study has focused upon the improvement of accuracy standards.

Most research studies of flexible manipulators have been based on the assumption that the parameters of the system are known. When the flexible manipulator reflects unknown parameters, imperfect modeling conditions, or variations in parameters, control strategies become more complicated. With conventional control strategies, control of the flexible manipulator becomes more difficult to achieve in proportion to model uncertainties. For instance, when the payload is uncertain, when the system modeling is imperfect, or when the parameters vary in time, the control scheme for the system should have the ability to adjust or adapt to these effects.

Based upon this consideration, the physical model of a mechanical manipulator developed for this study is based upon the assumptions that the manipulator has a one-link flexible arm and the system contains unknown parameters or variations in parameters. For development of the mathematical modeling for this flexible manipulator, an assumed-mode method is employed. The assumed-mode method is based upon a set of admissible functions, which satisfy the geometric boundary conditions of the system under consideration, and generalized coordinates, used in conjunction with the application of Lagrange's equations to obtain an approximate formulation of the equations of motion. This method has the advantage of reducing a continuous system to a multi-degree-of-freedom system, quite similar to the Rayleigh-Ritz method. The dynamic response is then obtained, based on the mode-superposition method in which a set of coupled equations can be transformed into a set of uncoupled equations through use of the normal modes of the system. When reduction of the number of assumed modes is necessary, the mode-acceleration method suggested by Williams [44] may be applied.

To control a system for which the characteristics are imperfectly known or the parameters are varied, adaptive control systems can offer highly effective control schemes [8,19,20]. Among various adaptive control systems, the model reference adaptive system is investigated for the flexible manipulator control scheme.

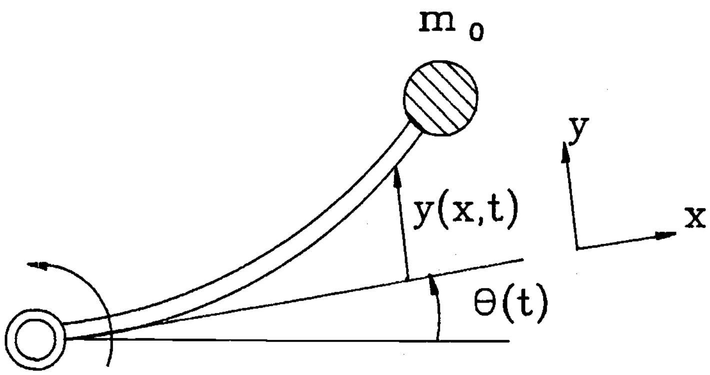
In Chapter II, a physical model of a flexible manipulator is described, accompanied by the introduction of the assumed-mode method for constructing a mathematical model. In Chapter III, the concept of the model reference adaptive system, in conjunction with hyperstability theory, is discussed. The application of the model reference adaptive system to system control, represented in the form of partial differential equations, is difficult. Problems in this form of expression can be overcome by use of the assumed-mode method for mathematical modeling, and a modified control scheme for the model reference adaptive system is presented in Chapter IV. Computer simulation of the model and the results of this study are also included in Chapter IV and conclusions are offered in Chapter V.

II. MODEL DESCRIPTION

There are various methods of describing nonrigid, distributed parameter systems. However, it is difficult to precisely analyze the mathematical models of flexible manipulators due to their nonlinearities and coupled terms. Moreover, control maneuvers for a system described in partial differential equations are complicated. In this chapter, a physical model of a one-link flexible manipulator is considered and a discrete method for the approximation of a continuous system, the assumed-mode method, is discussed. The principal advantage of the application of the assumed-mode method to the modeling of a flexible manipulator is the convenient application of control strategies without the loss of accuracy.

2.1 Physical Model

The one-link flexible manipulator represented in Figure 2.1 has been selected as the physical model for this investigation. When desired, the study of one-link flexible manipulators may be extended to the examination of multi-link flexible manipulators. For the



Assumptions:

- Small defelction
- Rotary intertia and shear deformation effects ignored
- Payload at the tip of the manipulator
- Base motion prespecified

Figure 2.1 Physical Model

model selected, the following simplifying assumptions have been considered:

1) Motion occurs entirely in the x-y plane;
2) The deflection, $y(x,t)$, of the manipulator during motion is so slight that any axial extension may be ignored;
3) Torsinal deflection is neglected;
4) The model is considered as a Bernoulli-Euler beam, i.e., rotational inertia and shear deflection are ignored; and
5) The arm remains straight at rest, i.e., at rest deflection is zero.

In partial differential equations, the mathematical model of the flexible beam is represented in terms of the position x and the time t. In the following section, an assumed-mode method for the development of a mathematical model of the flexible manipulator is introduced.

2.2 Theoretical Background of the Assumed-Mode Method

Precise expressions of a flexible beam are presented in the form of partial differential equations. In a practical sense, the use of a partial differential model is cumbersome and, in many cases, closed-form solutions of the exact mathematical model cannot be obtained. Consequently, various methods of approximating continuous systems have been developed [*9,12,22,32*].

Among these methods, two discrete approximations of a flexible beam, the Galerkin method and the assumed-mode method, have been considered during the course of this investigation.

The Galerkin method is a procedure which allows elimination of spatial dependence by discretizing spatial variables, resulting in ordinary rather than partial differential equations. The assumed-mode method discretizes the equations for kinetic energy, potential energy, and external forces prior to application of the Lagrange equations, generating equations of motion for the physical model in generalized coordinates which approximate the dynamic responses of the flexible system. Though the two approaches are quite similar, the second method encompasses more convenient discretization since it provides a solution series. Therefore, it is the method chosen for the derivation of the equations of motion for the flexible manipulator. For the generation of expressions of the dynamic model in uncoupled form, the mode superposition method is employed.

In order to understand the theoretical background of the assumed-mode method, it is first necessary to examine the Rayleigh-Ritz method and Galerkin's method, two procedures which allow conversion of eigenvalue problems of a continuous system into eigenvalue problems for a discrete system. This is done by the assumption of a solution in the form of a finite series, consisting of known functions multiplied by unknown coefficients. Depending on the method used, assumed functions may be selected as comparison functions or admissible functions. If the series consists of N functions, the corresponding eigenvalue problem yields N-eigenvalues and N associated eigenvectors. The components of each of the resulting N-dimensional eigenvectors are multiplied by their respective assumed functions to obtain the desired eigenfunctions. The following are distinctions of admissible functions and comparison functions as assumed functions, which are briefly discussed below.

1) Admissible functions are any arbitrary functions which satisfy all of the geometric boundary conditions of the eigenvalue problem, or the system under consideration, possessing derivatives of the order at least equal to that appearing in the strain energy expression for the system.

2) Comparison functions are any arbitrary functions which satisfy all boundary conditions (geometric and natural) of the eigenvalue problem, or the system under consideration, possessing derivatives of order at least twice that appearing in the strain energy expression for the problem or the system.

2.2.1 Rayleigh's Energy Method

Based on Rayleigh's principle [40], Rayleigh's method can be used to obtain an approximate value of the fundamental frequency of a system without solving the equations of motion for either a discrete or a continuous system. In the case of continuous systems, this method is useful when the system stiffness and mass are not uniformly distributed and an exact solution of the eigenvalue problem is impossible to obtain.

For a continuous system, the general eigenvalue problem takes the form

$$L[w] = \lambda\, M[w] \;, \qquad (2.2.1)$$

where L and M are linear homogeneous differential operators. Any eigenvalue, λ_i, with the associated eigenfunction, w_i, must satisfy Equation (2.2.1) and the associated boundary conditions of the problem. Therefore, Equation (2.2.1) can be rewritten as

$$L[w_i] = \lambda\, M[w_i], \; i = 1,2, - - - . \qquad (2.2.2)$$

Multiplying both sides of Equation (2.2.2) by w_i, integration over the domain x yields

$$\lambda_i = \frac{\int w_i \; L[w_i] \; dx}{\int w_i \; M[w_i] \; dx} \;, \quad i = 1,2, - - - . \qquad (2.2.3)$$

The expression of Rayleigh's quotient can be obtained from the assumption that the boundary conditions do not depend on the eigenvalue λ and that u be a comparison function as follows:

$$w^2 = \frac{\int u\ L[u]\ dx}{\int u\ M[u]\ dx} . \tag{2.2.4}$$

For illustration, this method can be applied to the transverse vibration of a flexible beam which is simulated as the physical model of a flexible manipulator.

Transverse displacement is described by

$$y(x,t) = X(x)\ q(t) , \tag{2.2.5}$$

where X(x) represents the transverse displacement at point x and q(t) the harmonic time dependent function. Its kinetic energy is expressed in the form

$$\begin{aligned} T(t) &= \frac{1}{2} \int_0^L m(x) \left[\frac{\partial y(x,t)}{\partial t}\right]^2 \\ &= \frac{1}{2} \dot{q}(t)^2 \int_0^L m(x)\ X(x)^2\ dx \end{aligned} \tag{2.2.6}$$

and potential energy in the form

$$\begin{aligned} V(t) &= \frac{1}{2} \int_0^L EI \left[\frac{\partial^2 y(x,t)}{\partial x^2} \right]^2 dx \\ &= \frac{1}{2} q(t)^2 \int_0^L EI \left[\frac{\partial^2 X}{\partial x^2}\right]^2 . \end{aligned} \tag{2.2.7}$$

Equations (2.2.6) and (2.2.7), introduced to Equation (2.2.4) with the energy conservation law, yield:

$$w^2 = R\{X(x)\} = \frac{\int_0^L EI \left[\frac{\partial^2 X}{\partial x^2}\right]^2 dx}{\int_0^L m(x)\ X^2(x)\ dx} . \tag{2.2.8}$$

This method can be used as a procedure for approximating a continuous system by a single-degree-of-freedom system and for calculation of approximate fundamental frequency.

2.2.2 The Rayleigh-Ritz Method

Rayleigh's quotient provides an upper bound for the first eigenvalue, λ_1,

$$R(u) \geq \lambda_1 \ , \qquad (2.2.9)$$

where the equality sign holds true if, and only if, the comparison function u is actually the first eigenfunction of the system, i.e. the true fundamental frequency is always smaller that the estimated one. In the Rayleigh-Ritz method, the main object is to minimize the estimate. The method for a multi-degree-of-freedom system approximates the frequencies of the reduced number of modes. It selects the minimizing sequence series of admissible functions, X_i, which satisfy all boundary conditions of the system, and constructs a linear combination,

$$w_n = \sum_{i=1}^{N} U_i X_i \ , \qquad (2.2.10)$$

where X_i are preselected, linearly independent functions and U_i are unknown coefficients to be obtained.

The substitution of w_n into Rayleigh's quotient leads to

$$R(w_n) \equiv w_n{}^2 = \frac{\int w_n \ L[w_n] \ dx}{\int w_n \ M[w_n] \ dx} = \frac{N(w_n)}{D(w_n)} \ , \qquad (2.2.11)$$

where

$$N(w_n) = \int w_n \, L[w_n] \, dx \text{ and} \tag{2.2.12}$$

$$D(w_n) = \int w_n \, M[w_n] \, dx \, , \tag{2.2.13}$$

indicate, respectively, the numerator and denominator of Rayleigh's quotient.

For the determination of the coefficient U_i, Ritz proposed to make Rayleigh's quotient, $R(w_n)$, stationary, leading to

$$\frac{\partial R(w_n)}{\partial U_i} = 0, \quad i=1,2, - - -, N \, . \tag{2.2.14}$$

Then, Equation (2.2.13) gives

$$N(w_n) \frac{\partial D(w_n)}{\partial U_i} - D(w_n) \frac{\partial N(w_n)}{\partial U_i} = 0 \tag{2.2.15}$$

and the condition (2.2.15) becomes

$$\frac{\partial N(w_n)}{\partial U_i} - w^2 \frac{\partial D(w_n)}{\partial U_i} = 0, \quad i=1,2,- - -,N \tag{2.2.16}$$

where min $R(w_n)$ is denoted as w^2.

Then, let

$$\left.\begin{aligned} k_{ij} &= \int X_i \, L[X_j] \, dx \\ m_{ij} &= \int X_i \, M[X_j] \, dx \end{aligned}\right\} \; i,j = 1,2,- - -,N \tag{2.2.17}$$

and if the system is self-adjoint, then

$$k_{ij} = k_{ji}, \; m_{ij} = m_{ji} \, . \tag{2.2.18}$$

Since the operators L and M are linear, Equations (2.2.12) and (2.2.13) become

$$N = \int \sum_{i=1}^{N} U_i \, X_i \, L\{ \sum_{j=1}^{N} U_j \, X_j \} \, dx$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} k_{ij} \, U_i \, U_j \, , \text{ and} \tag{2.2.19}$$

$$D = \int \sum_{i=1}^{N} U_i \ X_i \ M\{ \sum_{j=1}^{N} U_j \ X_j \ \} \ dx$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} m_{ij} \ U_i \ U_j \ . \qquad (2.2.20)$$

The partial derivatives of N and D with respect to U_i, and the symmetric condition of the coefficients, yields

$$\frac{\partial N}{\partial U_r} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(k_{ij} \frac{\partial U_i}{\partial U_r} U_j + k_{ij} \frac{\partial U_j}{\partial U_r} U_i\right)$$

$$= \sum_{j=1}^{N} k_{rj} \ a_j, \ r=1,2, \ - \ - \ -, \ N \ . \qquad (2.2.21)$$

In similar fashion, one obtains

$$\frac{\partial D}{\partial U_j} = 2 \sum_{i=1}^{N} m_{ji} \ U_i, \quad j=1,2, \ - \ - \ -, \ N \ . \qquad (2.2.22)$$

The substitution of Equations (2.2.21) and (2.2.22) into (2.2.16) then leads to

$$\sum_{j=1}^{N} (k_{ij} - w^2 \ m_{ij}) \ U_j = 0, \ i=1,2,- \ - \ -,N \qquad (2.2.23)$$

or

$$([k] - w^2 \ [m]) \ \{U\} = \{0\} \qquad (2.2.24)$$

where [k] and [m] are $N \times N$ symmetric matrices. The eigenfunctions associated with the estimated eigenvalue, w, are then determined by introducing the coefficient U_i into Equation (2.2.10) as follows:

$$w_n = \sum_{i=1}^{N} U_i \ X_i \qquad (2.2.25)$$

2.2.3 Galerkin's Method

Galerkin's method [9,22] seeks approximate solution of boundary value problems with a series of comparison functions which satisfy all the boundary conditions and possess derivatives of order at least twice that appearing in the strain energy expression. The error can be determined by the substitution of a series of comparison functions into the differential equation, with the condition that the integral of the weighted error over the domain be zero. Then, an eigenvalue problem for an N-degree-of-freedom system, associated with N series of comparison functions, can be represented as.

$$L[w] = \lambda M[w] , \qquad (2.2.26)$$

where L and M are, respectively, self-adjoint linear, homogeneous operators of the orders 2p and 2q. In general, the function w is subjected to boundary conditions which do not depend on the eigenvalue λ. The solutions of eigenvalue problem can be assumed in the form

$$w_n = \sum_{i=1}^{N} U_i X_i , \qquad (2.2.27)$$

where U_i are coefficients to be determined and X_i are comparison functions. The introduction of Equation (2.2.27) into Equation (2.2.26) gives the error

$$e = L[w_n] - \lambda M[w_n] , \qquad (2.2.28)$$

where λ is the estimate of the eigenvalue, λ.

The representation of the condition that the weighted error integrated over the domain is zero can be written

$$\int e X_i \, dx = 0, \ i=1,2, - - -, N . \qquad (2.2.29)$$

Then, let

$$\int X_j \, L[w_n] \, dx = \sum_{i=1}^{N} X_i \int X_j \, L[X_i] \, dx \qquad (2.2.30)$$

$$= \sum_{i=1}^{N} K_{ri} \, a_i \ , \ r=1,2,- - -,N , \qquad (2.2.31)$$

where the coefficients, K_{ri}, are symmetric,

$$K_{ri} = K_{ir} = \int X_r \, L[X_i] \, dx, \ r=1,2,- - -,N , \qquad (2.2.32)$$

since L is self-adjoint.

Similarly, one obtains

$$\int X_i \, M[w_n] \, dx = \sum_{j=1}^{N} m_{ij} \, a_j, \ i=1,2,- - -,N , \qquad (2.2.33)$$

where the coefficients, m_{ij}, are given by

$$m_{ij} = m_{ji} = \int X_i \, M[X_j] dx \qquad (2.2.34)$$

and are symmetric because M is self-adjoint. Equations (2.2.31) through (2.2.34) yield

$$\sum_{i=1}^{N} (K_{ij} - \lambda \, m_{ij}) a_j = 0, \ i=1,2,- - -,N , \qquad (2.2.35)$$

which is called Galerkin's equation, representing an eigenvalue problem for an N-degree-of-freedom system. This result is similar to Equation (2.2.24) obtained by the Rayleigh-Ritz method. When a series of comparison functions are used, rather than the admissible functions used in the Rayleigh-Ritz method, results of both the Galerkin method and the Rayleigh-Ritz method are identical.

2.2.4 Assumed-Mode Method

This method assumes a solution of boundary value problems in the form

$$y_N(x,t) = \sum_{i=1}^{N} X_i q_i(t) \quad , \qquad (2.2.36)$$

where X_i are admissible functions satisfying the geometric boundary conditions and q_i are the generalized coordinates. The substitution of Equation (2.2.36) into expressions for the kinetic energy, T, and the potential energy, V, and the application of Lagrange's equations yield the equations of motion for the N-degree-of-freedom system. The functions, $X_i(x)$, represent the displacement shape for the entire structure under consideration. They must form a linearly independent set.

The kinetic energy expression, T(t), and the potential energy expression, V(t), can be written

$$T(t) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} m_{ij} \, \dot{q}_i(t) \, \dot{q}_j(t) \qquad (2.2.37)$$

and

$$V(t) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} k_{ij} \, q_i(t) \, q_j(t) \quad , \qquad (2.2.38)$$

where the mass components, m_{ij}, depend upon the mass distribution of the system, the selected admissible functions, X_i, and the stiffness components, k_{ij}, depend upon the stiffness properties of the system, and the admissible functions X_i and its derivatives.

The application of Lagrange's equation for a conservative system of Equations (2.2.37) and (2.2.38), leads to the equation of motion,

$$\sum_{j=1}^{N} m_{ij}\, \ddot{q}_j + \sum_{j=1}^{N} k_{ij}\, q_j = 0, \quad i=1,2,- - -,N \qquad (2.2.39)$$

From the assumption that the dynamic response of the system is harmonic motion, Equation (2.2.39) leads to the representation of the eigenvalue problem in the form

$$\sum_{j=1}^{N} (k_{ij} - w^2\, m_{ij})\, q_j = 0, \quad i=1,2,- - -,N \qquad (2.2.40)$$

and its matrix form for Equations (2.2.39) and (2.2.40) can be written, respectively,

$$[m]\{\ddot{q}\} + [k]\{q\} = \{0\} \qquad (2.2.41)$$

and

$$([k] - w^2[m])\{q\} = \{0\}\ . \qquad (2.2.42)$$

Equation (2.2.40) then has the same form that Galerkin's method expressed in Equation (2.2.35).

2.2.5 Summary

Three common methods for discretizing a continuous parameter system have been compared. In summary, Galerkin's method bears a result identical to the

Rayleigh-Ritz method when the latter approach is used in a minimizing sequence, with a series of comparison functions in place of admissible functions. In some instances, the Rayleigh-Ritz method may be considered as a special case of applying the assumed-mode method [22]. Moreover, Galerkin's method leads to results identical to the assumed-mode method, but with differences in approach.

Given its convenient approach, the assumed-mode method has been used to develop the equations of motion of the physical model presented in section 2.1. The introduction of kinetic energy, potential energy, and generalized forces into Lagrange's equations may be expressed in the form

$$\frac{d}{dt}\left(\frac{\partial T}{\partial q_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i, \quad i=1,2,- - -,N \ , \quad (2.2.43)$$

which leads to a system of linear, ordinary differential equations describing the motion of the manipulator. In Equation (2.2.43), T(t) and V(t) are, respectively, kinetic energy and potential energy, Q_i represents generalized forces, and the generalized coordinates are $q_i(t)$.

2.3 Mathematical Modeling of The Physical Model Using the Assumed-Mode Method

The fundamental approach of the assumed-mode method discussed in section 2.2 is employed to obtain the equations of motion of the N-degree-of-freedom system of the physical model depicted in Figure 2.2. With the assumptions given for the proposed model in section 2.1, kinetic energy, T(t), and potential energy, V(t), can be determined. From the mechanics of materials, one obtains

$$\alpha \approx \frac{\partial y}{\partial x}$$
$$M_{x1} = EI \frac{\partial \alpha}{\partial x} \qquad (2.3.1)$$

where EI is the flexural rigidity of the beam.

The kinetic energy can be written

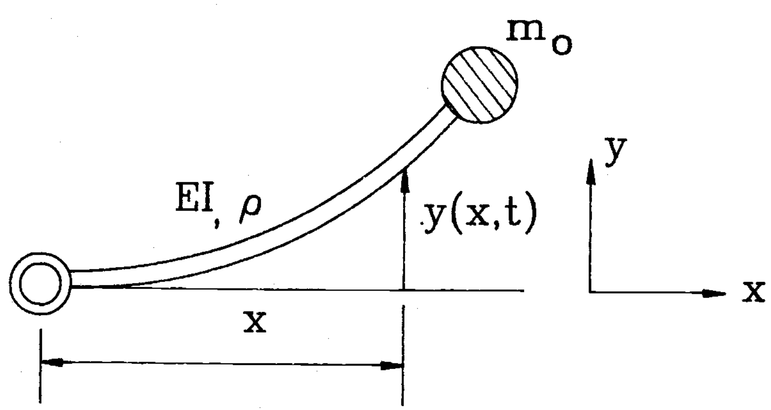
$$T(t) = \frac{1}{2} \int_0^L \rho A \; \{v(x,t)\}^2 \; dx$$
$$+ \frac{1}{2} m_o \; \{v(L,t)\}^2 \; , \qquad (2.3.2)$$

where y(x,t) is the velocity of the infinitesimal element of the beam and m is the payload at the tip of the beam. The velocity v(x,t) can be approximated in the form

$$v(x,t) \approx x \frac{\partial \theta(t)}{\partial t} + \frac{\partial y(x,t)}{\partial t} \; . \qquad (2.3.3)$$

The potential energy, V(t), can be obtained in the form

$$V(t) = \frac{1}{2} \int_0^L EI \; \{\frac{\partial^2 y(x,t)}{\partial x^2}\}^2 \; . \qquad (2.3.4)$$



EI : Flexural rigidity of the beam

ρ : Density of the beam

L : Cross-sectional area

m_o : Tip mass

Figure 2.2 Physical Model

The deflection, $y(x,t)$, is assumed by

$$y(x,t) = \sum_{i=1}^{N} X_i(x)\, q_i(t) \ , \tag{2.3.5}$$

where X_i are the admissible functions which satisfy the geometric conditions

$$X_i(0) = \left.\frac{\partial X_i}{\partial x}\right|_{x=0} = 0 \tag{2.3.6}$$

since

$$y(0,t) = \left.\frac{\partial y(x,t)}{\partial x}\right|_{x=0} = 0 \tag{2.3.7}$$

for all t.

By the substitution of Equations (2.3.5) and (2.3.6) into Equations (2.3.2) and (2.3.4), kinetic energy can be written

$$\begin{aligned} T = {} & \frac{1}{2} \left[\int_0^L \rho A x^2\, dx + m_o L^2\right] \{\dot{\theta}(t)\}^2 \\ & + \sum_{i=1}^{N} \left[\left\{\int_0^L \rho A x\, X_i(x)\, dx + m_o L\, X_i(L)\right\} \dot{q}_i(t)\, \dot{\theta}(t)\right] \\ & + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[\left\{\int_0^L \rho A x\, X_i(x) X_j(x) dx \right.\right. \\ & \left.\left. + m_o L\, X_i(L)\, X_j(L)\right\} \dot{q}_i(t)\, \dot{q}_j(t)\right] \end{aligned} \tag{2.3.8}$$

or

$$\begin{aligned} T = {} & \frac{1}{2} a\{\dot{\theta}(t)\}^2 + \sum_{i=1}^{N} b_i\, \dot{q}_i(t)\, \dot{\theta}(t) \\ & + \sum_{i=1}^{N} \sum_{j=1}^{N} m_{ij}\, \dot{q}_i(t)\, \dot{q}_j(t) \ , \end{aligned} \tag{2.3.9}$$

where

$$a = \int_0^L \rho A\, x^2\, dx + m_o\, L^2 \;, \qquad (2.3.10)$$

$$b_i = \int_0^L \rho A x\, X_i(x)\, dx + m_o\, L X_i(L) \;, \qquad (2.3.11)$$

and

$$m_{ij} = \int_0^L \rho A X_i(x) X_j(x)\, dx + m_o X_i(L)\, X_j(L) \;. \qquad (2.3.12)$$

Similarly, potential energy can be modified in the form

$$V(t) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} k_{ij}\, q_i(t)\, q_j(t) \;, \qquad (2.3.13)$$

where

$$k_{ij} = \int_0^L EI\, X_i''(x)\, X_j''(x)\, dx \;. \qquad (2.3.14)$$

Substitution of Equations (2.3.9) and (2.3.13) into Lagrange's equation yields the discrete equations of motion in the form

$$[m]\, \{\ddot{q}(t)\} + [k]\, \{q(t)\} = -\{b\}\, \ddot{\theta}(t) \;, \qquad (2.3.15)$$

where [m] and [k] are N × N matrices and {b} is the N × 1 matrix, the components of which are expressed in Equation (2.3.11). The generalized coordinate matrix, {q}, is then determined from these equations of motion.

Each admissible function, $X_i(x)$, in Equation (2.3.5) must be a continuous function of x, and its first and second derivative with respect to x must be continuous since the strain energy expression contains $y''(x,t)$. The notation (") indicates the second derivative of the function with respect to x. Even though it is not necessary that the functions X_i satisfy the natural boundary conditions, it is possible to use comparison functions that satisfy both the geometric and the natural boundary conditions. In this study, the functions $X_i(x)$ have been selected as the exact modes of a cantilever beam of the form

$$X_i(x) = (\sinh z_i + \sin z_i)[\cos(z_i x/L) - \cosh(z_i x/L)] - (\cosh z_i + \cos z_i)[\sin(z_i x/L) - \sinh(z_i x/L)] \quad i=1,2,---,N, \tag{2.3.16}$$

where z_i are solutions of the characteristic equation of a cantilever beam with a mass, m_o, at the tip, i.e.,

$$1 + h(\cos z_i \sinh z_i - \sin z_i \cosh z_i) + \cos z_i \cosh z_i = 0, \quad i=1,2,---,N, \tag{2.3.17}$$

and where

$$h = \frac{m_o}{\rho AL} \quad . \tag{2.3.18}$$

The results of the substitution of Equation (2.3.16) into Equations (2.3.11), (2.3.12), and (2.3.14) are presented in Appendix A. The dynamic response of the system represented in Equation (2.3.15) is obtained by letting

$$b_i = 0, \quad i=1,2, ---, N \tag{2.3.19}$$

with the harmonic functions, q_i, in the form

$$\{q(t)\} = \{u\}_i \cos(wt-\alpha), \quad i=1,2,---,N, \tag{2.3.20}$$

where $\{u\}_i$ is a scalar vector. This leads Equation (2.3.15) into the eigenvalue problem,

$$\{[k] - w_i^2[m]\}\{u\}_i = \{0\}, \quad i=1,2,- - -,N \, . \qquad (2.3.21)$$

For non-trivial solutions of Equation (2.3.21), it is necessary to satisfy

$$\det \{[k] - w_i^2[m]\} = 0 \, , \qquad (2.3.22)$$

which is termed the characteristic equation. Equation (2.3.22) yields the eigenvalues or squared natural frequencies, w_i^2, and the corresponding eigenvectors or natural mode, $\{u\}_i$. If the eigenvalues are ordered from the lowest to the highest,

$$0 \leq w_1^2 \leq w_2^2 \leq - - - \leq w_i^2 \leq w_N^2 \, , \qquad (2.3.23)$$

then the modal matrix can be written

$$[U] = [u_1 \; u_2 \; - - - \; u_N] \, , \qquad (2.3.24)$$

where $u_i = \{u\}_i$.

The orthogonality property may be expressed as

$$\{u\}_i^T \, [m] \, \{u\}_j = \begin{cases} 0, & \text{if } i \neq j \\ M_i, & \text{if } i = j \end{cases} \qquad (2.3.25)$$

and

$$\{u\}_i^T \, [k] \, \{u\}_j = \begin{cases} 0, & \text{if } i \neq j \\ K_i, & \text{if } i = j \end{cases} \qquad (2.3.25)$$

The i^{th} and j^{th} modes are said to be orthogonal with respect to the mass and stiffness matrices. This property yields an important procedure to transform the equations of motion in the generalized coordinates into the description in the principal coordinates. It also generates the modal mass matrix, [M], and the modal stiffness matrix, [K], in the form of a diagonal matrix by post- and pre-multiplication of the modal matrix [U] and its transpose matrix to, respectively, the mass matrix, [m], and the stiffness matrix, [k], as follows:

$$[M] = [U]^T[m][U] = \text{diag}(M_1\ M_2\ -\ -\ -\ M_N) \quad (2.3.26)$$

and

$$[K] = [U]^T[k][U] = \text{diag}(K_1\ K_2\ -\ -\ -\ K_N)\ . \quad (2.3.27)$$

The dynamic response of the equation of motion expressed by Equation (2.3.15) can be determined by the normal-mode, using the properties expressed in Equations (2.3.26) and (2.3.27) by transforming Equation (2.3.15) in generalized coordinates into principal coordinates. If the principal coordinates have a relationship in the form

$$\{q(t)\} = [U]\{g(t)\}\ , \quad (2.3.28)$$

where $\{g(t)\}$ are the principal coordinates, the equations of motion in the principal coordinates can be represented in the form

$$[M]\{\ddot{g}(t)\} + [K]\{g(t)\} = \{B(t)\}\ , \quad (2.3.29)$$

where

$$[M] = [U]^T[m][U] = \text{modal mass matrix}\ , \quad (2.3.30)$$

$$[K] = [U]^T[k][U] = \text{modal stiffness matrix}\ , \quad (2.3.31)$$

and

$$\{B\} = [U]^T\{b\}\ddot{\theta}(t) = \text{modal force vector}\ . \quad (2.3.32)$$

Since the modal mass matrix and the modal stiffness matrices are diagonal, Equation (2.3.29) can be rewritten as N uncoupled equations,

$$M_i \ddot{g}_i(t) + K_i g_i(t) = B_i(t), \quad i=1,2,- - -,N \quad ,(2.3.33)$$

where M_i and K_i are given by Equations (2.3.26) and (2.3.27) and $B_i(t)$ are obtained by

$$B_i(t) = \{u\}_i^T \{b_i\} \ddot{\theta}(t) \ . \qquad (2.3.34)$$

The initial conditions in principal coordinates can then be determined by

$$y(x,0) = \sum_{i=1}^{N} X_i(x)\ q(0) = \{X_i\}^T [U]\ \{g(0)\} \ , \qquad (2.3.35)$$

$$\dot{y}(x,0) = \sum_{i=1}^{N} X_i(x)\ \dot{q}(0) = \{X_i\}^T [U]\ \{\dot{g}(0)\} \ , \qquad (2.3.36)$$

or

$$\{q(0)\} = [U]\{q(0)\} \qquad (2.3.37)$$

and

$$\{\dot{q}(0)\} = [U]\{\dot{g}(0)\} \ . \qquad (2.3.38)$$

The multiplication of these equations by $[U]^T$ [m] leads to

$$[U]^T [m] \{q(0)\} = [U]^T [m] [U] \{g(0)\}$$
$$= [M]\{g(0)\} \qquad (2.3.39)$$

and

$$[U]^T [m] \{\dot{q}(0)\} = [U]^T [m] [U] \{\dot{g}(0)\}$$
$$= [M]\{\dot{g}(0)\} \ . \qquad (2.3.40)$$

Since the matrices [M] and [K] are diagonal, the modal initial conditions are given by

$$g_i(0) = (\frac{1}{M_i}) \{u\}_i^T [m] \{q(0)\}, \ i=1,2,- - -,N \tag{2.3.41}$$

$$\dot{g}_i(0) = (\frac{1}{M_i}) \{u\}_i^T [m] \{\dot{q}(0)\}, \ i=1,2,- - -,N \tag{2.3.42}$$

The use of the Duhamel integral with Equations (2.3.33), (2.3.41) and (2.3.42) yields the ith modal response in the form

$$g_i(t) = g_i(0) \cos(w_i t) + (\frac{1}{w_i}) \dot{g}_i(0) \sin(w_i t)$$

$$+ (\frac{1}{M_i w_i}) \int_0^t B_i(t) \sin w_i(t-\tau) \, d\tau \ , \tag{2.3.43}$$

where w_i is given by Equation (2.3.23). The final result can be obtained by introducing Equations (2.3.43) and (2.3.28) into Equation (2.3.5) in the form

$$y(x,t) = \{X(x)\} [U]\{g(t)\} \ , \tag{2.3.44}$$

where

$$\{X\} = \{X_1 \ X_2 \ - - - \ X_N\} \tag{2.3.45}$$

and

$$\{g\} = \{g_1 \ g_2 \ - - - \ g_N\}^T \ . \tag{2.3.46}$$

The assumed-mode method, in conjunction with mode-superposition method, can be used for the response of a system with a special type of damping called modal damping. The equations of motion of this system in generalized coordinates can be expressed in the form

$$[m]\{\ddot{q}(t)\} + [c]\{\dot{q}(t)\} + [k]\{q(t)\}$$
$$= \{p(t)\} , \qquad (2.3.47)$$

where the matrices [m] and [k] are identical to those in Equation (2.3.15), {p(t)} is the generalized force matrix, and [c] is the system damping matrix.

If the eigenvectors, $\{u\}_i$, and the eigenvalues, $w_i{}^2$, are obtained from the relationships expressed in Equations (2.3.21) and (2.3.22), and if the system damping matrix [c] satisfies the condition

$$\{u\}_i{}^T\ [c]\ \{u\}_j = \begin{cases} 0, & \text{if } i \neq j \\ C_i, & \text{if } i = j \end{cases} , \qquad (2.3.48)$$

then the type of damping is classified as modal damping. Therefore, the equation of motion of a system with the modal damping in principal coordinates can be written

$$M_i\ \ddot{g}_i(t) + C_i\ \dot{g}_i(t) + K_i\ g_i(t)$$
$$= P_i(t),\ i=1,2,- - -,\ N \qquad (2.3.49)$$

or

$$\ddot{g}(t) + 2\ \zeta_i\ w_i\ \dot{g}_i(t) + w_i{}^2\ g_i(t)$$
$$= \frac{1}{M_i}\ P_i(t)\ i=1,2,- - -,\ N , \qquad (2.3.50)$$

where

$$K_i = \{u\}_i{}^T\ [k]\ \{u\} , \qquad (2.3.51)$$
$$M_i = \{u\}_i{}^T\ [m]\ \{u\} , \qquad (2.3.52)$$

and

$$\zeta_i = \frac{C_i}{2M_i w_i} = \left(\frac{1}{2M_i w_i}\right) \{u\}_i^T [c] \{u\}_i \ . \qquad (2.3.53)$$

In a practical sense, the damping matrix can be approximated in the form of Equation (2.3.48). Further study of the application of adaptive control laws to a flexible manipulator for which damping can be approximated in the above form will be invaluable. In this study, it is assumed that the system damping matrix can be represented in the form of proportional damping, termed Rayleigh damping, defined by

$$[c] = c_o[m] + c_1[k] \ , \qquad (2.3.54)$$

where c_o and c_1 are constants chosen to produce specified modal factors for two given modes.

By the introduction of the orthogonality conditions

$$\{u\}_i^T [m] \{u\}_j = M_i \ \delta_{ij} \qquad (2.3.55)$$

and

$$\{u\}_i^T [k] \{u\}_j = K_i \ \delta_{ij} = w_i^2 \ M_i \ \delta_{ij} \ , \qquad (2.3.56)$$

where δ_{ij} is the kroneker delta, one obtains

$$\{u\}_i^T[c] \{u\}_j = C_i$$
$$= (c_o + c_1 w_i^2) \ M_i \ \delta_{ij} \ . \qquad (2.3.57)$$

The comparison of Equation (2.3.57) with Equation (2.3.53) yields

$$\zeta_i = \frac{1}{2} \left(\frac{c_o}{w_i} + c_1 \ w_i\right) \ . \qquad (2.3.58)$$

For the damping factors to be specified for all modes of interest, the following procedure can be utilized. The damping matrix in generalized coordinates can be written

$$[c] = ([U]^T)^{-1}\ [C]\ [U]^{-1}\ . \qquad (2.3.59)$$

From the relationship expressed in Equation (2.3.26), one obtains

$$[U]^{-1} = [M]^{-1}[U]^T[m] \qquad (2.3.60)$$

and

$$([U]^{-1})^{-1} = [m][U][M]^{-1}\ . \qquad (2.3.61)$$

The substitution of Equations (2.3.60) and (2.3.61) into Equation (2.3.59) yields

$$[c] = ([m][U][M]^{-1})\ [C]\ ([M]^{-1}[U]^T[m]) \qquad (2.3.62)$$

or

$$[c] = \sum_{i=1}^{N} (\frac{2\zeta_i w_i}{M_i})\ ([m]\{u\}_i)\ ([m]\{u\}_i)^T\ . \qquad (2.3.63)$$

The Equation (2.3.63) can be truncated to a limited number of the lower-frequency modes as follows:

$$[c] = \sum_{i=1}^{Nc} (\frac{2\zeta_i w_i}{M_i})\ ([m]\{u\}_i)\ ([m]\{u\}_i)^T\ . \qquad (2.3.64)$$

The Equation (2.3.64) can then be further modified to provide damping in the modes higher than N_c in the form

$$[c] = a_1[k]$$

$$+\sum_{i=1}^{Nc^{-1}} (\frac{2\zeta_i' w_i}{M_i})\ ([m]\{u\}_i)\ ([m]\{u\}_i)^T\ , \qquad (2.3.65)$$

where

$$a_1 = \frac{2\zeta_{Nc}}{w_{Nc}} \tag{2.3.66}$$

and

$$\zeta_i' = \zeta_i - \zeta_{Nc}(\frac{w_i}{w_{Nc}}) \ . \tag{2.3.67}$$

Therefore, the solutions for Equations (2.3.49) or (2.3.50) can be expressed in the form

$$\begin{aligned} g_i(t) = (\frac{1}{M_i w_{di}}) \int_0^t & P_i(\tau) \exp\{-\zeta_i w_i(t-\tau)\} \\ & \times \sin\{w_{di}(t-\tau)\}d\tau \\ & + g_i(0) \exp\{-f_i w_i t\} \cos\{w_{di} t\} \\ & + (\frac{1}{w_{di}})\{\dot{g}_i(0)\} \\ & + \zeta_i w_i g_i(0)\} \exp\{-\zeta_i w_i t\} \sin\{w_{di} t\} \ , \end{aligned} \tag{2.3.68}$$

where

$$w_{di} = w_i \sqrt{1 - \zeta_i^2} \ . \tag{2.3.69}$$

The advantage of this procedure is that the state variables in Equations (2.3.15) or (2.3.49) can be represented, respectively, as the modal state variables are in Equations (2.3.29) or (2.3.49) and the control laws can be directly applied. The measurement of system state variables can be converted either into generalized coordinates or into in principal coordinates from, respectively, the relationships in Equations (2.3.5) or Equation (2.3.28). In this study, dynamic strain gauges are considered as the feedback sensors located along the manipulator. In such a case, the measured state variables can be expressed either in the form

$$\{q(t)\} = [X'']^{-1} \{y(t)''\}_m \tag{2.3.70}$$

in generalized coordinates, or in the form

$$\{g(t)\} = [U]^{-1} [X'']^{-1} \{y(t)''\}_m \tag{2.3.71}$$

in principal coordinates, where $[X'']^{-1}$ indicates the inverse of the matrix $[X'']$ and $\{y(t)\}_m$ are the values of the gauge readings at the positions of x_m, $m=1,2,- - -,N$ and time t. The matrix $[X'']$ consists of

$$[X''] = [X_1'' \ X_2'' \ - - - \ X_N]^T \ , \tag{2.3.72}$$

where

$$X_I'' = \frac{\partial^2}{\partial x^2} [X_1(x_i)X_2(x_i) \ - - - \ X_N(x_i)] \tag{2.3.73}$$

and where x_i are the locations of the dynamic strain gauges in x coordinates. The following chapter will include details of the use of this property.

III. MODEL REFERENCE ADAPTIVE SYSTEM DESIGN PROBLEM AND GENERAL DISCUSSION

The performance of conventional control schemes is limited when the parameters of the system under consideration are poorly known, when there are significant variations in the system parameters, or when the model is constructed imperfectly. The adaptive control system has been developed to overcome such difficulties. Among the choices of adaptive control systems, the model reference adaptive system, as modified for application to the control of a flexible manipulator, is discussed in this chapter.

3.1 Model Reference Adaptive System Theory

The adaptive control system is characterized by a property allowing the system to self-redesign or self-adjust its system according to changes in enviromental conditions. Among the variety of adaptive control systems, the model reference adaptive system implements self-adjusting adaptations by direct comparisons between the outputs of the reference model and that of the adjustable system. In effect, the adaptation mechanism adjusts the parameters of the adjustable system so that the output differences finally vanish. Figure 3.1 shows a typical representation of the basic model reference adaptive system.

Changes in environment, unpredictable variations in parameters, or imperfect modeling causes output errors between the reference model and the adjustable system. The adaptation mechanism measures these differences, adjusting the parameters of the adjustable system. The main work of the model reference adaptive system application is to design an adaptation mechanism with the ability to tune the adjustable system, based on comparison of the reference model and adjustable system outputs. The whole system must be stable during this operation.

The stability of the model reference adaptive system can be achieved by the application of various methods, including local parametric optimization theory, Lyapunov redesign, or hyperstability and positivity concepts. In this study, the latter approach has been used for the design of the model reference adaptive system. The balance of this chapter is concerned with mathematical descriptions of the model reference adaptive system and its design with reference to hyperstability and positivity concepts. The extension of its application to the control of a flexible manipulator is

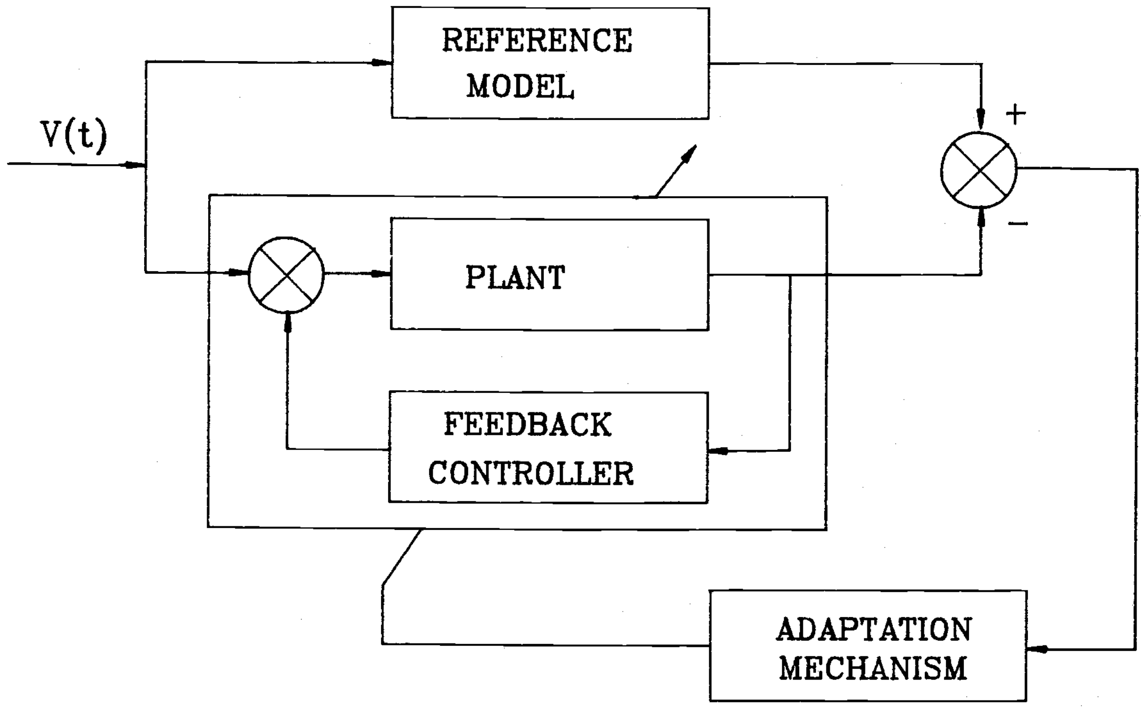


Figure 3.1 Basic Structure of Model Reference Adaptive System

then developed. For a review and fundamental theoretical study of the model reference adaptive system, see Landau [18,19] and Leininger [20].

3.2 Model Reference Adaptive System Representation

Classification methods for model reference adaptive systems were introduced by Landau [19]. Among these classifications, the parallel model reference adaptive system will be considered since its fundamental properties can be extended easily to other configurations and various applications. Figure 3.2 indicates the basic structure of the parallel model reference adaptive system. To describe the model reference adaptive system in the format of a state-variable description, the reference model can be given by

$$\{\dot{r}\} = [A_r]\{r\} + [B_r]\{v\}, \quad \{r(0)\} = \{r_o\} \ , \qquad (3.2.1)$$

where

$\{r(t)\}$ = the N-dimensional model reference state vector, and

$\{v\}$ = the input vector where $[A_r]$ and $[B_r]$ are system matrices, which are, respectively, constant N×N- and N×M-dimensional matrices.

The reference model is assumed to be stable and completely controllable [28].

The adjustable system can be represented in the form

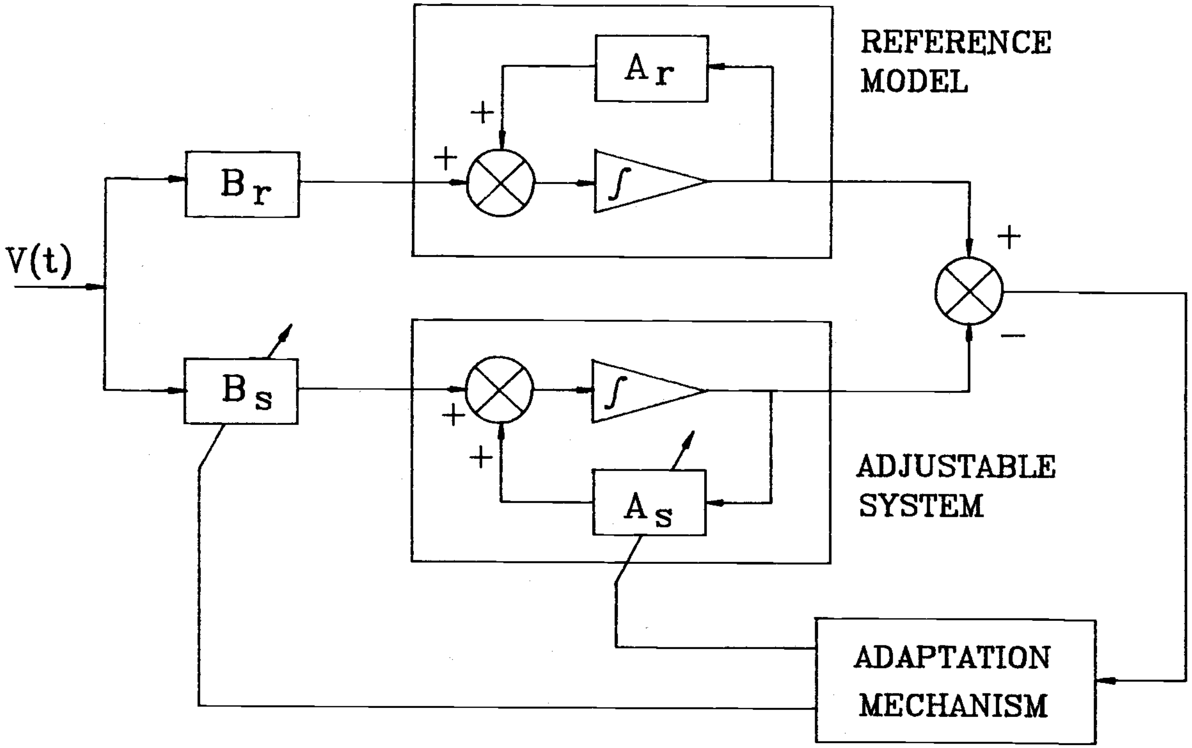


Figure 3.2 Parallel Model Reference
Adaptive System

$$\{\dot{s}\} = [A_s]\{s\} + [B_s]\{v\}, \{s(0)\} = \{s_o\}, [A_s(0)] = [A_{so}], [B_s(0)] = [B_{so}] , \quad (3.2.2)$$

where

$\{s(t)\}$ = the N-dimensional adjustable system state vector and

$[A_s]$ and

$[B_s]$ = time varying matrices which, respectively, have the identical dimension of $[A_r]$ and $[B_r]$ in Equation (3.2.1).

When the generalized state error vector, $\{e\}$, is defined by

$$\{e\} = \{r\} - \{s\} , \quad (3.2.3)$$

the adaptation mechanisms use the values of $\{e\}$ to construct the parametric matrices $[A_s]$ and $[B_s]$. The principal objective of the design of the adaptation law is that the parametric matrices, $[A_s]$ and $[B_s]$, are properly adjusted in order that the error vector, $\{e\}$, approaches zero for an arbitrary input, $\{v\}$. When no difference in parameters initially exists between the reference model and the adjustable system, the parametric matrices $[A_s]$ and $[B_s]$ should remain in their original state. Moreover, the adaptation mechanism must be capable of memorizing the values of the parameters which lead the error vector, $\{e\}$, to zero, i.e., the adaptation mechanism must contain an integral component whose values are dependent upon not only $\{e(t)\}$ at current time, t, but also upon the values of $\{e(\tau)\}$ throughout past time, τ, when $\tau \leq t$.

In other words, the fundamental problem in the design of the reference model adaptive system is formulation of the adaptation mechanism allowing the elimination of an unknown initial difference at $t = t_o$, between the reference model and the adjustable system parameters. This condition is represented in the form of a perfect asymptotic adaptation as follows:

$$\lim_{t\to\infty}\{e(t)\} = \lim_{t\to\infty}[\{r(t)\} - \{s(t)\}] = \{0\} , \qquad (3.2.4)$$

$$\lim_{t\to\infty}\{[A_r] - [A_s]\} = [0] , \qquad (3.2.5)$$

and

$$\lim_{t\to\infty}\{[B_r] - [B_s]\} = [0] . \qquad (3.2.6)$$

Equations (3.2.5) and (3.2.6) can be rewritten

$$\lim_{t\to\infty}[A_s] = [A_r] \qquad (3.2.7)$$

and

$$\lim_{t\to\infty}[B_s] = [B_r] , \qquad (3.2.8)$$

respectively.

The adaptation law, which contains an integrator in the adaptation mechanism, can be expressed in the forms

$$[A_s(e,t)] = \int_0^t [R_1(e,t,\tau)]d\tau + [R_2(e,t)] + [A_{so}] \qquad (3.2.9)$$

and

$$[B_s(e,t)] = \int_0^t [S_1(e,t,\tau)]d\tau + [S_2(e,t)]$$

$$+ [B_{so}] , \qquad (3.2.10)$$

where $[R_1]$ and $[R_2]$ are N $\times$ N matrices and $[S_1]$ and $[S_2]$ are N $\times$ M matrices. The first term on the right hand sides of Equations (3.2.9) and (3.2.10) provide the memory for the adaptation mechanism and the second term indicates the elements of the adaptation mechanism which vanish when the error vector, {e}, becomes zero. Figure 3.3 shows the parallel model reference adaptive system in state-space representation.

The matrices of the adaptation mechanism, $[R_1]$, $[R_2]$, $[S_1]$, and $[S_2]$, must be determined in order that the system remain stable throughout entire operations and that the conditions represented by Equations (3.2.4), (3.2.5) and (3.2.6) are satisfied. Since the information necessary to implement these requirements is limited to the values of {e}, representation of the system in terms of the error vector {e} is required. Therefore, the equivalent feedback representation of the state error system is applicable to this implementation.

The substraction of Equation (3.2.2) from Equation (3.2.1) leads to

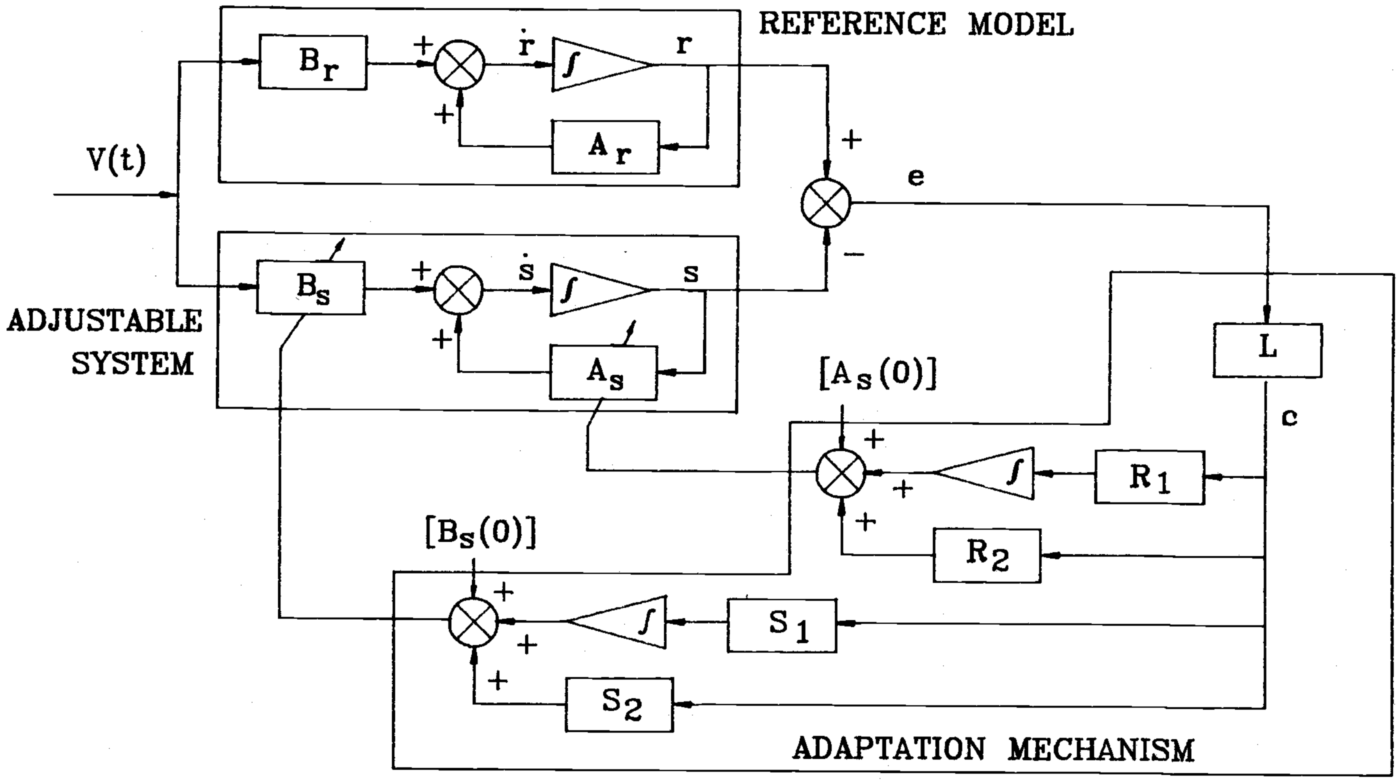


Figure 3.3 Parallel Model Reference Adaptive System
In Space-State Representation

$$\{\dot{e}\} = \{\dot{r}\} - \{\dot{s}\} = [A_r]\{r\} - [A_s]\{s\} + ([B_r]-[B_s])\{v\} \qquad (3.2.11)$$

or

$$\{\dot{e}\} = [A_r]\{e\} + ([A_r]-[A_s])\{s\} + ([B_r]-[B_s])\{v\} \ . \qquad (3.2.12)$$

Equation (3.2.12) is obtained by adding and substracting the term $[A_r]\{s\}$ on the right-hand side of Equation (3.2.11). Furthermore, the substitution of Equations (3.2.9) and (3.2.10) into Equation (3.2.12) transforms the description of the model reference adaptive system into equivalent representation of the system, as follows:

$$\{\dot{e}\} = [Ar]\{e\} + ([Ar]-[Aso]-[R1]'-[R2])\{s\} + ([B_r]-[B_{so}]-[S_1]'-[S_2])\{v\} \ , \qquad (3.2.13)$$

where

$$[R_1]' = \int_0^t [R_1(e,t,\tau)] \, d\tau \qquad (3.2.14)$$

and

$$[S_1]' = \int_0^t [R_1(e,t,\tau)] \, d\tau \ . \qquad (3.2.15)$$

Figure 3.4 represents the equivalent feedback representation of the state error system of the parallel model reference adaptive system. The equivalent system can be divided into two parts, characterized as linear time-invariant and non-linear time-varying. When the

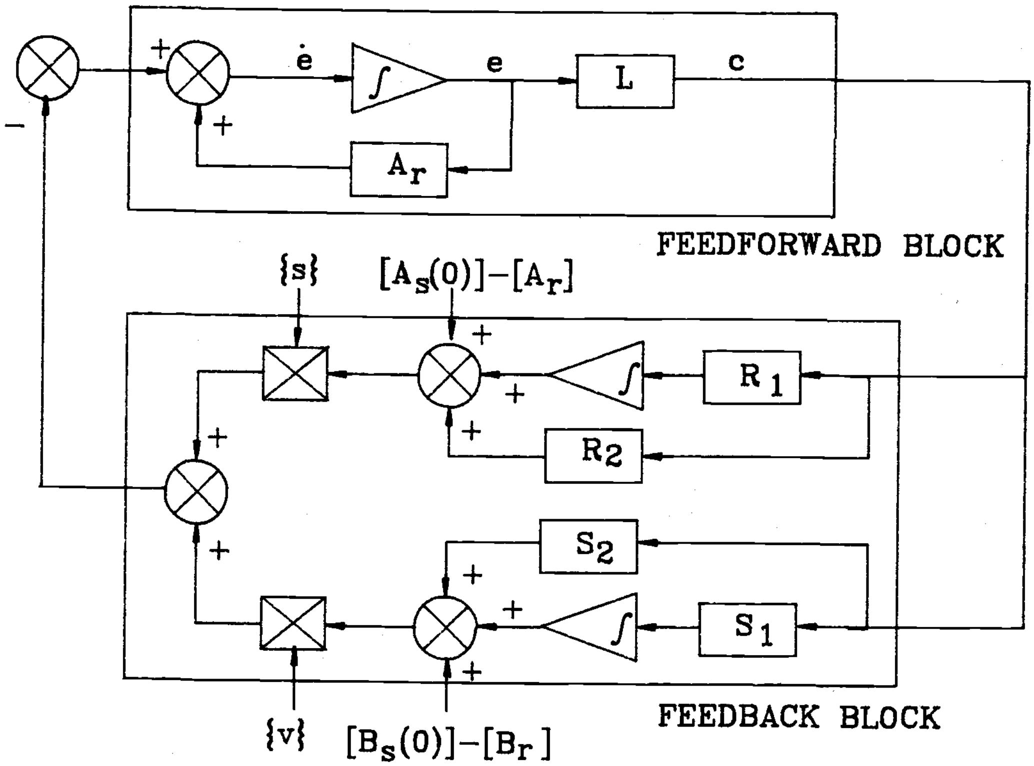


Figure 3.4 Equivalent Feedback Representation

matrix, $[A_r]$, is predetermined, processing the values of {e} through the linear compensator enhances assurance of the stability of the linear part. The adaptation mechanism then applies the values of {c}, which are obtained by the linear compensator, [L], in the relationship

$$\{c\} = [L]\{e\} \qquad (3.2.16)$$

rather than the direct application of {e}. The matrix gain, [L], must be determined, based on the stability requirements of the system. This process is discussed in greater detail in the following section.

The equivalent representation of the state error system can be modified by introducing Equation (3.2.16) as an element of the linear part, expressed as follows:

$$\{\dot{e}\} = [Ar]\{e\} + [I]\ \{W1\}\ , \qquad (3.2.17)$$

$$\{c\} = [L]\{e\}\ , \qquad (3.2.18)$$

and

$$\{W\} = -\{W1\} = ([R_1]'+[R_2]+[A_{so}]-[A_r])\{s\} + ([S_1]'+[S_2]+[B_{so}]-[B_r])\{v\}\ , \qquad (3.2.19)$$

where the matrix, {W}, indicates the output of the feedbackword block in Figure 3.4, in turn representing the Equations (3.2.17), (3.2.18), and (3.2.19).

In the following section, the method of designing the adaptation mechanism based upon the use of equivalent representation of the model reference adaptive system is discussed. Among the various methodological options, hyperstability and positivity concepts have been selected as the underlying design principles.

3.3 Model Reference Adaptive System Design Based on Hyperstability and Positivity Concepts

The design of the adaptation mechanism of the model reference adaptive system includes the fundamental requirement of system stability during operations. Of the three basic design options, i.e., the local parametric optimization method, the Lyapunov redesign method, and hyperstability theory, the hyperstability approach is the most useful method for the design of the model reference adaptive system.

The local parametric optimization approach synthesizes adaptive loops by the use of sensitivity functions [30,31]. The method poses difficulties due to time dependence and non-linearity of the model reference adaptive system. The Lyapunov redesign approach is limited because of its difficulties in extending the adaptation laws for a globally stable model reference adaptive system. However, a third approach, the hyperstability theory introduced by V. M. Popov [31], has been recognized as a succesful method by which the model reference adaptive system can incorporate a large family of adaptation laws and has been adapted for that purpose in this study.

In the previous section the model reference adaptive system was represented as an equivalent of the state error system, as expressed in Equations (3.2.17), (3.2.18), and (3.2.19). Figure 3.5 shows the system divided into two blocks, a linear time-invariant feedforward block and a non-linear, time-varying feedbackward block. If the feedback system is globally stable for all feedback blocks, satisfying the Popov integral inequality,

$$P(0,t1) \equiv \int_{0}^{t1} \{W_1\}^T \{c\} \, dt \geq p_o^2$$

$$\text{for all } t1 \geq 0 \ , \qquad (3.3.1)$$

where $\{c\}$ and $\{W_1\}$ are, respectively, input and output vectors of the feedback block and p_o^2 is a finite positive constant, then the feedback system is hyperstable and the feedfoward block is called a hyperstable block.

For hyperstable conditions, the transfer matrix of the feedforward block must satisfy the properties of a positive dynamic system. To solve the stability problem using the hyperstability approach, the original problem must first be cast as a stability problem related to the feedback system. This expression must have the ability to isolate one part for verification of the Popov integral inequality of Equation (3.3.1), while the remainder is used to verify a corresponding positivity condition assuring the hyperstability of the entire system [19].

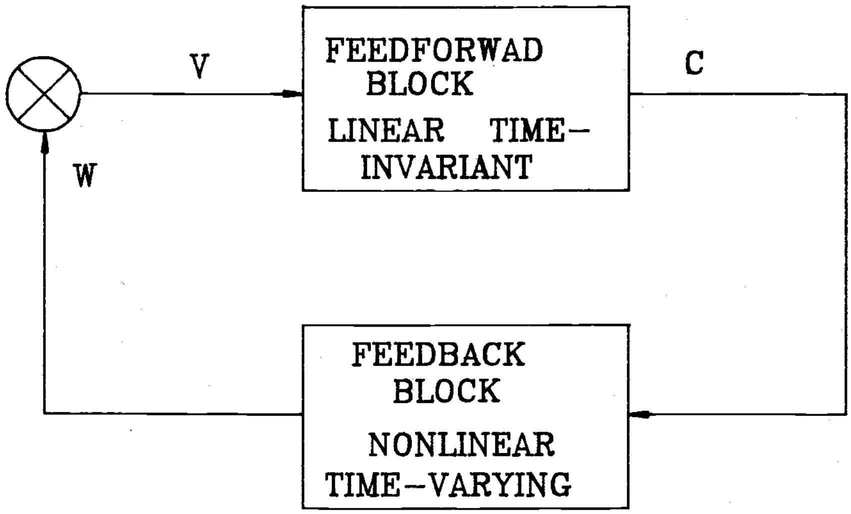


Figure 3.5 General Representation of Feedback System

The system which results, termed a positive dynamic system, may be mathematically defined in the following sequence [19,20,31], based on the concept and properties of positive dynamic systems.

3.3.1 Definition 1

A matrix, $[R(w)]$, of a complex variable, $h = a + ib$, is a Hermitian matrix if

$$[R(h)] = [R(\bar{h})]^T , \tag{3.3.2}$$

where $\bar{h}$ is the complex conjugate of h.

Some properties of a Hermitian matrix include the following:

1) $[R(h)]$ is a square matrix and its diagonal terms are real;
2) The eigenvalues of $[R(h)]$ are always real; and
3) The quadratic form $\{u\}^T[R(h)]\{\bar{u}\}$ is always real, where $\{u\}$ is any vector of complex components.

3.3.2 Definition 2

An $N \times N$ matrix, $[R(h)]$, of real rational functions of the complex variable, w, can be defined as positive real if:

1) All elements of $[R(h)]$ are analytic in $Re[h] > 0$;
2) Any purely imaginary pole, ib, of any element of $[R(h)]$ is a simple pole, and the associated

residue matrix of [R(h)] is a nonnegative definite Hermitian; and

3) The matrix, $[R(ib)] + [R(-ib)]^T$, is a positive semidefinite Hermitian for all real values of h, which are not poles of any element of [R(h)].

Alternatively, conditions 2 and 3 may be replaced by

4) The matrix $[R(h)] + [R(\bar{h})]^T$ is non-negative definite Hermitian in Re[h]>0.

3.3.3 <u>Definition 3</u>

An N × N matrix, [R(h)], of real rational functions is strictly positive real if all elements of $[R(ib)] + [R(-ib)]^T$ are positive definite Hermitian for all real w. The positivity of a continuous linear time-invariant system is expressed as

$$\{\dot{s}\} = [A]\{s\} + [B]\{v\} \tag{3.3.3}$$

and

$$\{c\} = [F]\{s\} + [G]\{v\} , \tag{3.3.4}$$

where {s} is an N-dimensional state vector, and {v} and {c} are, respectively, M-dimensional input and output vectors (see Definition 4).

3.3.4 <u>Definition 4</u>

Equations (3.3.3) and (3.3.4) are positive if the integral, P(0,t1), can be written in the form

$$P(0,t1) \equiv \int_0^{t1} \{c\}^T \{v\}\, dt = [\alpha(s)]_o^{t1}$$

$$+ \int_0^{t1} [\beta(s,v)]\, dt \geq -p_o{}^2 \text{ for all } t1 \geq 0 \quad ,(3.3.5)$$

where $[\beta(s,v)] \geq 0$ for all $\{s\} \in R^N$ and $\{v\} \in C^M$, and the functions $[\alpha(s)]$ and $[\beta(s,v)]$ are defined for all {s} and {v}.

In Equations (3.3.3) and (3.3.4), the pair, ([A],[B]), are assumed to be completely controllable and the pair, ([F],[A]), are completely observable. The square transfer matrix [T(h)] of the system can then be written

$$[T(h)] = [G] + [F]\,(h[I]-[A])^{-1}\,[B]\ . \qquad (3.3.6)$$

3.3.4.1 <u>Theorem</u>

The positivity properties may be expressed in various equivalent formulations to provide convenient application flexibility. The following theorem represents some of these equivalent property formulations for the system expressed in Equations (3.3.3) and (3.3.4).

1) Equations (3.3.3) and (3.3.4) are positive in conformity with Definition 2;

2) In Equation (3.3.6), [T(h)] is a positive real transfer matrix;

3) There exists a symmetric positive definite matrix, [P], a symmetric positive semidefinite

matrix, [D], and matrices [S] and [R] in order that

$$[P][A] + [A]^T[P] = -[D] , \tag{3.3.7}$$

$$[B]^T[P] + [S]^T = [C] , \tag{3.3.8}$$

$$[G] + [G]^T = [R] , \tag{3.3.9}$$

and

$$\begin{bmatrix} [D] & [S] \\ [S]^T & [R] \end{bmatrix} \geq [0] ; \tag{3.3.10}$$

4) There is the Kalman-Yakubovitch-Popov Lemma, in which a symmetric positive marix, [P], and matrices [K] and [L] exist in order that

$$[P][A] + [A]^T[P] = -[L][L]^T , \tag{3.3.11}$$

$$[B]^T[P] + [K]^T[L]^T = [F] , \tag{3.3.12}$$

and

$$[K]^T[K] = [G] + [G]^T ; \tag{3.3.13}$$

5) The Hermitian matrix $[Z(-h,h)] = [T(-h)]^T + [T(h)]$ is positive semidefinite for all $h = -ib$ in which $\det(ib[I] - [A]) \neq 0$;

6) Every solution, $\{s(s_o,v,t)\}$, of Equations (3.3.3) and (3.3.4) verifies the following equality:

$$\int_0^{t1} \{c\}^T\{v\}dt = \frac{1}{2}\{s(t1)\}^T[P]\{s(t1)\} - \frac{1}{2}\{s_o\}^T[P]\{s_o\} + \frac{1}{2}\int_0^{t1} (\{s\}^T[D]\{s\} + 2\{v\}^T[S]\{s\}+\{v\}^T[R][v\})dt , \tag{3.3.14}$$

and

7) For {s)0)} = {0}, for any input vector function, {y(t)}, and its corresponding solution, {s(0,v,t)}, or for the system in Equations (3.3.3) and (3.3.4), the following inequality is satisfied:

$$\int_0^{t1} \{c\}^T\{v\}dt \geq 0 \ . \qquad (3.3.15)$$

Proposition 3 of the theorem implies proposition 4. When [G] is [0] in Equation (3.3.4), the following lemma can be established.

3.3.4.2 Lemma 1

The linear time-invariant system,

$$\{\dot{s}\} = [A]\{s\} + [B]\{v\} \qquad (3.3.16)$$

and

$$\{c\} = [D]\{s\} \ , \qquad (3.3.17)$$

are positive and the transfer matrix,

$$[T(h)] = [D]^T(h[I]-[A])^{-1} \ [B] \ , \qquad (3.3.18)$$

is a positive real transfer matrix if, and only if, there exists a symmetric positive definite matrix, [P], and a symmetric positive semidefinite matrix, [D], in order that

$$[P][A] + [A]^T[p] = -[D] \qquad (3.3.19)$$

and

$$[B]^T[P] = [D] \ . \qquad (3.3.20)$$

An appropriate proposition of the theorem given in section 3.3.4.1 can be used for the test or for the construction of a positive system. Similarily, the positivity of the case of a linear, time-varying, multi-variable system given by

$$\{\dot{s}\} = [A(t)]\{s\} + [B(t)]\{v\} \qquad (3.3.21)$$

and

$$\{c\} = [F(t)]\{s\} + [G(t)]\{v\} \ , \qquad (3.3.22)$$

where {s} is an N-dimensional state vector and {v} and {c} are, respectively, M-dimensional input and ouput vectors, can be characterized as given by Definition 5.

3.3.5 <u>Definition 5</u>

The system given in Equations (3.3.21) and (3.3.22) is positive if the integral, P(0,t1), can be written

$$P(0,t1) \equiv \int_0^{t1} \{c\}^T \{v\}\, dt = [\alpha(s)]_0^{t1}$$

$$+ \int_0^{t1} [\beta(s,v)]\, dt \geq -p_o^2 \text{ for all } t1 \geq 0 \ , \qquad (3.3.23)$$

where $[\beta(s,v)] \geq 0$ for all $t1 \geq 0$. (3.3.24)

From direct extension of the results of the theorem given in section 3.3.4.1, two sufficient conditions for the positivity of the system in Equations (3.3.21) and (3.3.22) are as given in the lemma 2 (section 3.3.5.1).

3.3.5.1 <u>Lemma 2</u>

The system represented by Equations (3.3.21) and (3.3.22) is positive if there exists an asymmetric time-varying positive definite matrix, [P(t)], differential with respect to t, a symmetric time-varying semidefinite matrix, [D(t)], and matrices [S(t)] and [R(T)] in order that

$$[\dot{P}(t)] + [A(t)]^T[P(t)] + [P(t)][A(t)] = -[D(t)] \; , \qquad (3.3.25)$$

$$[B(t)]^T[P(t)] + [S(t)]^T = [C(t)] \; , \qquad (3.3.26)$$

$$[G(t)] + [G(t)]^T = [R(t)] \; , \qquad (3.3.27)$$

and

$$\begin{bmatrix} [D(t)] & [S(t)] \\ [S(t)]^T & [R(t)] \end{bmatrix} \geq [0] \; . \qquad (3.3.28)$$

3.3.5.2 <u>Lemma 3</u>

The system represented by Equations (3.3.21) and (3.3.22) is positive if every solution, $\{s(s_o,v,t)\}$, satisfies the following equality:

$$\int_0^{t1} \{c\}^T\{v\}dt = \frac{1}{2}\{s(t1)\}^T[P]\{s(t1)\} - \frac{1}{2}\{s_o\}^T[P(t)]\{s_o\} + \frac{1}{2}\int_0^{t1} (\{s\}^T[D(t)]\{s\} + 2\{v\}^T[S(t)]\{s\} + \{v\}^T[R][v\}) \, dt \; , \qquad (3.3.29)$$

with

$$[P(t)] > o, \begin{bmatrix} [D(t)] & [S(t)] \\ [S(t)]^T & [R(t)] \end{bmatrix} \geq [0] \quad \text{for all } t1 \geq 0 \; . \qquad (3.3.30)$$

The concept of the positive dynamic system is then applied to the design of the model reference adaptive system, with the incorporation of additional hyperstability theory definitions. A feedback system can be represented both by a feedforward block,

$$\{\dot{s}\} = [A]\{s\} + [B]\{v\} = [A]\{s\} - [B]\{w\} \; , \quad (3.3.31)$$

$$\{c\} = [F]\{s\} + [G]\{v\} = [F]\{s\} - [G]\{w\} \; , \quad (3.3.32)$$

and by a feedbackward block,

$$\{w\} = \{f(c,t,\tau)\}, \; \tau \leq t \; , \quad (3.3.33)$$

where {s} is the N-dimensional state vector of the feedforward block, and {v} and {c} are, respectively, M-dimensional input and output vectors of the feedforward block. The pair, ([A],[B]), are completely observable, and {f(.)} denotes a vector functional. This system can then be represented as shown in Figure 3.3. It follows that the Popov integral inequality can be written as

$$P(0,t1) \equiv \int_0^{t1} \{w\}^T \{v\} \, dt \geq -p_o^2$$

$$\text{for all } t1 \geq 0 \; , \quad (3.3.34)$$

where p_o^2 is the same value as given in Equation (3.3.1). This inequality is used in the definitions which follow.

3.3.6 Definition 6

The system given in Equations (3.3.31) and (3.3.32) is hyperstable if there exists a positive constant, $\delta > 0$, and a positive constant, $\gamma_o > 0$, in order that all solutions of $\{s[s_o,t]\}$ satisfy the inequality

$$\|\{s(t)\}\| < \delta[\|\{s_o\}\| + \gamma_o] \text{ for all } t\geq 0 \qquad (3.3.35)$$

for any feedback block, $\{w\} = \{f(\{c\},t,\tau)\}$, satisfying the Popov integral inequality of Equation (3.3.34).

3.3.7 Definition 7

The system given in Equations (3.3.31), (3.3.32), and (3.3.33) is asymptotically hyperstable if

1) It is hyperstable and
2) $\lim_{t\to\infty} \{s(t)\} = \{0\}$ for a feedback block, $\{w\} = \{f(c,t,\tau)\}$, which satisfies the Popov integral inequality of Equation (3.3.34).

3.3.8 Definition 8

The system given in Equations (3.3.31) and (3.3.32) is asymptotically hyperstable if it is globally asymptotically stable for all feedback blocks given in Equation (3.3.33) which satisfy the Popov integral inequality of Equation (3.3.34).

3.3.9 Definition 9

A block described by the input-output relation,

$$\{w\} = [O]\{c\} , \qquad (3.3.36)$$

where $\{w\}$ and $\{c\}$ are piecewise vector functions defined for $t\geq t_o$ and $[O]$ is an operator acting on the input $\{c\}$, is termed hyperstable if it satisfies the Popov integral inequality of Equation (3.3.34).

Definitions 6 through 9 can be extended to continuous linear, time-varying, feedforward blocks or to discrete-time systems. Popov introduced the following theorems, which can be applied to the design of model reference adaptive systems.

3.3.9.1 Popov Theorem 1

The necessary and sufficient condition for the hyperstability of the feedback system given in Equations (3.3.31), (3.3.32), and (3.3.33) is that the transfer matrix,

$$[T(h)] = [G] + [F](h[I] - [A])^{-1}[B] \ , \qquad (3.3.37)$$

must be a positive real transfer matrix.

3.3.9.2 Popov Theorem 2

The necessary and sufficient condition for the asymptotic hyperstability of the feedback system given in Equations (3.3.31), (3.3.32), and (3.3.33) is that the transfer matrix of Equation (3.3.37) must be a strictly real transfer matrix.

A linear time-varying feedback block can be written,

$$\{\dot{s}\} = [A(t)]\{s\} + [B(t)]\{v\} \qquad (3.3.38)$$

and

$$\{c\} = [F(t)]\{s\} + [G(t)]\{v\} \ , \qquad (3.3.39)$$

where $[A(t)]$, $[B(t)]$, $[F(t)]$, and $[G(t)]$ are time-varying matrices with piecewise continous elements defined for all $t \leq t_o$.

3.3.9.3 Popov Theorem 3

The sufficient condition for the hyperstability of the system given in Equations (3.3.38), (3.3.39), (3.3.33), and (3.3.34) is that the blocks of Equations (3.3.38) and (3.3.39) verify one of the positivity lemmas given in sections 3.3.5.1 and 3.3.5.2.

The definitions and theorems presented in section 3.3 can be applied to the design of the adaptation mechanism of a model reference adaptive system. The parallel model reference adaptive system given in Equations (3.2.1), (3.2.2), and (3.2.3), with adaptation law in the form

$$[A_s(c,t)] = \int_0^t [R_1(c,t,\tau)]\, d\tau + [R_2(c,t)] + [A_{so}] \tag{3.3.40}$$

and

$$[B_s(c,t)] = \int_0^t [S_1(c,t,\tau)]\, d\tau + [S_2(c,t)] + [B_{so}] \ , \tag{3.3.41}$$

can be represented in the equivalent feedback system as follows:

$$\{\dot{e}\} = [Ar]\{e\} + [I]\{w_1\} \ , \tag{3.3.42}$$

$$\{c\} = [D]\{e\} \ , \tag{3.3.43}$$

and

$$\{w\} = -\{w_1\} = \{ \int_0^{t1} [R_1(c,t,\tau)]d\tau + [R_2(c,t)]$$

$$+ [A_{so}] - [A_r] \} \{s\} + \{ \int_0^{t1} [S_1(c,t,\tau)] d\tau$$

$$+ [S_2(c,t)] + [B_{so}] - [B_r] \} \{v\} . \qquad (3.3.44)$$

Determination of the matrices, [D], $[R_1]$, $[R_2]$, $[S_1]$, and $[S_2]$, must provide the following conditions:

$$\lim_{t\to\infty} \{e(t)\} = \{0\} , \qquad (3.3.45)$$

$$\lim_{t\to\infty} [A_s(c,t)] = [A_r]$$

and

$$\lim_{t\to\infty} [B_s(c,t)] = [B_r] . \qquad (3.3.46)$$

When the Popov integral inequality is applied to the equivalent feedback system for the model reference adaptive system expressed in Equations (3.3.42), (3.3.43), and (3.3.44), it yields the inequality

$$P(0,t1) = \int_0^{t1} \{c\}^T \{ \int_0^{t} [R1(c,t,\tau)] \, d\tau$$

$$+ [R_2(c,t)] + [A_o] \} \{s\} \, dt +$$

$$\int_0^{t1} \{c\}^T \{ \int_0^{t} [S1(c,t,\tau)] \, d\tau$$

$$+ [R_2(c,t)] + [A_o] \} \{v\} \, dt \geq -p_0^2 , \qquad (3.3.47)$$

where $[A_o] = [A_{so}] - [A_r]$, $[B_o] = [B_{so}] - [B_r]$, and p_o^2 have the same properties as given in Equation (3.3.1). Equation (3.3.47) can then be divided by two inequalities in the forms

$$P_R(0,t1) = \int_0^{t1} \{c\}^T \{ \int_0^t [R1(c,t,\tau)]\, d\tau + [R_2(c,t)] + [A_o] \} \{s\}\, dt \geq -p_R^2 \quad (3.3.48)$$

and

$$P_{SR}(0,t1) = \int_0^{t1} \{c\}^T \{ \int_0^t [S1(c,t,\tau)]\, d\tau + [S_2(c,t)] + [B_o] \} \{v\}\, dt \geq -p_S^2 , \quad (3.3.49)$$

where p_R^2 and p_S^2 are finite positive constants. Furthermore, Equations (3.3.48) and (3.3.50) can each be split by two corresponding sub-inequalities. Since the inequalities in Equations (3.3.48) and (3.3.49) take the same form, the formulation of solutions for any one of them can also be extended to the others. Therefore, Equation (3.3.48) can be replaced by the following sub-inequalities,

$$P_{R1}(0,t1) = \int_0^{t1} \{c\}^T \{ \int_0^t [R1(c,t,\tau)]\, d\tau + [A_o] \} \{s\}\, dt \geq -p_{R1}^2 \quad (3.3.50)$$

and

$$P_{R2}(0,t1) = \int_0^{t1} \{c\}^T [R_2(c,t,\tau)] \{s\} d\tau \geq -p_{R2}^2 , \quad (3.3.51)$$

where p_{R1}^2 and p_{R2}^2 are finite positive constants. The matrices, $[R_1]$ and $[R_2]$, which satisfy the inequality conditions in Equations (3.3.50) and (3.3.51) can be obtained by the following lemmas.

3.3.9.4 Lemma 4

The inequality of Equation (3.3.50) is satisfied by

$$[R_1(c,t,\tau)] = [R_{11}(t-\tau)]\ \{c(\tau)\}\ \langle[R_{12}]\{s(\tau)\}\rangle^T , \quad (3.3.52)$$

where

$[R_{11}(t-\tau)]$ = a positive definite square matrix kernel whose Laplace transform is a positive real transfer matrix with a pole at $h = 0$,

$[R_{12}]$ = a positive definite matrix.

3.3.9.5 Lemma 5

The inequality of Equation (3.3.51) is satisfied by

$$[R_2(c,t)] = [R_{21}(t)]\ \{c(t)\}\ \langle[R_{22}]\{s(t)\}\rangle^T , \quad (3.3.53)$$

where $[R_{21}(t)]$ and $[R_{22}(t)]$ are time-varying positive semidefinite matrices for all $t \geq 0$.

From these lemmas [19], solutions which satisfy the inequality in Equation (3.3.48) can be found in Equations (3.3.52) and (3.3.53). Similarily, the solutions which satisfy the inequality in Equation (3.3.49) can be written

$$[S_1(c,t,\tau)] = [S_{11}(t-\tau)]\ \{c(\tau)\}\ \langle[S_{12}]\{v(\tau)\}\rangle^T \quad \text{for all } t \geq \tau \quad (3.3.54)$$

and

$[S_2(c,t)]$

$$= [S_{21}(t)]\ \{c(t)\}\ \{[S_{22}]\{s(t)\}\}^T\ , \qquad (3.3.55)$$

where

$[S_{11}(t-\tau)]$ = a positive definite matrix kernel whose Laplace transform is a positive real transfer matrix with a pole at h=0,

$[S_{12}]$ = a positive constant matrix, and

$[S_{21}(t)]$ and

$[S_{22}(t)$ = time-varying positive definite matrices for all $t \geq 0$.

For a special case of integral and proportional adaptation law, the matrices in Equations (3.3.52) through (3.3.55) can be modified in the forms

$$[R_{11}(t-\tau)] = [R_{11}] \geq 0,\ [S_{11}(t-\tau)]$$
$$= [S_{11}] \geq 0,\ \text{for all } t\geq 0\ , \qquad (3.3.56)$$
$$[R_{21}(t)] = [R_{21}],[S_{21}(t)] = [S_{21}],$$

and

$$[R_{22}(t)] = [R_{22}],[S_{22}(t)]$$
$$= [S_{22}]\ \text{for all } t\geq 0\ . \qquad (3.3.57)$$

The solutions in Equations (3.3.52) through (3.3.57) can be used for the feedback block, which is the structure of the adaptation mechanism. The theorem given in section 3.3.4.1 and the lemma given in section 3.3.4.2 can be applied to the condition that the feedforward block be a strictly positive transfer matrix. The adaptation laws, then, can be expressed in the following theorem.

3.3.9.6 Adaptation Laws Theorem

The parallel model reference adaptive system represented by Equations (3.2.1), (3.2.2), (3.2.3), (3.2.16), (3.3.40), and (3.3.41) is globally asymptotically stable if

1) $[R_1(c,t,\tau)]$, $[R_2(c,t)]$, $[S_1(c,t,\tau)]$, and $[S_2(c,t)]$ are given by Equations (3.3.52) to (3.3.55) and
2) The transfer matrix

$$[T(h)] = [D]\{h[I] - [A_r]\}^{-1} \tag{3.3.58}$$

is a strictly positive real matrix.

3.4 Convergence of the Parameters

The paramenter convergence problem is important in the application of the model reference adaptive system to the identification problems represented by Equation (3.3.46). Since the system is asymptotically stable in the {e} space, the following relations are valid.

$$\lim_{t\to\infty}(\{r\}-\{s\}) = \lim_{t\to\infty}\{e(t)\} = \{0\},\ \lim_{t\to\infty}\{\dot{e}(t)\}$$
$$= \{0\}\ . \tag{3.3.59}$$

The subtraction of Equation (3.2.2) from (3.2.1), and the addition and substraction of $[A_r]\{s\}$, yields

$$\{\dot{e}(t)\} = [A_r]\{e\} + \{[A_r] - [A_s(c,t)]\}\{s\}$$
$$+ \{[B_r] - [B_s(c,t)]\}\{v\}\ . \tag{3.3.60}$$

When t tends to ∞, the introduction of Equation (3.3.59) into Equation (3.3.60) gives

$$\{[A_r]-[A_s(c,t)]\}\{s\} + \{[B_r]-[B_s(c,t)]\}\{v\} = 0 \; . \tag{3.3.61}$$

Equation (3.3.59) indicates that the state of the adjustable system {s} can be replaced by {r} as t goes to infinity. Therefore, Equation (3.3.61) is modified in the form,

$$\{[A_r]-[A_s(c,t)]\}\{r\} + \{[B_r]-[B_s(c,t)]\}\{v\} = 0 \; . \tag{3.3.62}$$

If {r} and {v} are linearly independent vector functions, Equation (3.3.58) is valid, indicating that the parameters of the adjustable system asymptotically converge with the values of the parameters of the reference model. The functions {r} and {v} are linearly independent on condition that:

1) the reference model is completely controllable,
2) the components of {v} are linearly independent, and
3) each component of {v} contains at least $(N + 1)/2$ distinct frequencies [19].

IV. APPLICATION OF MODEL REFERENCE ADAPTIVE CONTROL SYSTEM TO CONTROL OF A FLEXIBLE MANIPULATOR

The application of the model reference adaptive system to control of a distributed parameter system is limited due to the necessity of mathematically modeling the system under consideration in the form of partial diferential equations. In particular, analysis of control system stability is a difficult task. However, the assumed-mode method discussed in Chapter II approximates the model expressed in the partial differential equations as discretized ordinary differential equations. Furthermore, the system model in generalized coordinates can be expressed in principal coordinates. These properties make it possible to apply the model reference adaptive system to the control of such distributed parameter systems as flexible manipulators. In this chapter, the design principles of a model reference adaptive system are modified in order that the major results developed from typical model reference adaptive systems can be applied to flexible manipulator control. Computer simulation of this application and the results of the simulation are discussed.

4.1 Modified Control Scheme of the Model Reference Adaptive System

The general representation of the model reference adaptive system discussed in the previous chapter can be summarized as follows.

1) The reference model may be expressed as

$$\{\dot{r}\} = [A_r]\{r\} + [B_r]\{v\}, \ \{r(0)\}=\{r_o\} \ ; \qquad (4.1.1)$$

2) The parallel adjustable system is:

$$\{\dot{s}\} = [A_s]\{s\} + [B_s]\{v\}, \ \{s(0)\} = \{s_o\},[A_s(0)]=[A_{so}] \ [B_s(0)]=[B_{so}] \ ; \qquad (4.1.2)$$

3) The state generalized error is:

$$\{e\} = \{r\} - \{s\} \ ; \qquad (4.1.3)$$

and

4) The adaptation mechanism:

$$\{c\} = [D] \ \{e\} \ , \qquad (4.1.4)$$

$$[A_s(c,t)] = \int_0^t [R_1(c,t,\tau)] \, d\tau + [R_2(c,t)] + [A_{so}] \ , \qquad (4.1.5)$$

and

$$[B_s(c,t)] = \int_0^t [S_1(c,t,\tau)] \, d\tau + [S_2(c,t)] + [B_{so}] \ . \qquad (4.1.6)$$

where

$$[R_1(c,t,\tau)] = [R_{11}(t-\tau)] \ \{c(\tau)\} \ \{[R_{12}]\{s(\tau)\}\}^T \ , \qquad (4.1.7)$$

$$[R_2(c,t,\tau)]$$
$$= [R_{21}(t)]\ \{c(t)\}\ \{[R_{22}(t)]\{s(t)\}\}^T\ , \qquad (4.1.8)$$
$$[S_1(c,t,\tau)]$$
$$= [S_{11}(t-\tau)]\ \{c(\tau)\}\ \{[S_{12}]\{v(\tau)\}\}^T\ , \qquad (4.1.9)$$

and

$$[S_2(c,t,\tau)]$$
$$= [S_{21}(t)]\ \{c(t)\}\ \{[S_{22}(t)]\{v(t)\}\}^T\ . \qquad (4.1.10)$$

The matrices in Equations (4.1.7) through (4.1.10) have the same properties as the correspondinding matrices in Equations (3.3.52) through (3.3.55). If the system expressed in Equations (4.1.1) and (4.1.2) represent multi-input, multi-output systems, the design of the adaptation mechanism becomes more complicated. This is also true for the case of a mathematical model in the generalized coordinates of a flexible manipulator. However, if there exists a modal matrix which can be used to transfer the equations of motion in principal coordinates, this difficulty can be overcome.

If an orthogonal matrix, [U], exists, then the state of the adjustable system, {s}, can be transformed into a new set of states, $\{s_u\}$:

$$\{s_u(t)\} = [U]^{-1}\ \{s\}\ . \qquad (4.1.11)$$

The adjustable system can be also transformed into

$$\{\dot{s}_u\} = [A_{us}]\{s_u\}+[B_{us}]\{v\} \qquad (4.1.12)$$

where

$$[A_{us}] = [U]^{-1} [A_s] [U]$$

$$[B_{us}] = [U]^{-1} [B_s] \ . \tag{4.1.13}$$

The adaptation mechanisms can be constructed by

$$\{c_u\} = [U]^{-1} \{c\} \ , \tag{4.1.14}$$

$$[A_{us}(c_u,t)] = \int_0^t [R_{u1}(c_u,t,\tau)] \, d\tau + [R_{u2}(c_u,t)] + [A_{uso}] \ , \tag{4.1.15}$$

and

$$[B_{us}(c_u,t)] = \int_0^t [S_{u1}(c_u,t,\tau)] \, d\tau + [S_{u2}(c_u,t)] + [B_{uso}] \tag{4.1.16}$$

where

$$[R_{u1}(c_u,t,\tau)] = [R_{u11}(t-\tau)]\{c_u(\tau)\} \{[R_{u12}]\{s_u(\tau)\}\}^T \ , \tag{4.1.17}$$

$$[R_{u2}(c_u,t,\tau)] = [R_{u21}(t)]\{c_u(t)\} \{[R_{u22}(t)]\{s_u(t)\}\}^T \ , \tag{4.1.18}$$

$$[S_{u1}(c_u,t,\tau)] = [S_{u11}(t-\tau)]\{c_u(\tau)\} \{[S_{u12}]\{v(\tau)\}\}^T \ , \tag{4.1.19}$$

$$[S_{u2}(c_u,t,\tau)] = [S_{u21}(t)]\{c_u(t)\} \{[S_{u22}(t)]\{v(t)\}\}^T \ , \tag{4.1.20}$$

$$[A_{uso}] = [U]^{-1}[A_s(0)][U] \ , \tag{4.1.21}$$

and

$$[B_{uso}] = [U]^{-1}[B_s(0)] \ . \tag{4.1.22}$$

The matrices in Equations (4.1.16) through (4.1.20) have the same equivalent properties as those in Equations (4.1.7) through (4.1.10) . For the application of the transformed adaptation mechanism in Equations (4.1.14) through (4.1.20), it must be verified that after it is transformed into the original state system, the adaptation mechanism satisfies the conditions in Equations (4.1.4) to (4.1.10).

From Equation (4.1.13), the adaptation mechanism can be transformed into the original state system in the form

$$[A_s]' = [U][A_{us}(c_u,t)][U]^{-1}$$
$$[B_s]' = [U][B_{us}(c_u,t)] \quad . \qquad (4.1.23)$$

The substitution of Equations (4.1.17) through (4.1.20) into Equation (4.1.23) and the introduction of Equations (4.1.21) and (4.1.22) yields

$$[A_s]' = \int_0^t [R_1(c_u,t,\tau)]'\, d\tau + [R_2(c_u,t)]' + [A_{so}] \qquad (4.1.24)$$

and

$$[B_s]' = \int_0^t [S_1(c_u,t,\tau)]'\, d\tau + [S_2(c,t)]' + [B_{uso}] \qquad (4.1.25)$$

where

$$[R_1(c_u,t,\tau)]' = [U]\ [R_{u11}(t-\tau)]\{c_u(\tau)\} \times \{[R_{u12}]\{s_u(\tau)\}\}^T\ [U]^{-1} \ , \qquad (4.1.26)$$

$$[R_2(c_u,t,\tau)]' = [U]\ [R_{u21}(t)]\{c_u(t)\} \times \{[R_{u22}(t)]\{s_u(t)\}\}^T\ [U]^{-1} \ , \qquad (4.1.27)$$

$$[S_1(c_u,t,\tau)]' = [U]\ [S_{u11}(t-\tau)]\{c_u(\tau)\} \times \{[S_{u12}]\{v(\tau)\}\}^T \ , \qquad (4.1.28)$$

and

$$[S_2(c_u,t,\tau)]' = [U]\,[S_{u21}(t)]\{c_u(t)\}$$
$$\times \{[S_{u22}(t)]\{v(t)\}\}^T \,. \tag{4.1.29}$$

When the property of the transpose of matrices

$$([A][B])^T = [B]^T[A]^T \tag{4.1.30}$$

and the property of the orthogonal matrix

$$[U]^{-1} = [U]^T \tag{4.1.31}$$

are applied to Equations (4.1.26) to (4.1.29), one obtains the following relations:

$$[R_1]' = [R_{11}]'\{c\}\{[R_{12}]'\{s\}\}^T \,, \tag{4.1.32}$$

$$[R_2]' = [R_{21}]'\{c\}\{[R_{22}]'\{s\}\}^T \,, \tag{4.1.33}$$

$$[S_1]' = [S_{11}]'\{c\}\{[S_{12}]'\{v\}\}^T \,, \tag{4.1.34}$$

and

$$[S_2]' = [S_{21}]'\{c\}\{[S_{22}]'\{v\}\}^T \,, \tag{4.1.35}$$

where

$$[R_{ij}]' = [U]\,[R_{uij}]\,[U]^{-1},\ i,j=1,2 \,, \tag{4.1.36}$$

$$[S_{i1}]' = [U]\,[S_{ui1}]\,[U]^{-1},\ i=1,2 \,, \tag{4.1.37}$$

and

$$[S_{i2}]' = [S_{i2}],\ i=1,2 \,. \tag{4.1.38}$$

It is obvious that the matrices $[R_{ij}]'$ and $[S_{ij}]'$ in Equations (4.1.36) to (4.1.38) possess the same properties as the matrices $[R_{ij}]$ and $[S_{ij}]$ in Equations (4.1.7) to (4.1.10) since $[R_{uij}]$ and $[S_{uij}]$ are determined by the Lemmas 3.3.9.4 and 3.3.9.5 and the matrix, $[U]$, is orthogonal. This leads to the following lemma.

4.1.1 Lemma 6

The adaptation mechanisms described by Equations (4.1.11), (4.1.14) through (4.1.20), and Equation (4.1.23) are equivalent to the adaptation mechanism in Equations (4.1.4) to (4.1.10). Therefore, the equivalent properties of the adaptation mechanism can be characterized by the following theorem.

4.1.2 Theorem

The following conditions of the model reference adaptive system adaptation mechanism expressed in Equations (4.1.1) and (4.1.2) are equivalent to each other:

1) The parallel model reference adaptive system expressed in Equations (4.1.1) through (4.1.6) is globally asymptotically stable if the adaptation mechanisms are given by Equations (4.1.24) through (4.1.29); and
2) The parallel model reference adaptive system expressed in Equations (4.1.1) through (4.1.6) and Equation (4.1.11) is globally stable if the adaptation mechanisms are given by Equations (4.1.14) through (4.1.22).

If the orthogonal matrix is identical with the modal matrix of the system described in Equation (2.3.46), this theorem can be employed for the control of the physical model represented in Chapter II. A detailed procedure for its application and a computer simulation are discussed in the following sections.

4.2 Application of the Model Reference Adaptive System to Control of a Flexible Manipulator

If the mathematical model of a flexible manipulator is described by the assumed-mode method, the equations of motion in generalized coordinates can be written as

$$[m]\{\ddot{q}\} + [c]\{\dot{q}\} + [k]\{q\} = \{p\} \ , \qquad (4.2.1)$$

where $\{q\}$ is from the approximation of $y(x,t)$ in the form

$$y(x,t) = \sum_{i=1}^{N} X_i(x) q_i(t) \ . \qquad (4.2.2)$$

The principal coordinates are then defined by

$$\{s(t)\} = [U]^{-1}\{q(t)\} \qquad (4.2.3)$$

by the use of eigenpairs corresponding to

$$([k] - w^2[m])\{u\}_i = \{0\}, \ i=1,2,- - -,N \ . \qquad (4.2.4)$$

The matrix $[U]$ in Equation (4.2.3) is given by

$$[U] = [\{u\}_1 \ \{u\}_2 \ - - - \ \{u\}_N] \ . \qquad (4.2.5)$$

Therefore, the equations of motion in principal coordinates can be written as

$$[M]\{\ddot{s}(t)\} + [C]\{\dot{s}(t)\} + [K]\{s(t)\} = \{P\} \ , \qquad (4.2.6)$$

where

$$[M] = [U]^{-1}[m][U] = \mathrm{diag}[M_1 \ M_2 \ - - - \ M_N] \ ,$$

$$[C] = [U]^{-1}[c][U] = \mathrm{diag}[C_1 \ C_2 \ - - - \ C_N] \ ,$$

$$[K] = [U]^{-1}[k][U] = \text{diag}[K_1\ K_2\ -\ -\ -\ K_N]\ ,$$
$$\{P\} = [U]^{-1}\ \{p\}\ , \qquad (4.2.7)$$

or

$$M_i\ \ddot{s}_i(t) + C_i\ \dot{s}_i(t) + K_i\ s_i(t) = P_i(t),$$
$$i=1,2,-\ -\ -,N\ . \qquad (4.2.8)$$

Note that the damping matrix, [c], is assumed to be modal damping as discussed in Chapter II. The system described in Equation (4.2.8) is seen to consist of N, independent, ordinary differential equations. The availability of the expression of the equations of motion in generalized and in principal coordinates provides for the application of the adaptation laws of a model reference adaptive system for a distributed parameter system.

The reference model, therefore, can be given by

$$[m_r]\{\ddot{q}_r\} + [c_r]\{\dot{q}_r\} + [k_r]\{q_r\} = \{b_r\}\{v\} \qquad (4.2.9)$$

where $[b_r]\{v\} = \{p_r]$.

Similarly, the adjustable system can be given by

$$[m_s]\{\ddot{q}_s\} + [c_s]\{\dot{q}_s\} + [k_s]\{q_s\}$$
$$= \{b_s\}\{v\}\ . \qquad (4.2.10)$$

Equation (4.2.9) can be also represented in the state system equation by the the form

$$\{\dot{r}(t)\} = [A_r]\ \{r(t)\} + [B_r]\ \{v\} \qquad (4.2.11)$$

where

$$\{r(t)\} = (\{q_r(t)\}^T \ \{\dot{q}_r(t)\}^T)^T , \tag{4.2.12}$$

$$[A_r] = \begin{bmatrix} [0] & [I] \\ [A_{r1}] & [A_{r2}] \end{bmatrix} = \begin{bmatrix} [0] & [I] \\ [m_r]^{-1}[k_r] & [m_r]^{-1}[c_r] \end{bmatrix} , \tag{4.2.13}$$

and

$$[B_r] = \begin{bmatrix} [0] \\ [B_{r1}] \end{bmatrix} = \begin{bmatrix} [0] \\ [m_r]^{-1}[b_r] \end{bmatrix} . \tag{4.2.14}$$

Similarly,

$$\{\dot{s}(t)\} = [A_s] \{s(t)\} + [B_s] \{v\} \tag{4.2.15}$$

where

$$\{s(t)\} = (\{q_s(t)\}^T \ \{\dot{q}_s(t)\}^T)^T , \tag{4.2.16}$$

$$[A_s] = \begin{bmatrix} [0] & [I] \\ [A_{s1}] & [A_{s2}] \end{bmatrix} = \begin{bmatrix} [0] & [I] \\ [m_s]^{-1}[k_s] & [m_s]^{-1}[c_s] \end{bmatrix} , \tag{4.2.17}$$

and

$$[B_s] = \begin{bmatrix} [0] \\ [B_{s1}] \end{bmatrix} = \begin{bmatrix} [0] \\ [m_s]^{-1}[b_s] \end{bmatrix} . \tag{4.2.18}$$

For the construction of the adaptation mechanism, the state values of both the reference model and the adjustable system must be measured as feedback information. Since the systems are descretized by the assumed modes, it is difficult to directly measure the values of the generalized or principal states. The construction of the sensors along the flexible manipulator arm is limited in practical terms by environmental conditions. However, these difficulties can be overcome by the use of strain gauges, the number of which are the same as the numer of assumed modes. If the dynamic strain gauges are placed at N locations along the manipulator arm with N assumed modes, the gauge readings can be expressed in the form:

$$\begin{Bmatrix} y_1'' \\ y_2'' \\ \cdot \\ \cdot \\ \cdot \\ y_N'' \end{Bmatrix} = \begin{bmatrix} X_{11}'' & X_{12}'' & \cdots & X_{1N}'' \\ X_{21}'' & X_{22}'' & \cdots & X_{2N}'' \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ X_{N1}'' & X_{N2}'' & \cdots & X_{NN}'' \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ \cdot \\ \cdot \\ \cdot \\ q_N \end{Bmatrix} \quad (4.2.19)$$

where

$$y_I = y_I(x_I, t) \ , \quad (4.2.20)$$

$$X_{IJ}'' = X_J''(x_I), \ I,J = 1,2,- - -,N \ , \quad (4.2.21)$$

and X_{IJ}'' indicates the second derivative of X_I with respect to x and x_I, which are the N-fixed locations of the dynamic gauges. Therefore, the state of the system in generalized coordinates can be written

$$\begin{Bmatrix} q_1 \\ q_2 \\ \cdot \\ \cdot \\ \cdot \\ q_N \end{Bmatrix} = \begin{bmatrix} X_{11}'' & X_{12}'' & \cdots & X_{1N}'' \\ X_{21}'' & X_{22}'' & \cdots & X_{2N}'' \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ X_{N1}'' & X_{N2}'' & \cdots & X_{NN}'' \end{bmatrix}^{-1} \begin{Bmatrix} y_1'' \\ y_2'' \\ \cdot \\ \cdot \\ \cdot \\ y_N'' \end{Bmatrix}_m$$

or in short form,

$$\{q(t)\} = [X'']^{-1}\{y''\}_m \ , \quad (4.2.22)$$

where $\{y''\}_m$ are the gauge readings. Similarly, the derivatives of the state can be represented in the form

$$\{\dot{q}(t)\} = [X'']^{-1}\ \{\dot{y}(t)''\} \tag{4.2.23}$$

where $\{\dot{y}(t)\}_m$ can be approximated by

$$\dot{y}_I(t) \approx \frac{\Delta U_I}{\Delta t} = \frac{U_I(x_I, t-\Delta t) - U_I(x_I, t)}{\Delta t},$$

$$I=1,2,- - -,N \tag{4.2.24}$$

if Δt is sufficiently small and the functions, $y_I(t)$, are continuous.

The adjustable system described by Equation (4.2.15) in a generalized coordinate state can be transformed into a principal state system in the form

$$\{s_p(t)\} = [U_{sp}]^{-1}\ \{s\}\ , \tag{4.2.25}$$

where

$$[U_{sp}] = \begin{bmatrix} [U_s] & [0] \\ [0] & [U_s] \end{bmatrix} . \tag{4.2.26}$$

In Equation (4.2.26), the matrix, $[U_s]$, is the modal matrix of the adjustable system equivalent to the matrix, [U], in Equation (4.2.5). The adjustable system in the principal state system can be written

$$\{\dot{s}_p(t)\} = [A_{sp}]\{s_p(t)\} + [B_{sp}]\{v\} \tag{4.2.27}$$

where

$$[A_{sp}] = [U_{sp}]^{-1}[A_s][U_{sp}]$$

$$= \begin{bmatrix} [0] & [I] \\ [U_{sp1}] & [U_{sp2}] \end{bmatrix} , \tag{4.2.28}$$

$$[B_{sp}] = [U_{sp}]^{-1}[B_s]\ , \tag{4.2.29}$$

and

$$[U_{sp1}] = [M_s]^{-1}[K_s] = \{[U_s]^{-1}[m_s][U_s]\}^{-1} \times \{[U_s]^{-1}[k_s][U_s]\} \tag{4.2.30}$$

and

$$[U_{sp2}] = [M_s]^{-1}[C_s] = \{[U_s]^{-1}[m_s][U_s]\}^{-1} \times \{[U_s]^{-1}[c_s][U_s]\} \ . \tag{4.2.31}$$

The matrices $[U_{sp1}]$ and $[U_{sp2}]$ are diagonal matrices whose diagonal elements are given by

$$U_{sp1i} = K_{si} \;/\; M_{si} = w_{si}^2 \ ,$$
$$U_{sp2i} = C_{si} \;/\; M_{si} = C_{spi} \ . \tag{4.2.32}$$

If the generalized state of the adjustable system is rearranged by

$$\{s_p,\} = \{q_{s1}\ \dot{q}_{s1}\ q_{s2}\ \dot{q}_{s2}\ - - -\ q_{sN}\ \dot{q}_{sN}\}^T \tag{4.2.33}$$

the equivalent expression of Equation (4.2.27) can be obtained by the form

$$\begin{Bmatrix} \dot{s}_1, \\ s_2, \\ s_3, \\ s_4, \\ \cdot \\ \cdot \\ \cdot \\ s_I, \\ s_J, \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & & \\ w_{s1}^2 & C_{sp1} & 0 & 0 & \cdots & & \\ 0 & 0 & 0 & 1 & \cdots & & \\ 0 & 0 & w_{s22} & C_{sp2} & \cdots & & \\ & \vdots & & \vdots & & & \\ & \vdots & & \vdots & & 0 & 1 \\ & \cdot & & \cdot & & w_{sN}^2 & C_{spN} \end{bmatrix} \begin{Bmatrix} s_1, \\ s_2, \\ s_3, \\ s_4, \\ \cdot \\ \cdot \\ \cdot \\ s_I, \\ s_J, \end{Bmatrix} + \begin{bmatrix} \{0\} \\ \{B_{sp}\}^{1T} \\ \{0\} \\ \{B_{sp}\}^{2T} \\ \cdot \\ \cdot \\ \cdot \\ \{0\} \\ \{B_{sp}\}^{NT} \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_N \end{Bmatrix} , \tag{4.2.34}$$

where $J = 2xN$ and $I = J-1$.

Equation (4.2.34) can then be considered as an N-independent matrix equation in the form

$$\left\{ \begin{matrix} s_I{'} \\ s_{I+1}{'} \end{matrix} \right\} = \begin{bmatrix} 0 & 1 \\ w_{sI}^2 & C_{sI} \end{bmatrix} \left\{ \begin{matrix} s_I{'} \\ s_{I+1} \end{matrix} \right\} + \begin{bmatrix} \{0\} \\ \{Bsp\}^{I+1T} \end{bmatrix} \left\{ v \right\}$$

$$I = 1,3,5,- - -,2(N-1) \ . \qquad (4.2.35)$$

If Equation (4.2.35) is considered as a system equivalent to Equation (4.1.12), the corresponding adaptive mechanism can be determined by Equations (4.1.14) through (4.1.22). These N-subadaptive mechanisms can then be combined and rearranged according to the arragement of principal states in Equation (4.2.25), following which the adaptation mechanisms can be transformed back into the system expressed in the generalized state. The introduction of the theorem in section 4.1.2 assures that the construction of the model reference adaptive system described above procedure is hyperstable.

The discussion in this section can be applied to the design of various types of model reference adaptive systems, particularly for the control of distributed parameter systems. The dual characteristics of the model reference adaptive system leads to applications in the area of system identification. In the following section, computer simulation of the application of the above argument to the parameter identification problem of a flexible manipulator is discussed.

4.3 Computer Simulation of the Application of the Model Reference Adaptive System to the Identification of a Flexible Manipulator

The dual characteristics of the model reference adaptive system make it possible to apply its control scheme both to the control of a system and to the problem of system identification. The control scheme discussed previously will be examined in terms of the problem of system identification, which is a less complex problem [24]. In the case of the identification problem, the reference model and the adjustable system are considered, respectively, as the plant to be identified and the adaptive identifier.

The general mathematical representation of a flexible manipulator obtained by the use of assumed-mode method can be given by

$$y(x,t) = \sum_{i=1}^{N} X_i(x)\, q_j(t) \qquad (4.3.1)$$

and

$$[m]\{\ddot{q}(t)\} + [c]\{\dot{q}(t)\} + [k]\{q(t)\} = \{b\}\, \ddot{\theta}(t) \qquad (4.3.2)$$

In this study, the mode functions, $X_i(x)$, are determined in the form of Equation (2.3.16) and the elements of the matrices [m], [k], and {b} are as represented in

Appendix A. The [c] matrix can be determined by Equations (2.3.53) and (2.3.48) when damping factors are once selected.

Equation (2.3.60) can be used for dynamic system responses. Readings of the dynamic strain gauges are simulated by Equation (4.2.19), which is transformed into Equation (4.2.22) for determination of the state values of the feedback mechanism. If different parameters exist between the reference model and the adjustable system, they can be expressed in the forms of Equations (4.1.4) through (4.1.10). Since Equation (4.1.1) can be transformed into the equations of motion in principal coordinates by the introduction of the modal matrix and the orthogonal matrix [U] in Equation (4.1.11), the adaptation mechanism can be determined in the forms of Equations (4.1.15) through (4.1.22). When the plant is constructed with unknown parameters, which may be the properties of the structure of the manipulator, the system can be considered as an imperfect model. On the other hand, if critical parametric variations exist during operations, the characteristics of the variations must be identified. In both cases, the model reference adaptive system can be applied to the identification of the plant. For simplification of the computer simulation in this study, the tip mass as the payload of the manipulator will be considered as the only possible parameter variation. Parameters and the adaptation mechanisms which have been used for the computer simulation are discussed in this section.

Common parameters for both the reference model as the plant to be identified and the adjustable system as the adaptive identifier are as follows:

Bending stiffness: $EI = 5.67\text{x}10^{-4}$ N mm^2

Arm length: $L = 2000$ mm

Mass per unit length: $\rho A = 4.0 \times 10^{-5}$ kg/mm

The payload of the reference model is considered as the unknown parameter, while the payload of the adjustable system is the known parameter adjusted by the adaptation mechanism. The goal is to have the parameters of the adjustable system converge with those of the reference model. The tip mass of the reference model and the adjustable system are, respectively, initially 3×10^{-3} kg and 1.5×10^{-3} kg for the simulation. The entire control system is shown in Figure 4.1.

The adaptation mechanisms are constructed according to the concept developed in previous sections. The compensator gain matrix, [L], for the i^{th} mode is

$$[L] = \begin{bmatrix} L_1 & L_2 \\ L_2 & L_3 \end{bmatrix} \qquad i=1,2,- - -,N\ , \qquad (4.3.3)$$

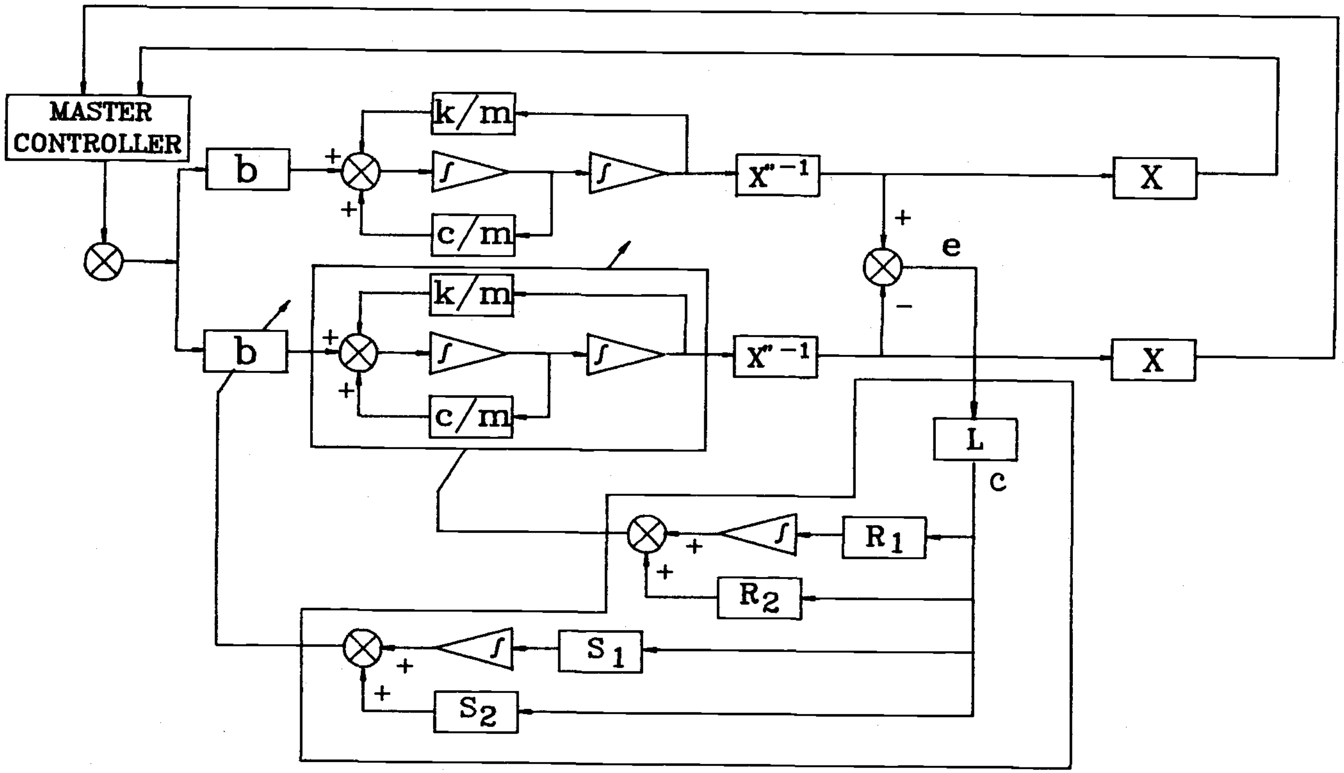


Figure 4.1 Block Diagram of Flexible Manipulator Model Reference Adaptive Control System

where

$$L_1 = \frac{c_i}{m_i} + \frac{k_i}{c_i} d_o, \; d_o > 1,$$

$$L_2 = 1 \; ,$$

$$L_3 = \frac{m_i}{c_i} d_o, \; d_o > 1 \; . \qquad (4.3.4)$$

The adaptation mechanism of integral and proportional laws have been proposed in Equations (3.3.56) and (3.3.57). For the integral and proportional adaptation law, the first adaptation block, $[R_{u1}]$, in Equation (4.1.17) can be determined by the following matrices:

$$[R_{u11}] = \begin{bmatrix} 1 & -\alpha \\ -\alpha & (\alpha^2+\beta_1{}^2) \end{bmatrix} R_{11},$$

$$i=1,2,- - -,N \qquad (4.3.5)$$

and

$$[R_{u12}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} R_{12},$$

$$i=1,2,- - -,N \; , \qquad (4.3.6)$$

where α, β_1, R_{11} and R_{12} are positive constants. For the simplification of the computer simulation, the value α can be chosen in the form

$$\alpha = \frac{c_{u1}}{c_{u2}} \qquad (4.3.7)$$

where c_{u1} and c_{u2} are given by Equation (4.1.14).

Similarly, the adaptation matrices, $[R_{u2}]$, in Equation (4.1.18) can be determined in same form as in Equations (4.3.5) and (4.3.6), only with different gains:

$$[R_{u21}] = \begin{bmatrix} 1 & -\alpha \\ -\alpha & (\alpha^2+\beta_2{}^2) \end{bmatrix} R_{21},$$

$$i=1,2,- - -,N \qquad (4.3.8)$$

and

$$[R_{u22}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} R_{22},$$

$$i=1,2,- - -,N \ . \qquad (4.3.9)$$

The adaptation blocks, $[S_{u1}]$ and $[S_{u2}]$, in Equations (4.1.19) and (4.1.20) are simplified as constants if v(t) is a one dimensional vector in order that they can be expressed in the form

$$S_{u1} = S_{u11} \times S_{u12} \qquad (4.3.10)$$

and

$$S_{u2} = S_{u21} \times S_{u22} \ . \qquad (4.3.11)$$

The numerical values of the structure of the adaptation mechanism used in the simulation program, *FLEX*, included in Appendix B, are given by

$$d_o = 2.5 \ ,$$

$$R_{11}R_{12}\beta_1 = 0.15 \ ,$$

$$R_{21}R_{22}\beta_2 = 0.4 \ ,$$

$$S_{u1} = 0.15 \ ,$$

and

$$S_{u2} = 0.4 \ . \qquad (4.3.12)$$

The first six mode shapes of the admissible functions given in Equation (2.3.16) are represented in Figures 4.2 through 4.7. Figures 4.8 to 4.10 show the deflections of the manipulator at the tip in individual cases of the reference model, the adjustable system, and the initial system without adaptative adjustment, when different adaptation gains are selected. Figures 4.11 to 4.13 represent the position errors of the manipulators considered as a rigid body, fixed at the initial state, and adjusted according to the corresponding adaptation mechanisms. The same conditions are given for each case, with the exception of the adaptation gains where numbers of the three admissible functions were used.

These figures illustrate the convergence of the adjustable system constructed by the adaptation laws developed in previous sections with the reference model. However, comparison of Figures 4.8 to 4.10 shows that the smaller the adaptation gains, the slower the rate of convergence. This tendency is shown in Figures 4.14 through 4.16, which represent the convergence of the payload for each case. The remainder of the figures in this chapter are results obtained with the adaptation gains given in Equation (4.3.12). Figure 4.17 shows the base motion, $\theta(t)$, given by

$$\theta(t) = -\pi \sin (1.5 \pi + t) . \tag{4.3.13}$$

Deflection differences of the adjustable system and the initial system are shown in Figure 4.18. While the deflection error of the intial system is not changed, the error of the adjustable system is diminished as time increases.

Error vectors filtered through the compensation matrix, [L], are represented in Figures 4.19 to 4.24: position error vectors in Figures 4.19 to 4.21 and velocity error vectors in Figures 4.22 to 4.24. Figures 4.19 and 4.22, Figures 4.20 and 4.23, and Figures 4.21 and 4.24, respectively, are the error vectors corresponding to the first, second, and third modes. The fact that the error vectors corresponding to the mode functions of the higher-frequencies die out faster than the lower-frequency mode functions, indicates the diminished effectiveness of the higher frequency to the system error.

The variations of the integral adaptation mechanisms are shown in Figures 4.25 to 4.27. As time increases, the values of the adaptation gains of the integral blocks become stationary. On the other hand, the proportional adaptation gains, which are represented in Figures 4.28 to 4.30, converge to zero as the adjustable system approaches the reference model.

The convergence of the parameters of the adjustable system are shown in Figures 4.31 to 4.36: B_i and M_i indicate elements of the [B] and [M] matrices of the dynamic equations of motion in the principal coordinates.

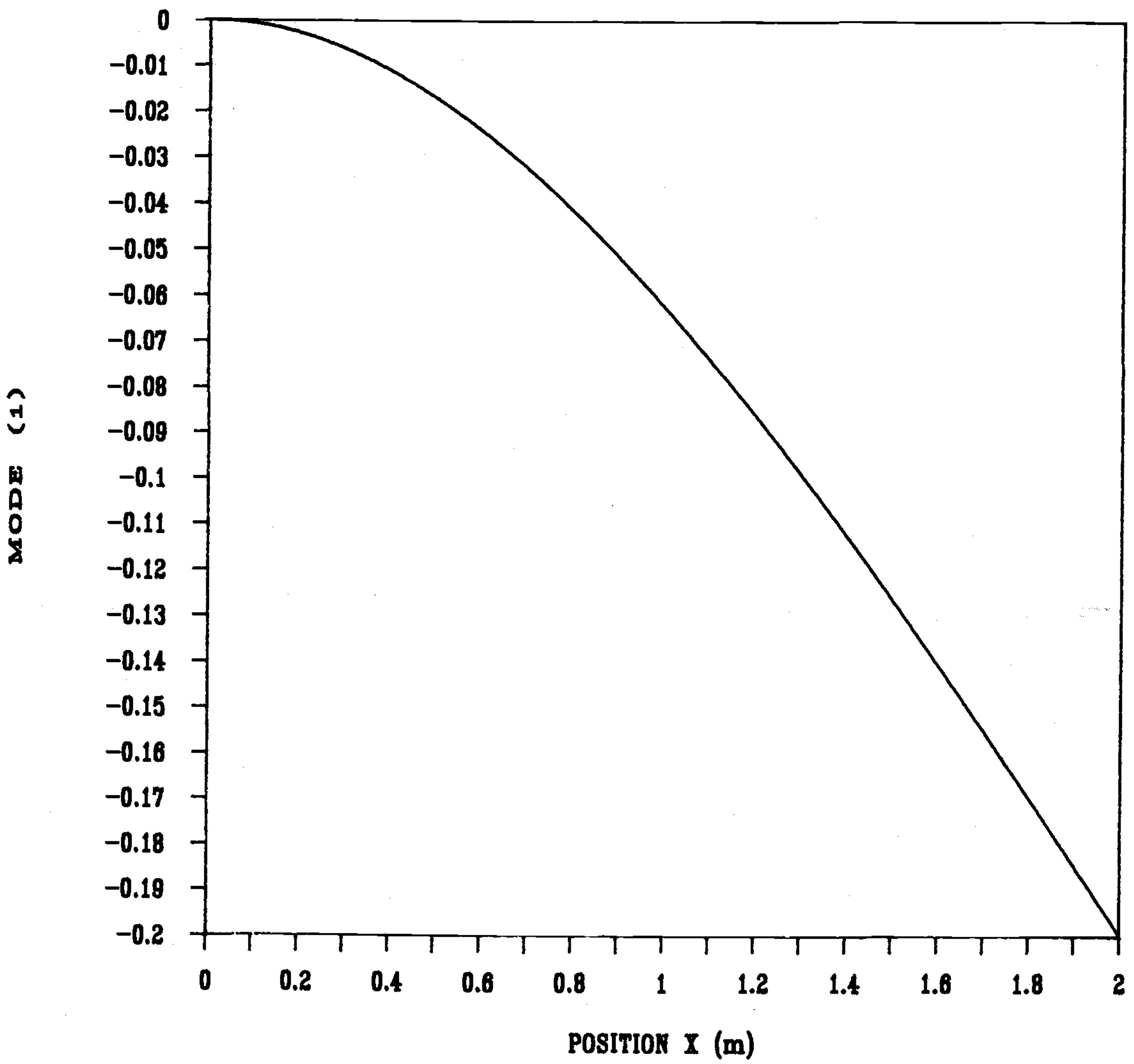


Figure 4.2 Mode Shape Function (1)

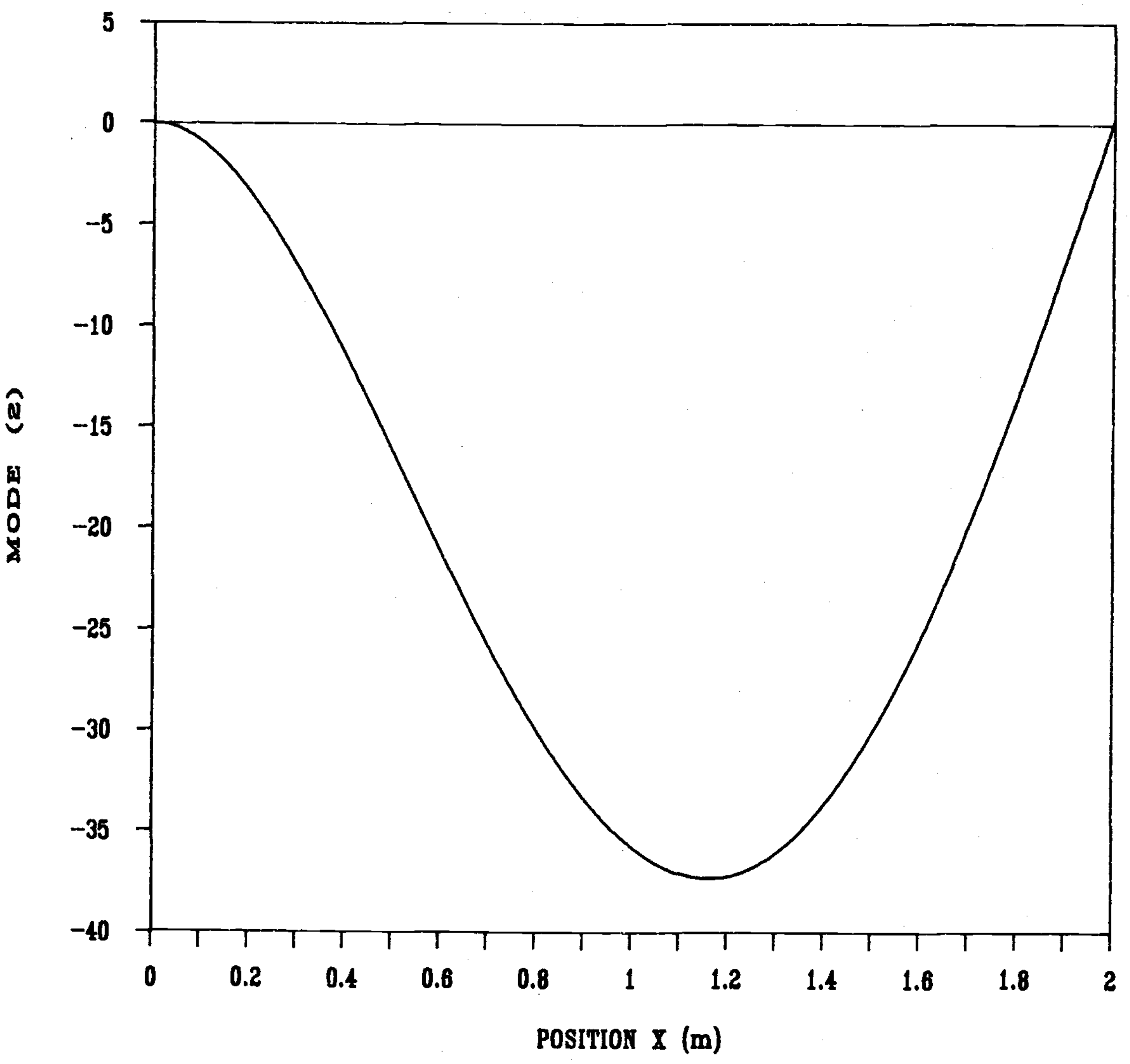


Figure 4.3 Mode Shape Function (2)

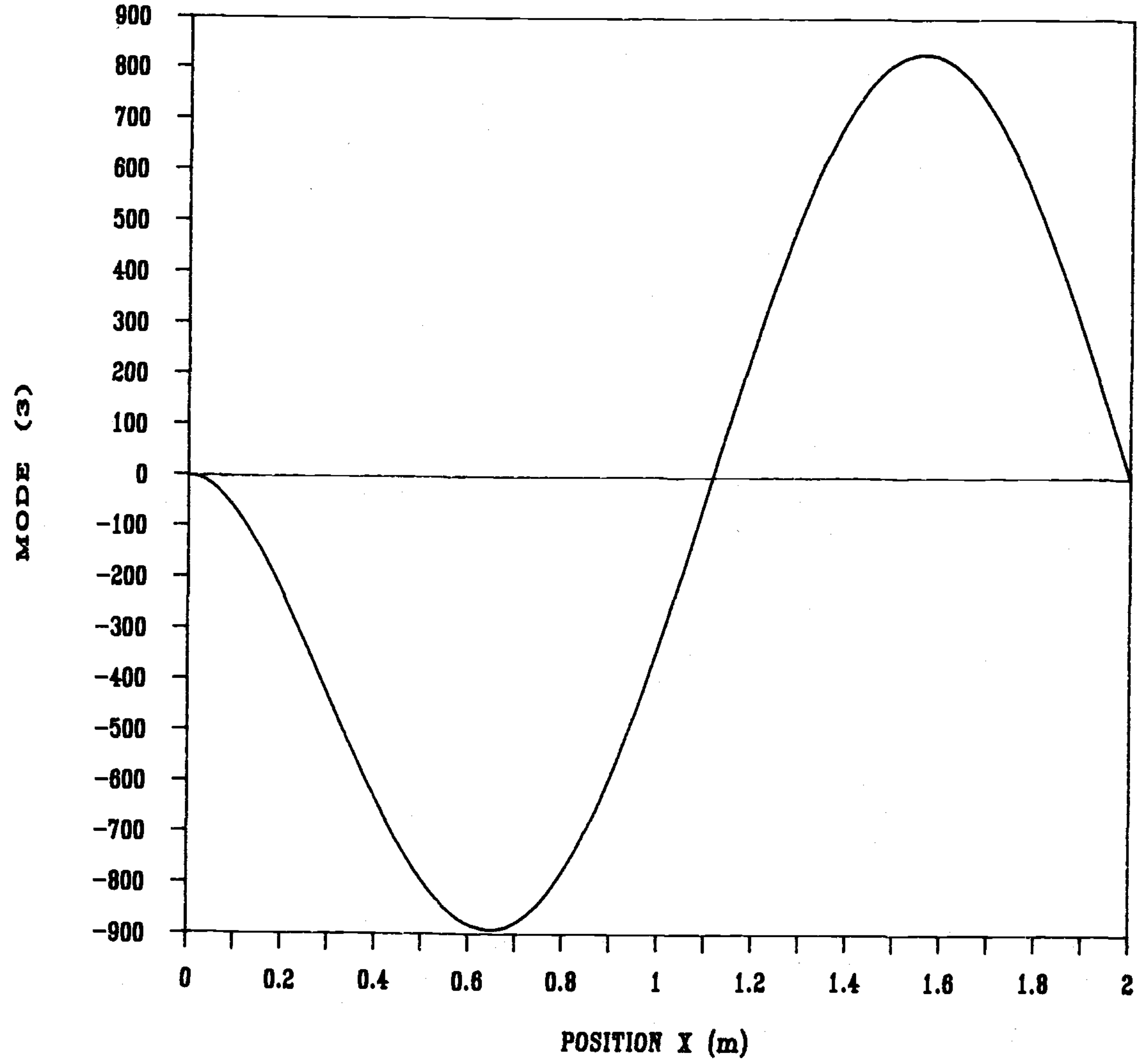


Figure 4.4 Mode Shape Function (3)

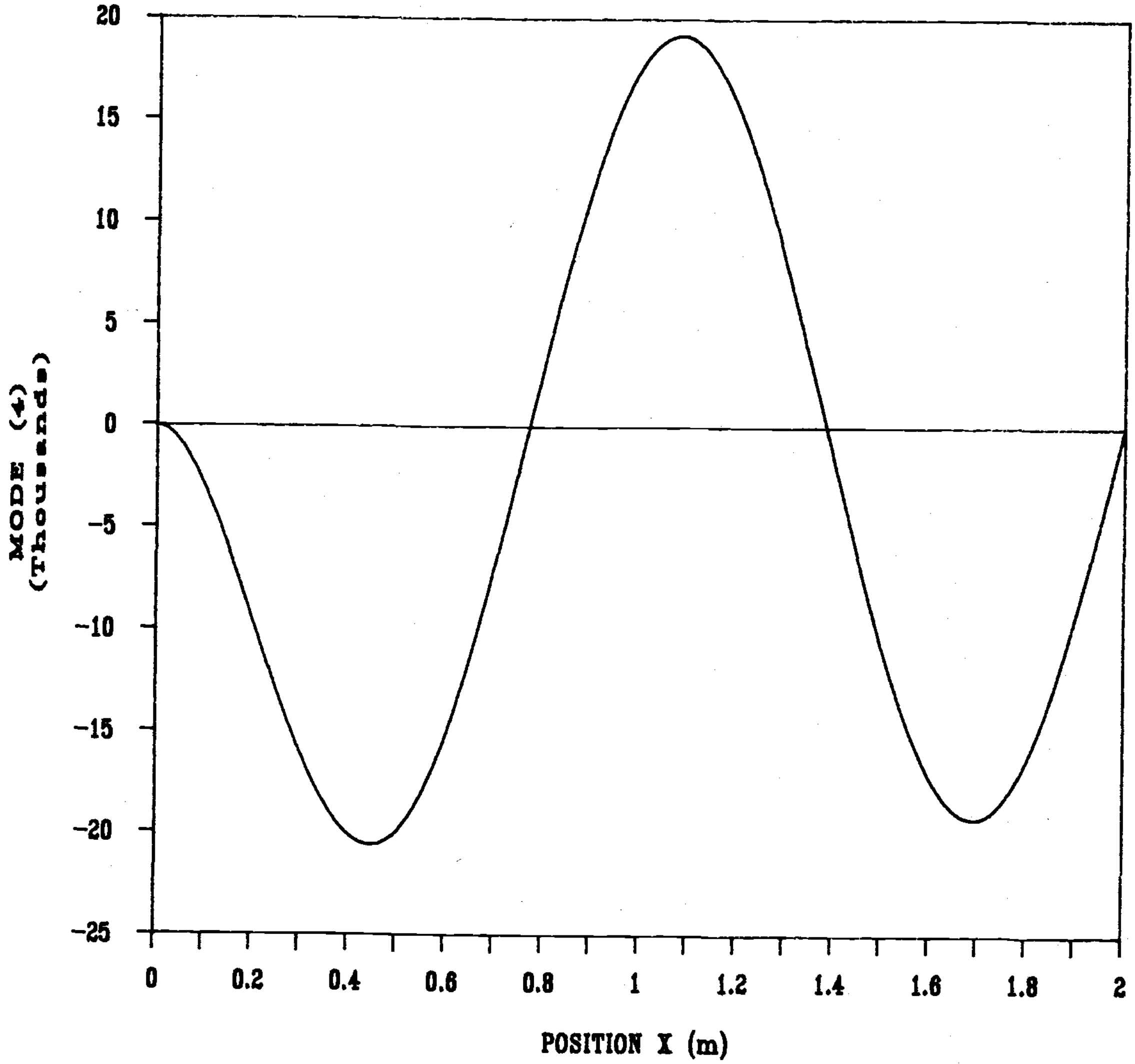


Figure 4.5 Mode Shape Function (4)

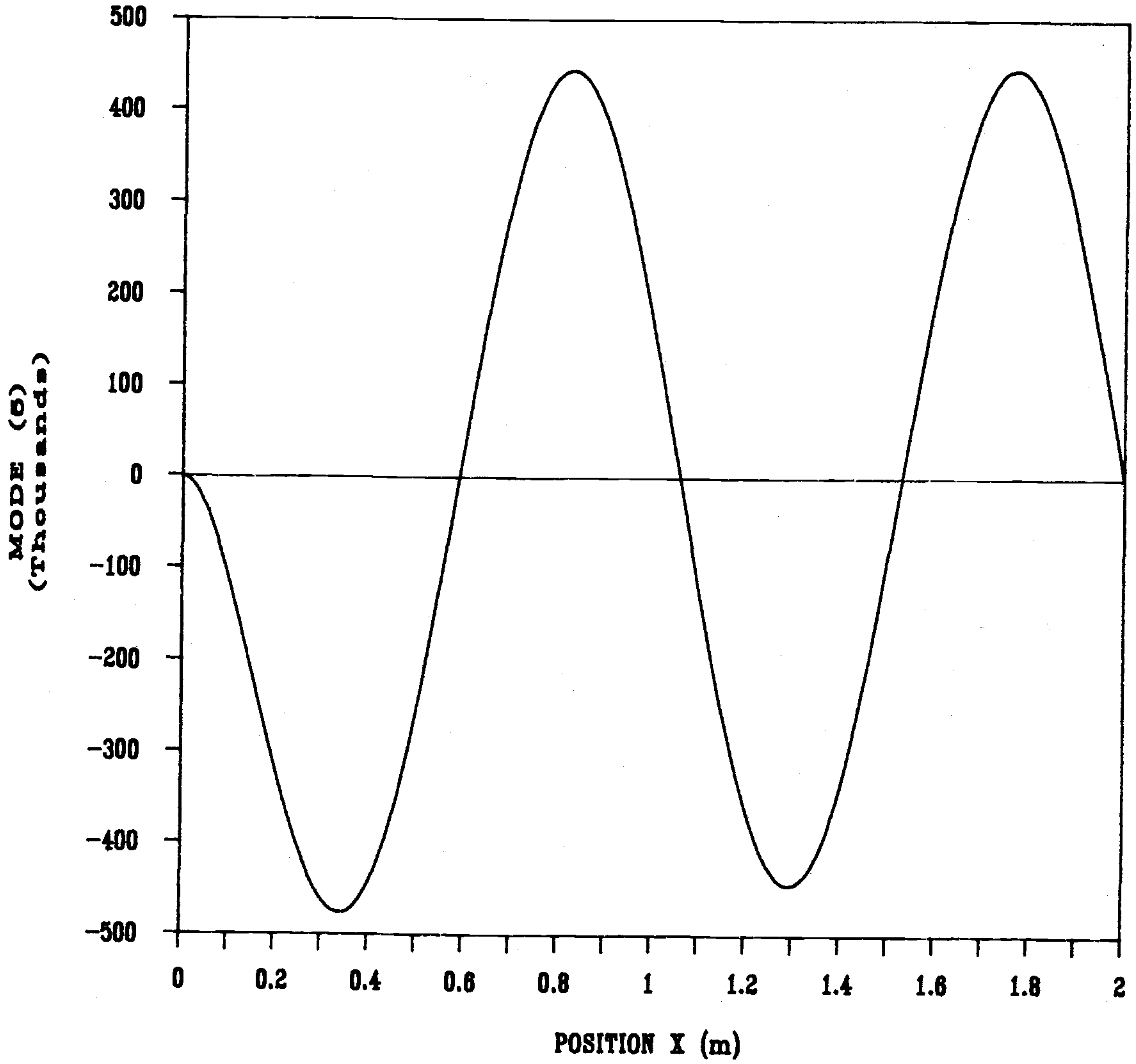


Figure 4.6 Mode Shape Function (5)

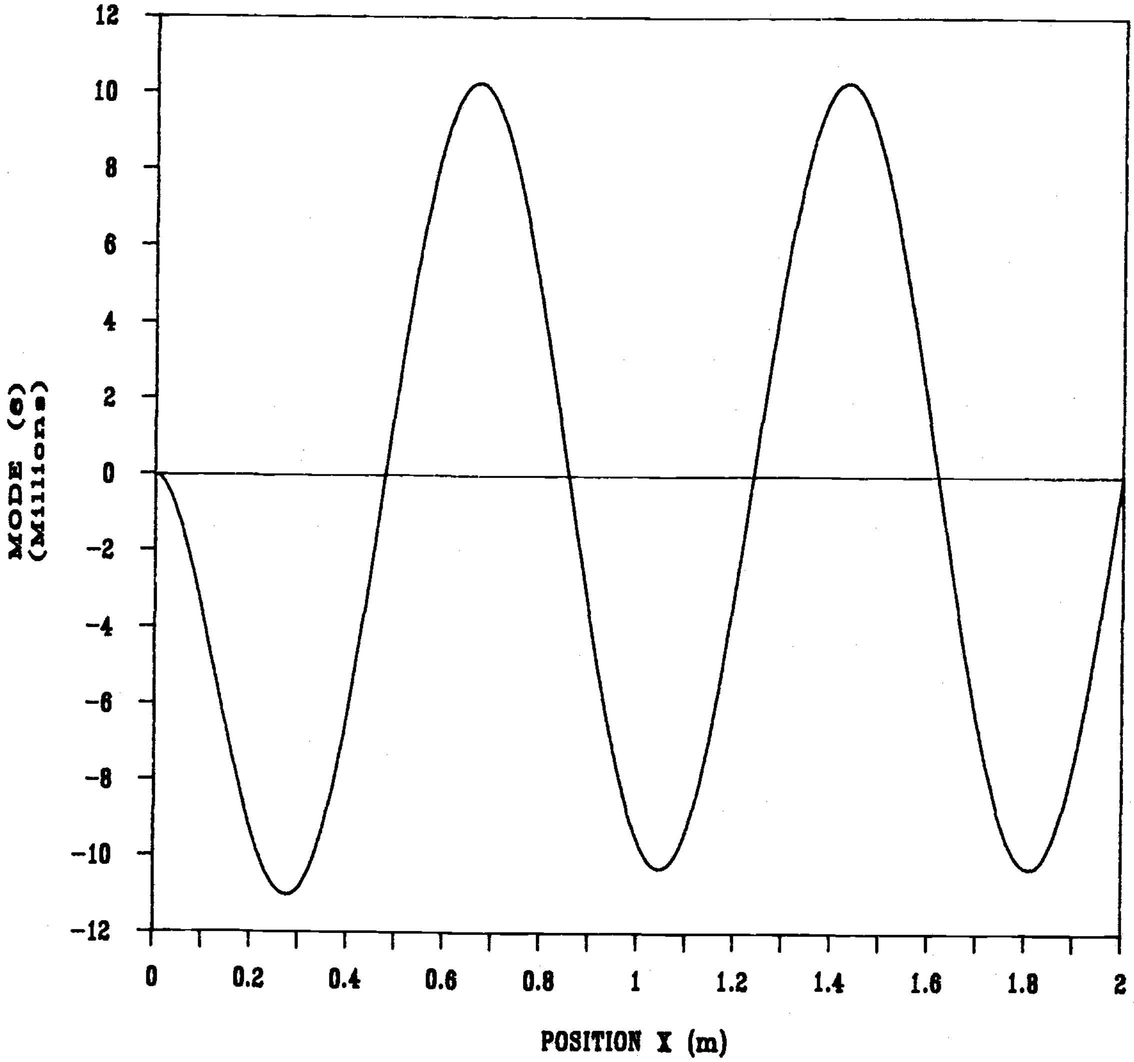


Figure 4.7 Mode Shape Function (6)

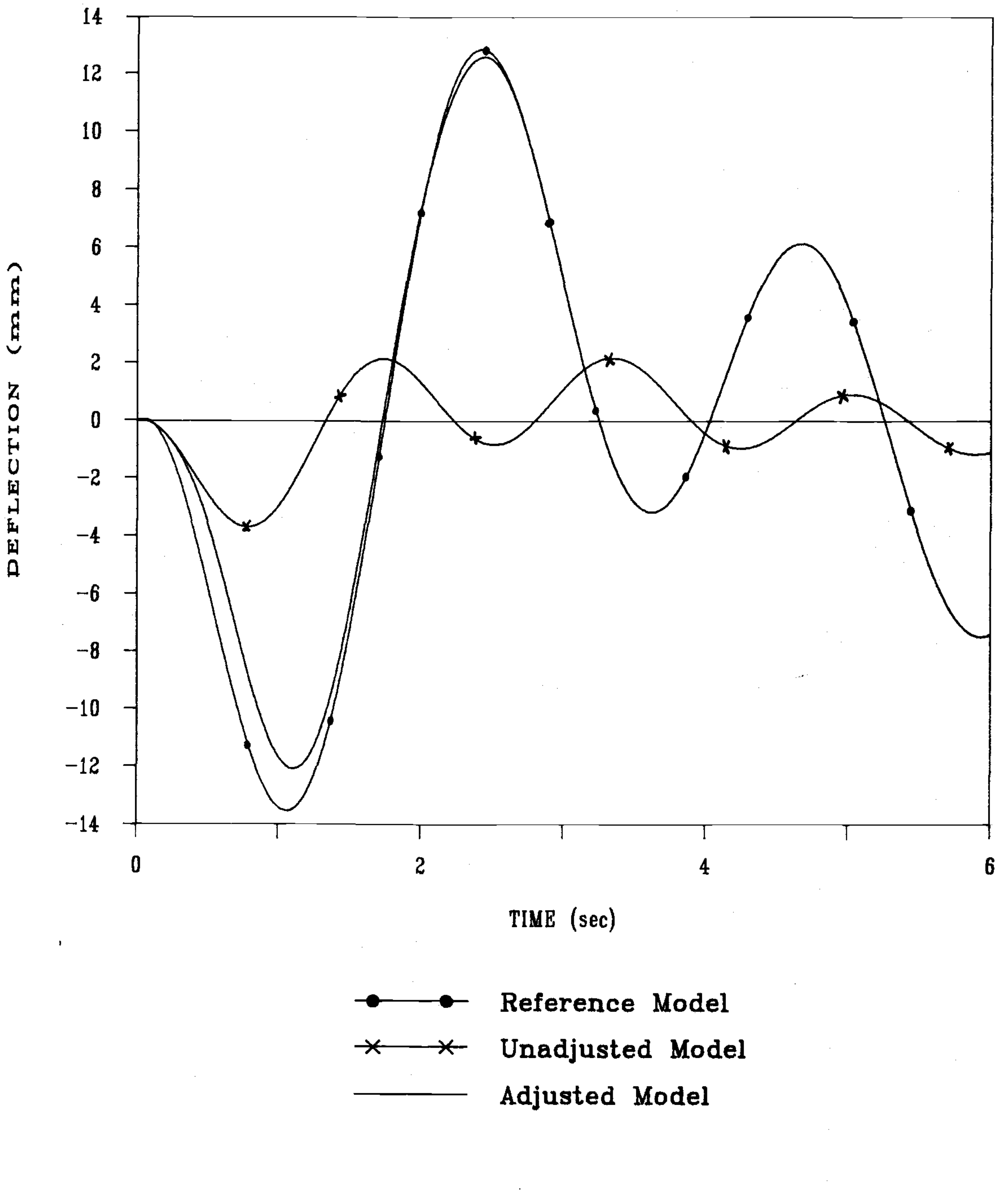


Figure 4.8 Tip Deflection: Integral Adaptive Gain, 0.5; Proportional Adaptive Gain, 1.0

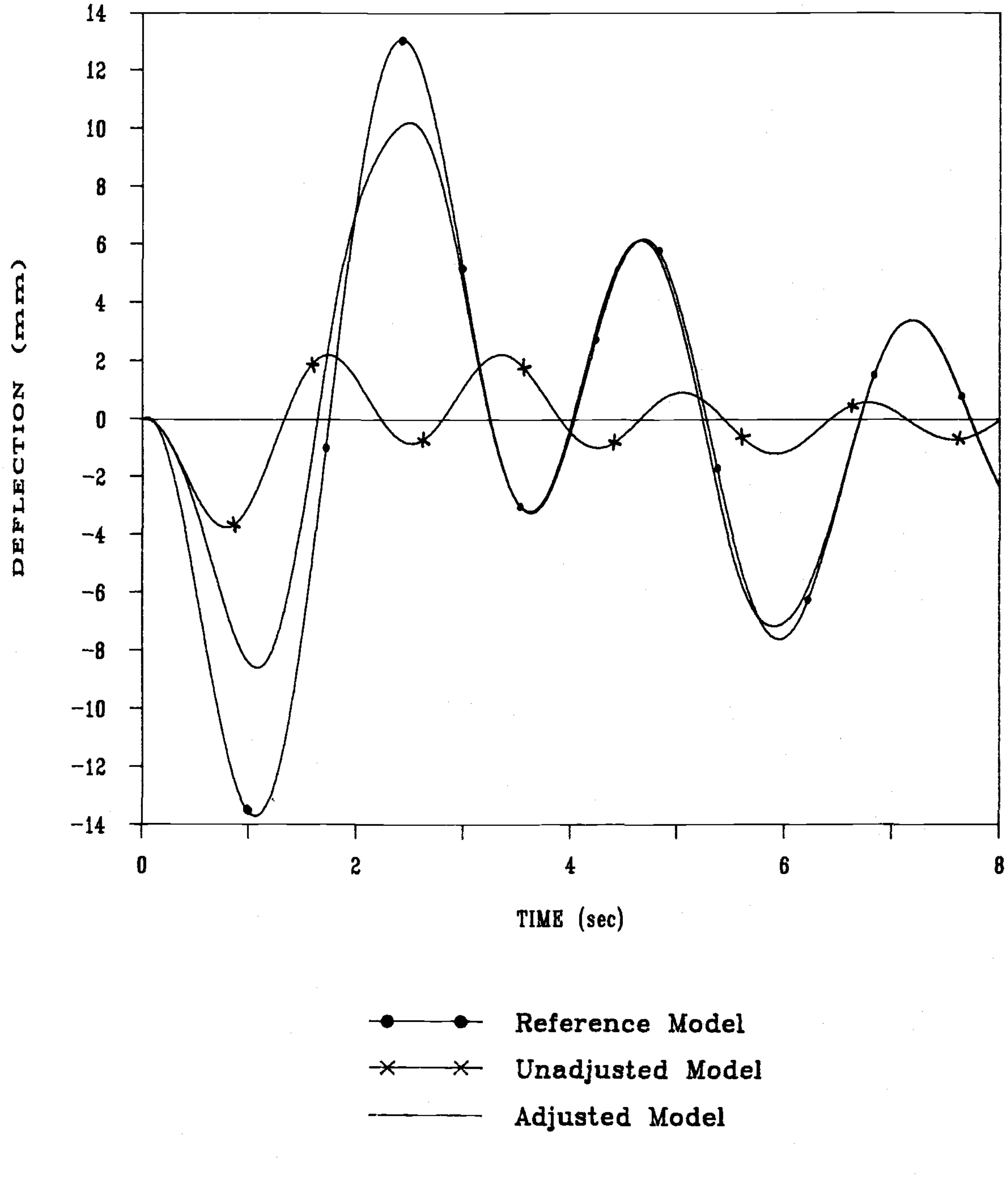


Figure 4.9 Tip Deflection: Integral Adaptive Gain, 0.15; Proportional Adaptive Gain, 0.4

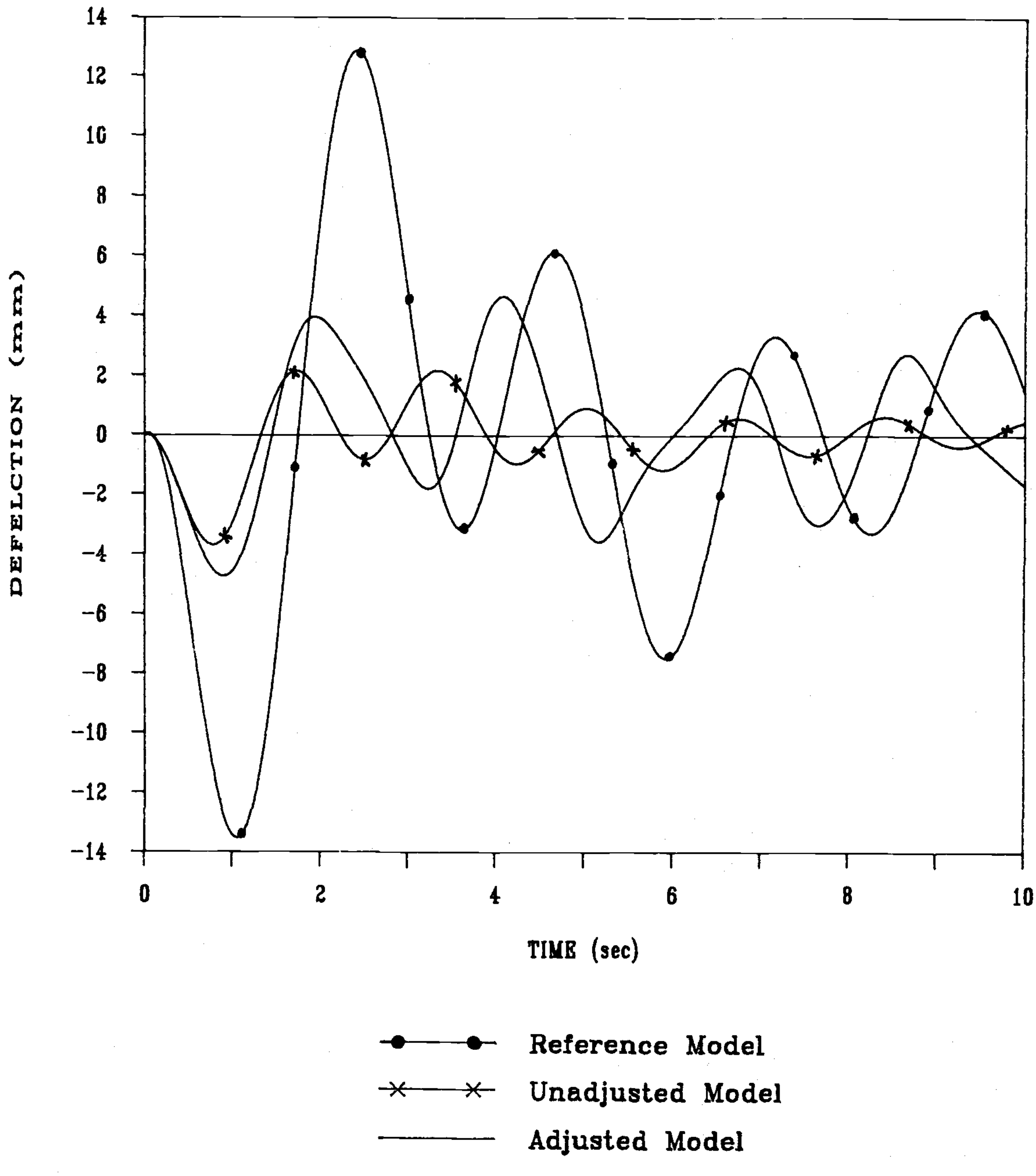


Figure 4.10 Tip Deflection: Integral Adaptive Gain, 0.05; Proportional Adaptive Gain, 0.1

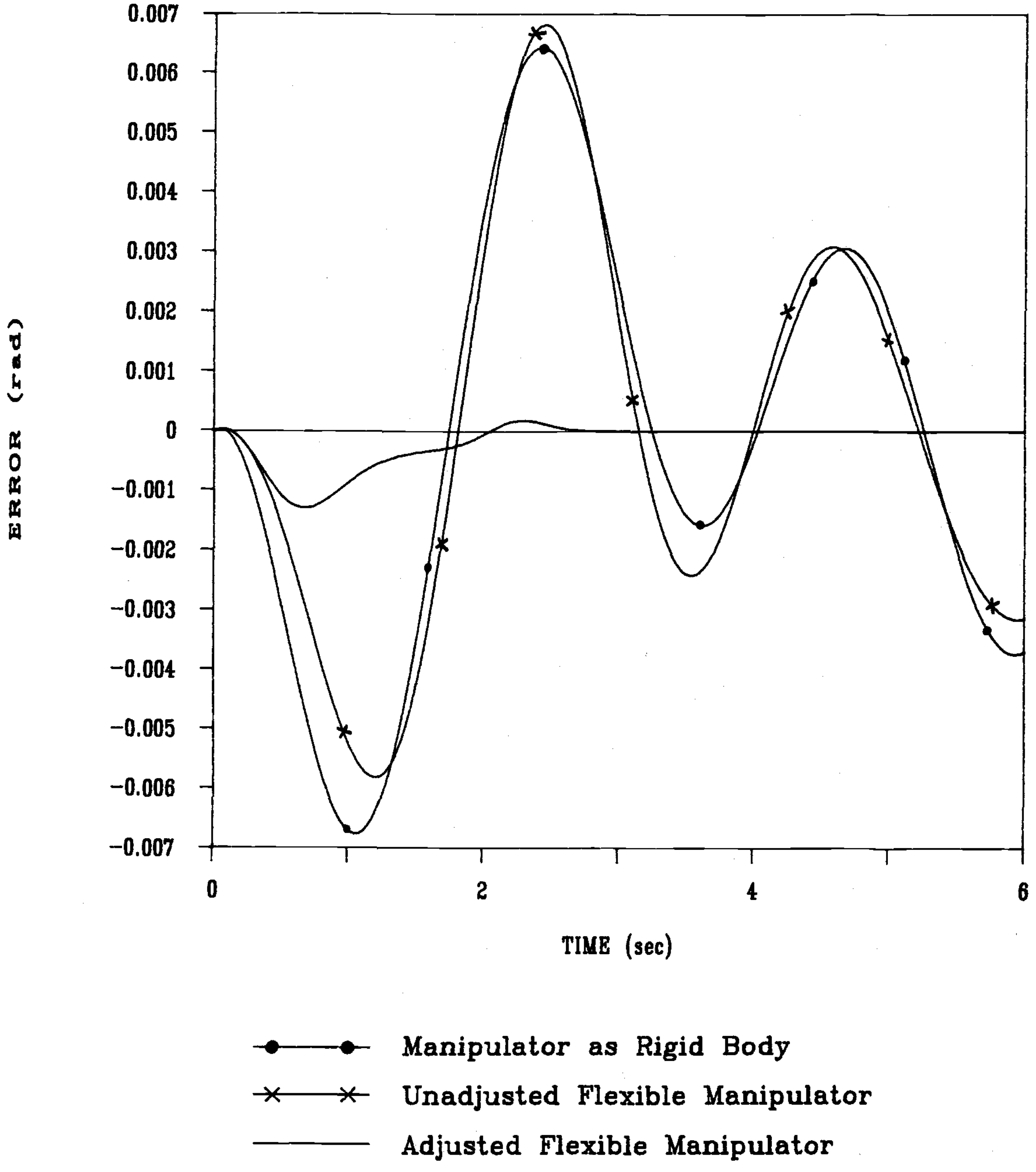


Figure 4.11 Tip Position Error: Integral Adaptive Gain, 0.5; Proportional Adaptive Gain, 1.0

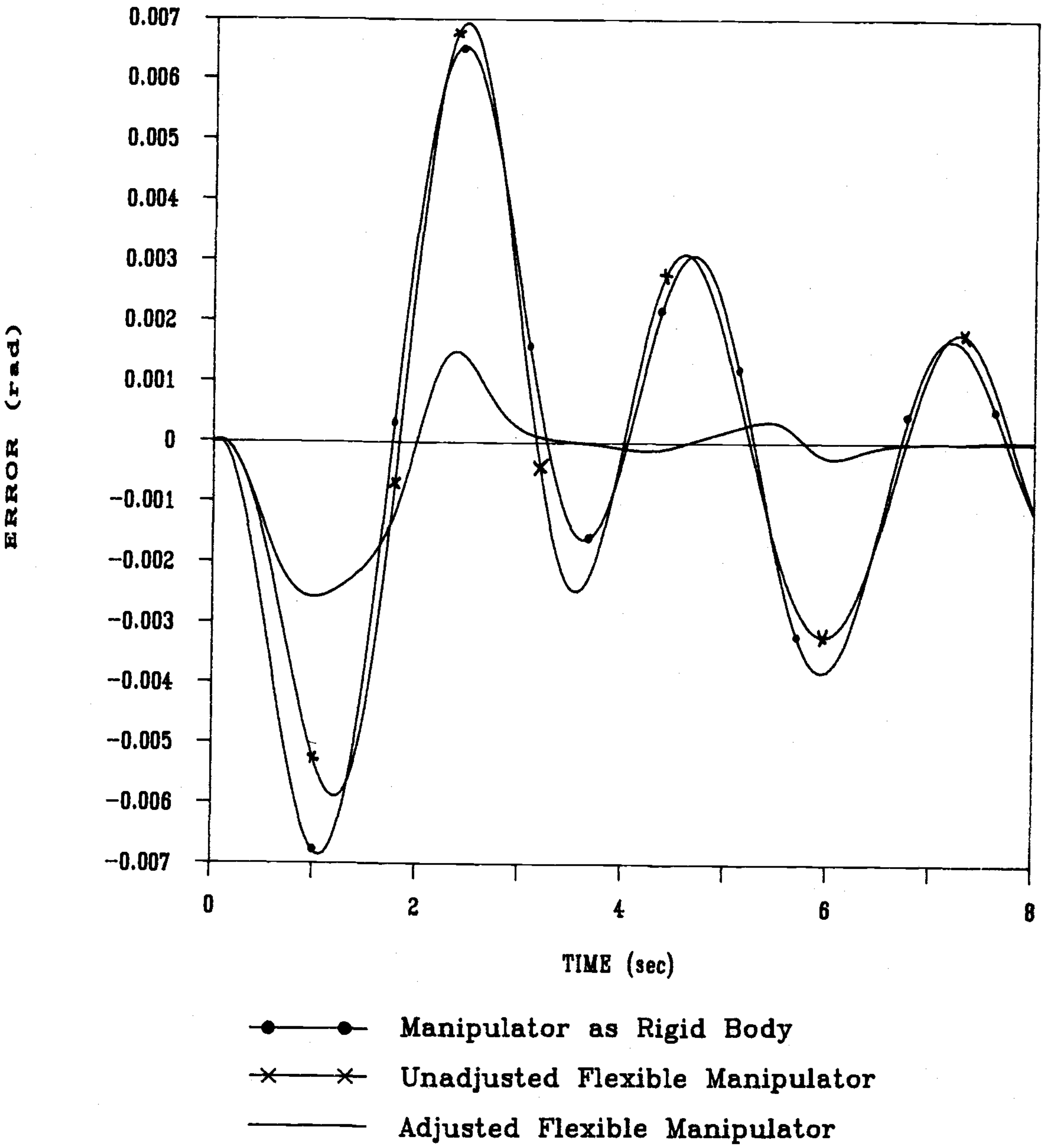


Figure 4.12 Tip Position Error: Integral Adaptive Gain, 0.15; Proportional Adaptive Gain, 0.4

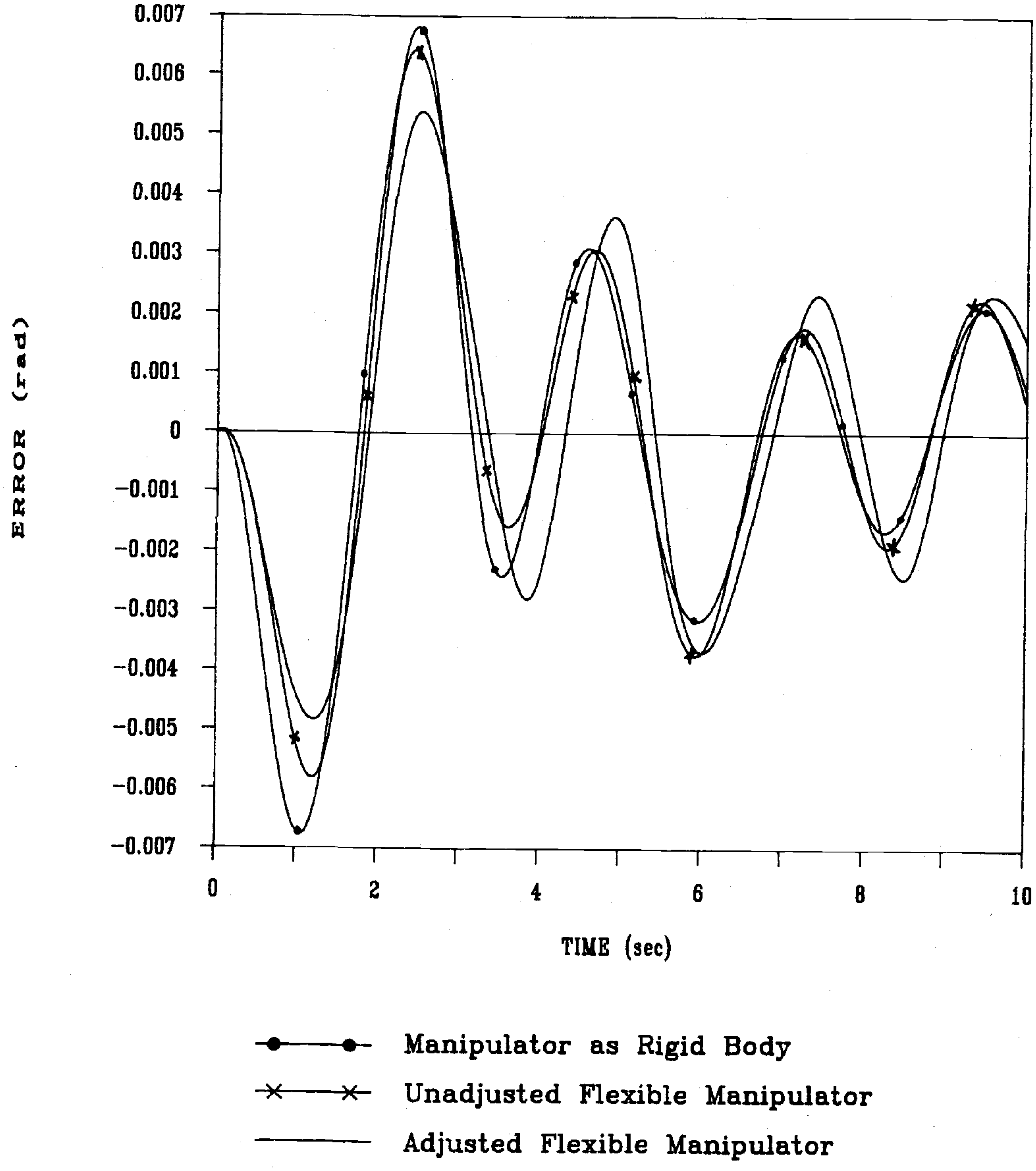


Figure 4.13 Tip Position Error: Integral Adaptive Gain, 0.05; Proportional Adaptive Gain, 0.1

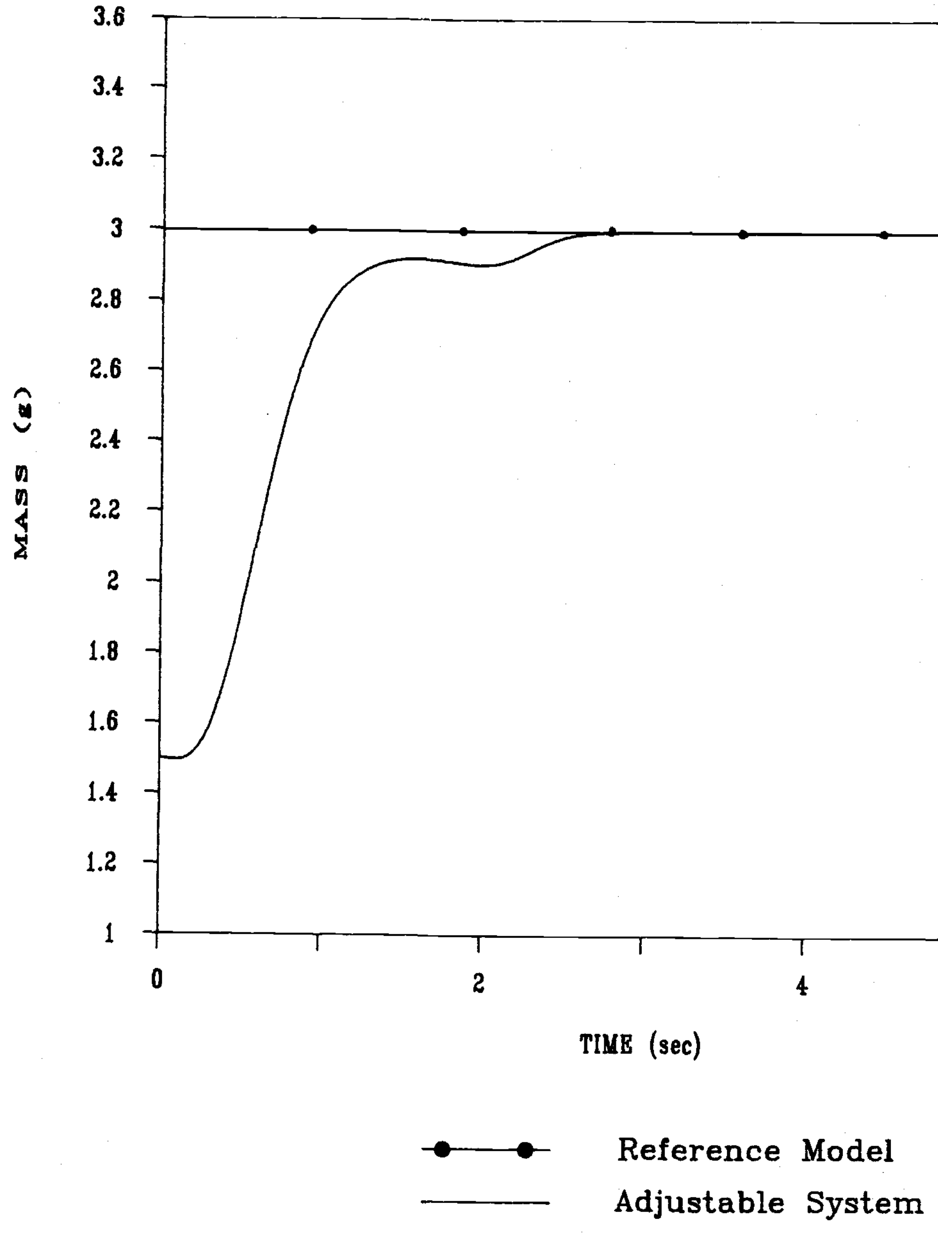


Figure 4.14 Parameter Convergence, Tip Mass: Integral Adaptive Gain, 0.5; Proportional Adaptive Gain, 1.0

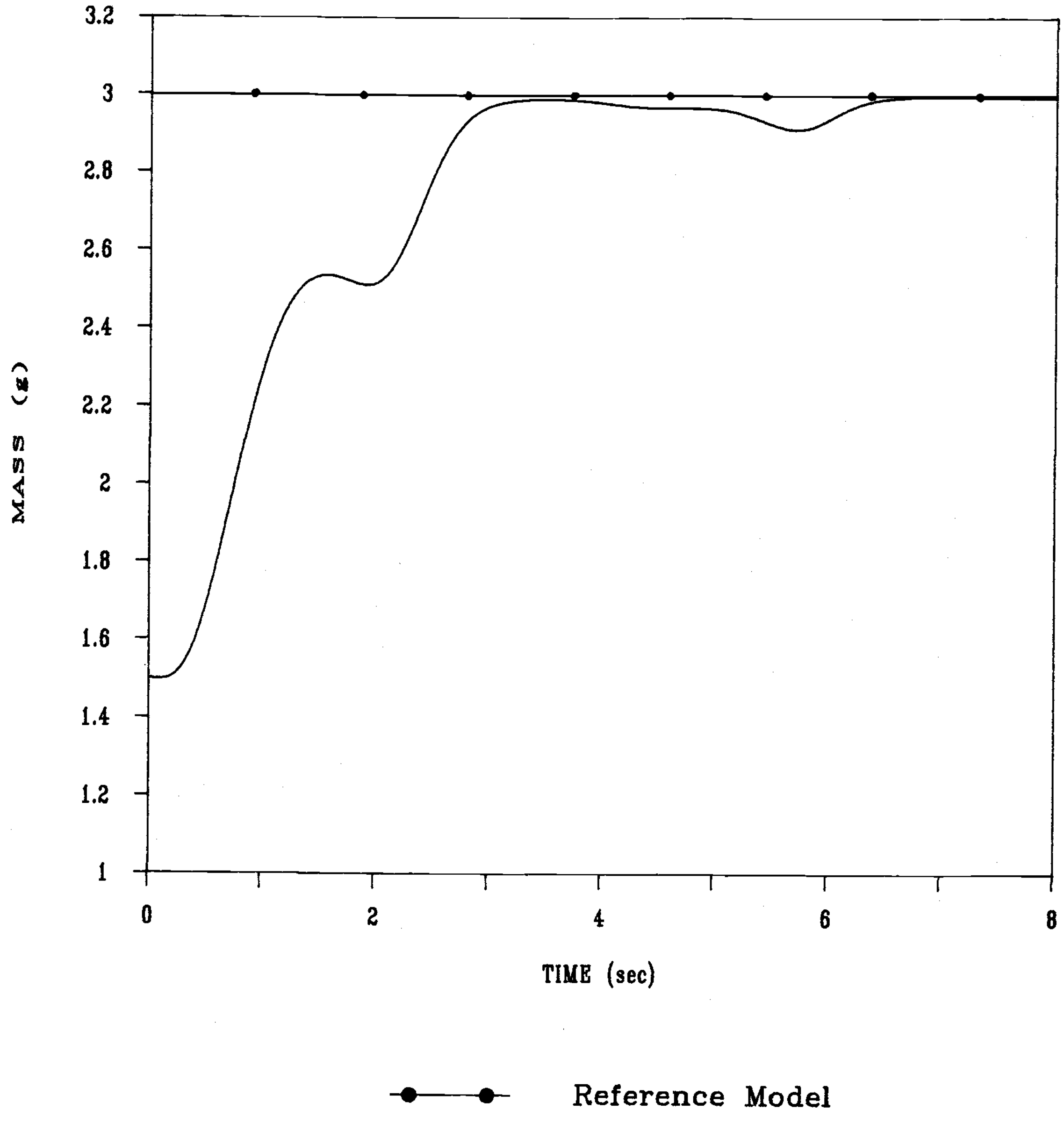


Figure 4.15 Parameter Convergence, Tip Mass: Integral Adaptive Gain, 0.15; Proportional Adaptive Gain, 0.4

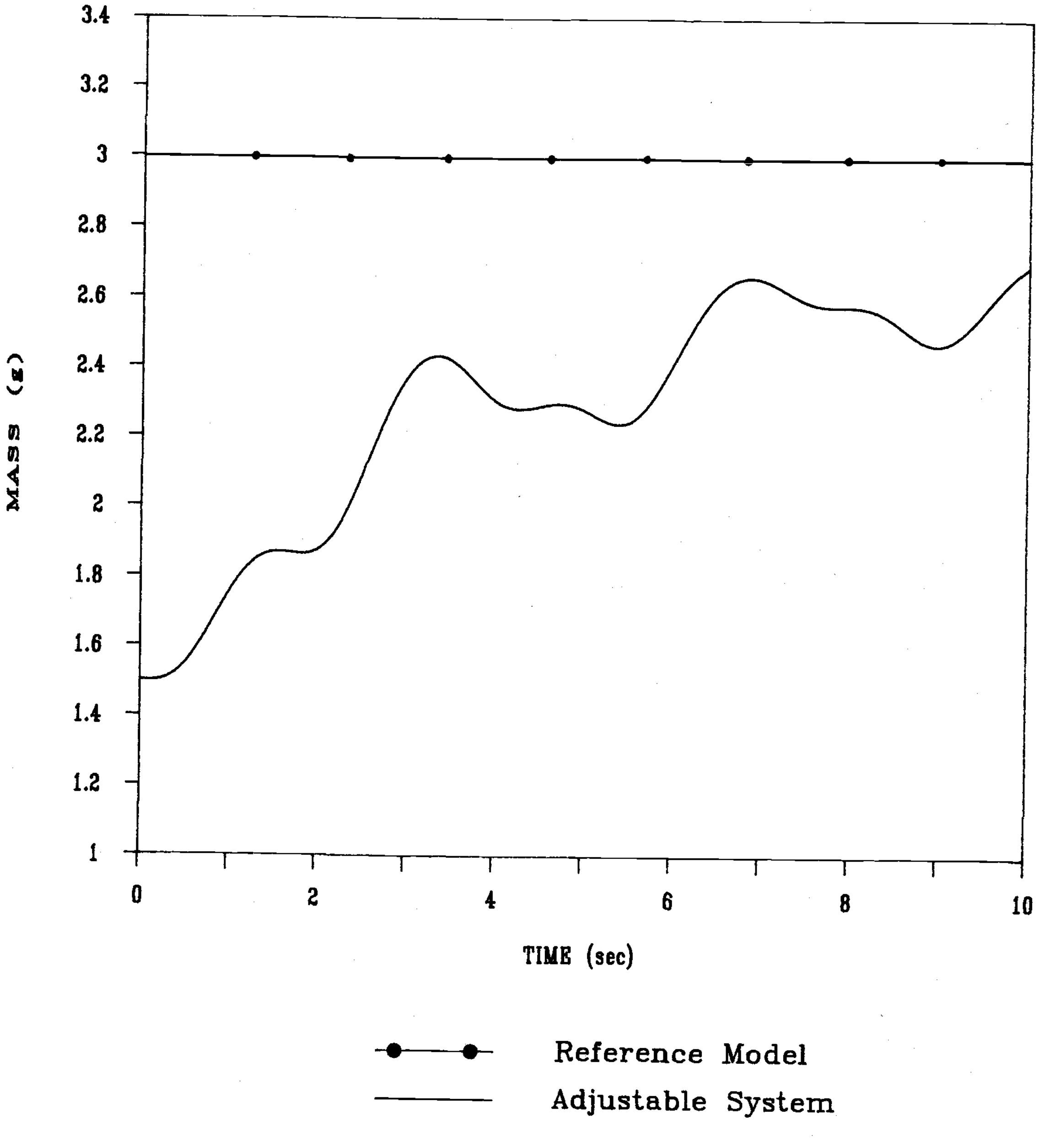


Figure 4.16 Parameter Convergence: Integral Adaptive Gain, 0.05; Proportional Adaptive Gain, 0.1

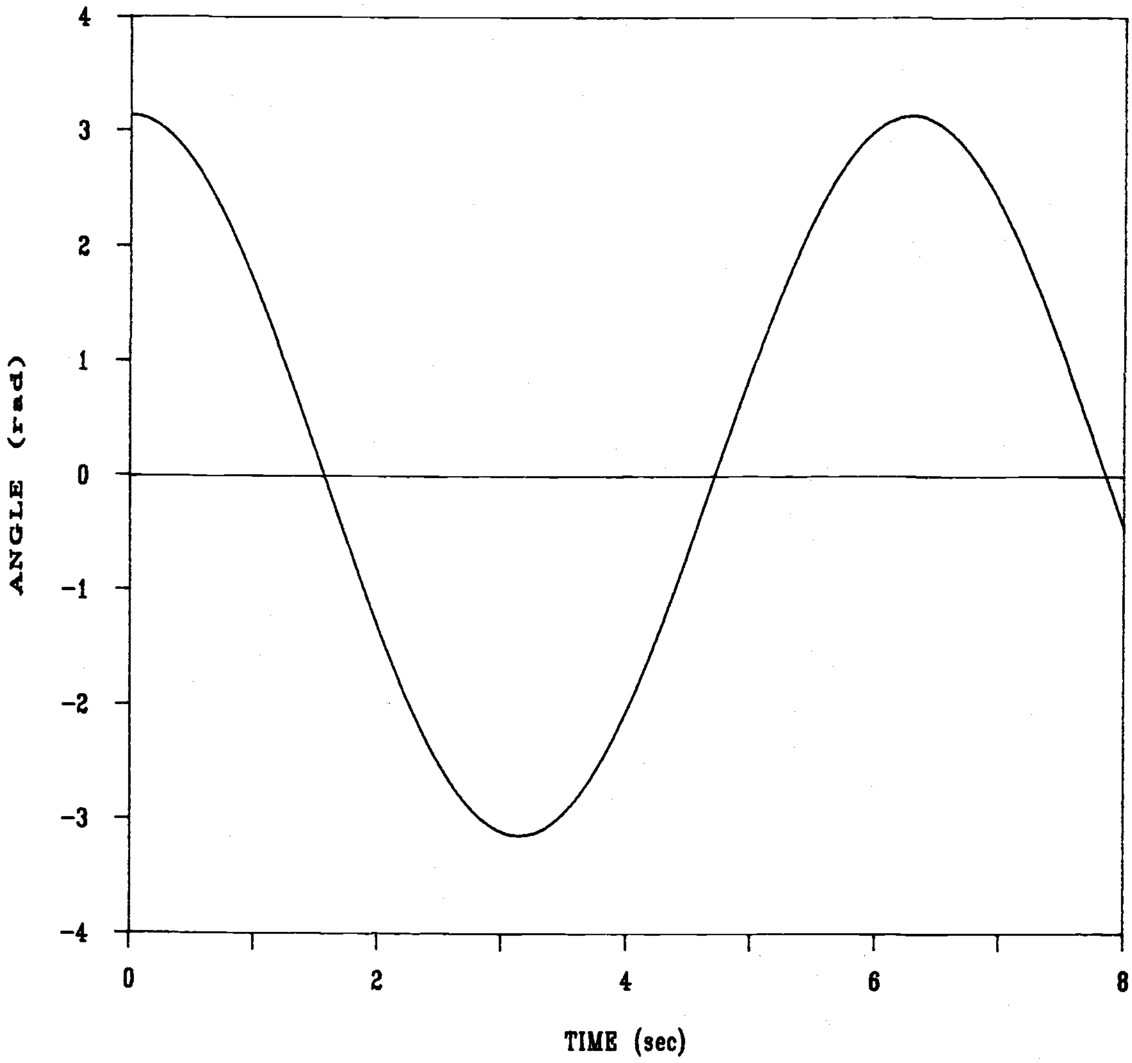


Figure 4.17 Prescribed Motion Base

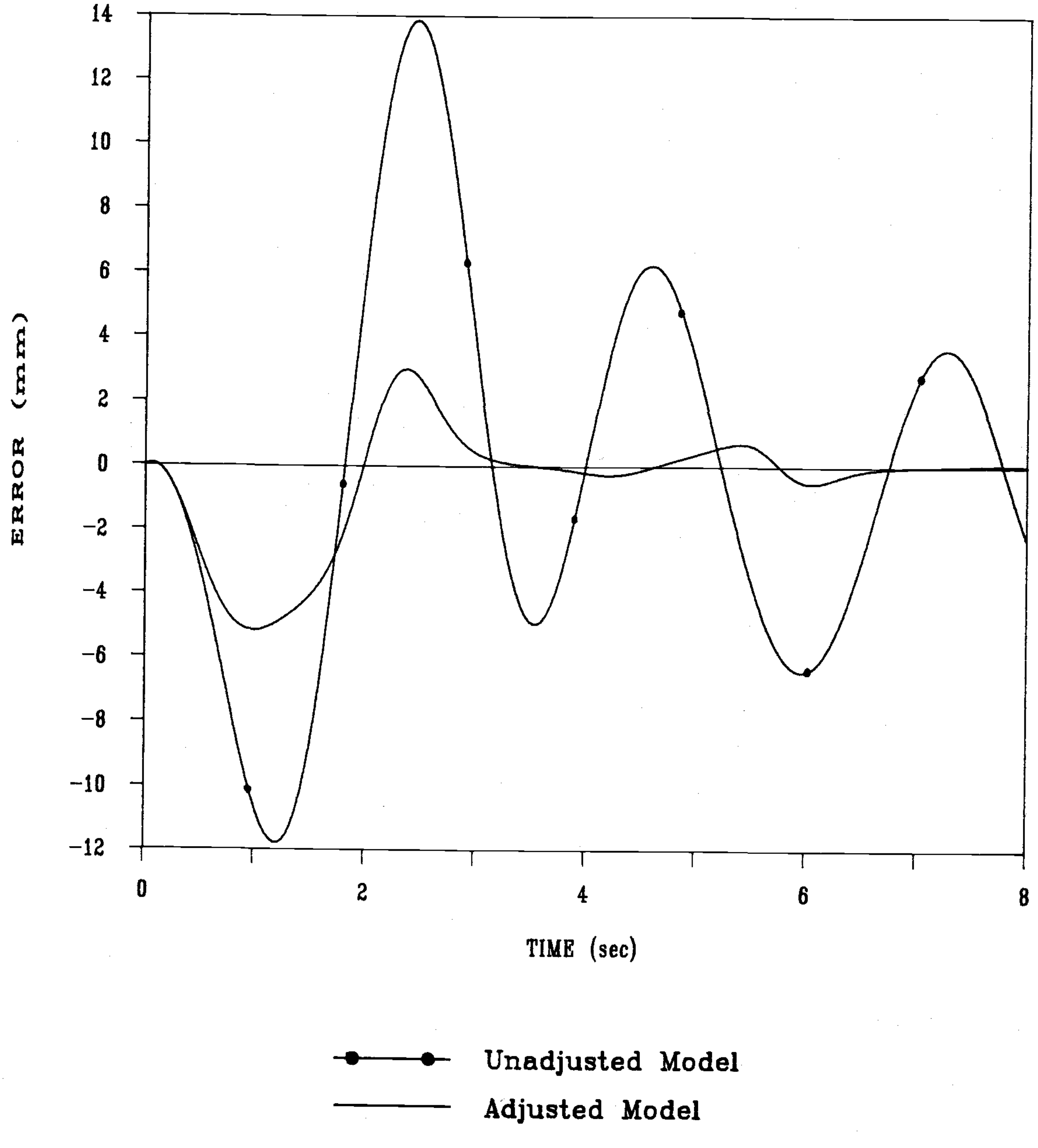


Figure 4.18 Tip Position Error: Integral Adaptive Gain, 0.15; Proportional Adaptive Gain, 0.4

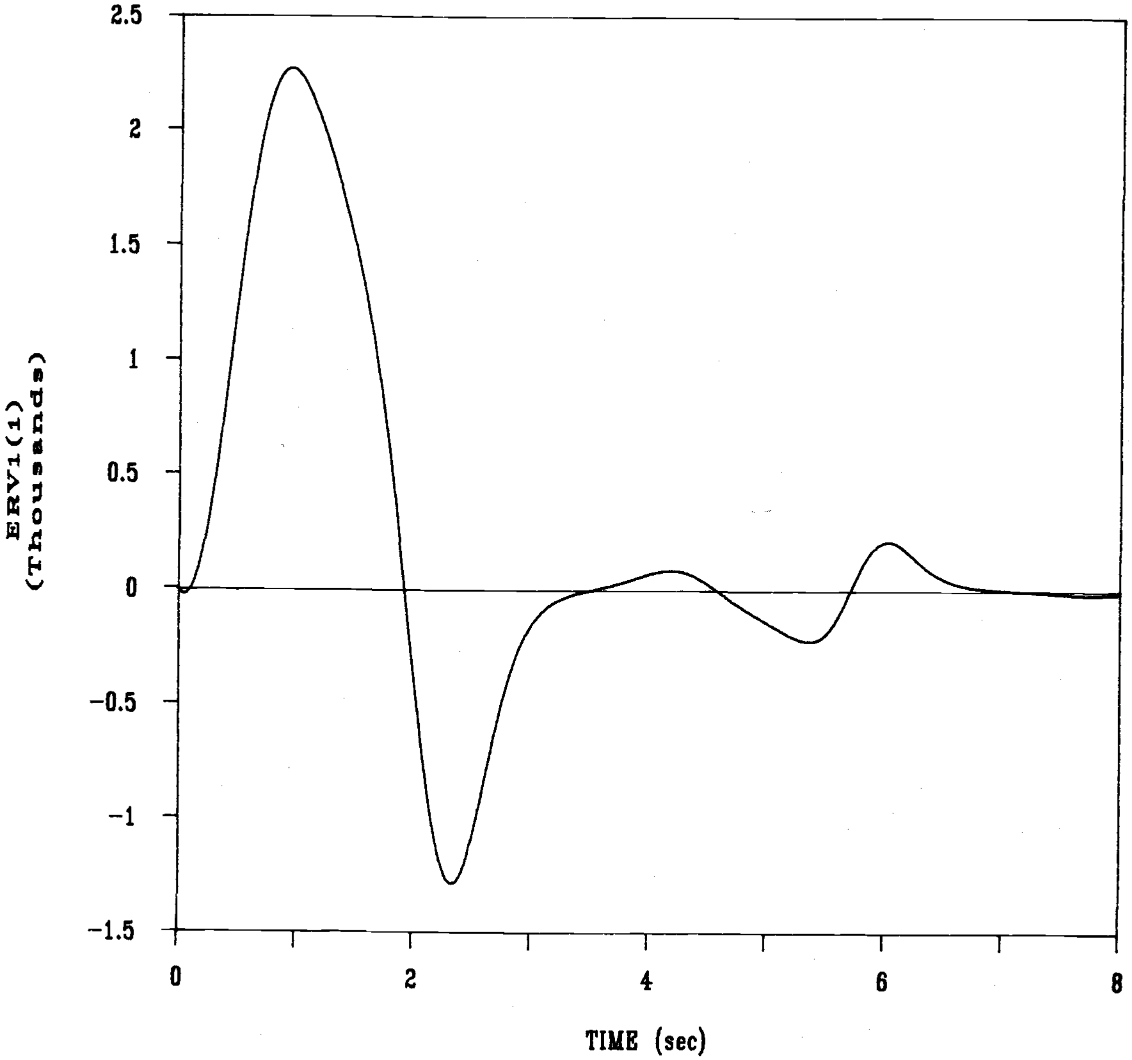


Figure 4.19 Transition of Position Error in Principal State for Mode (1)

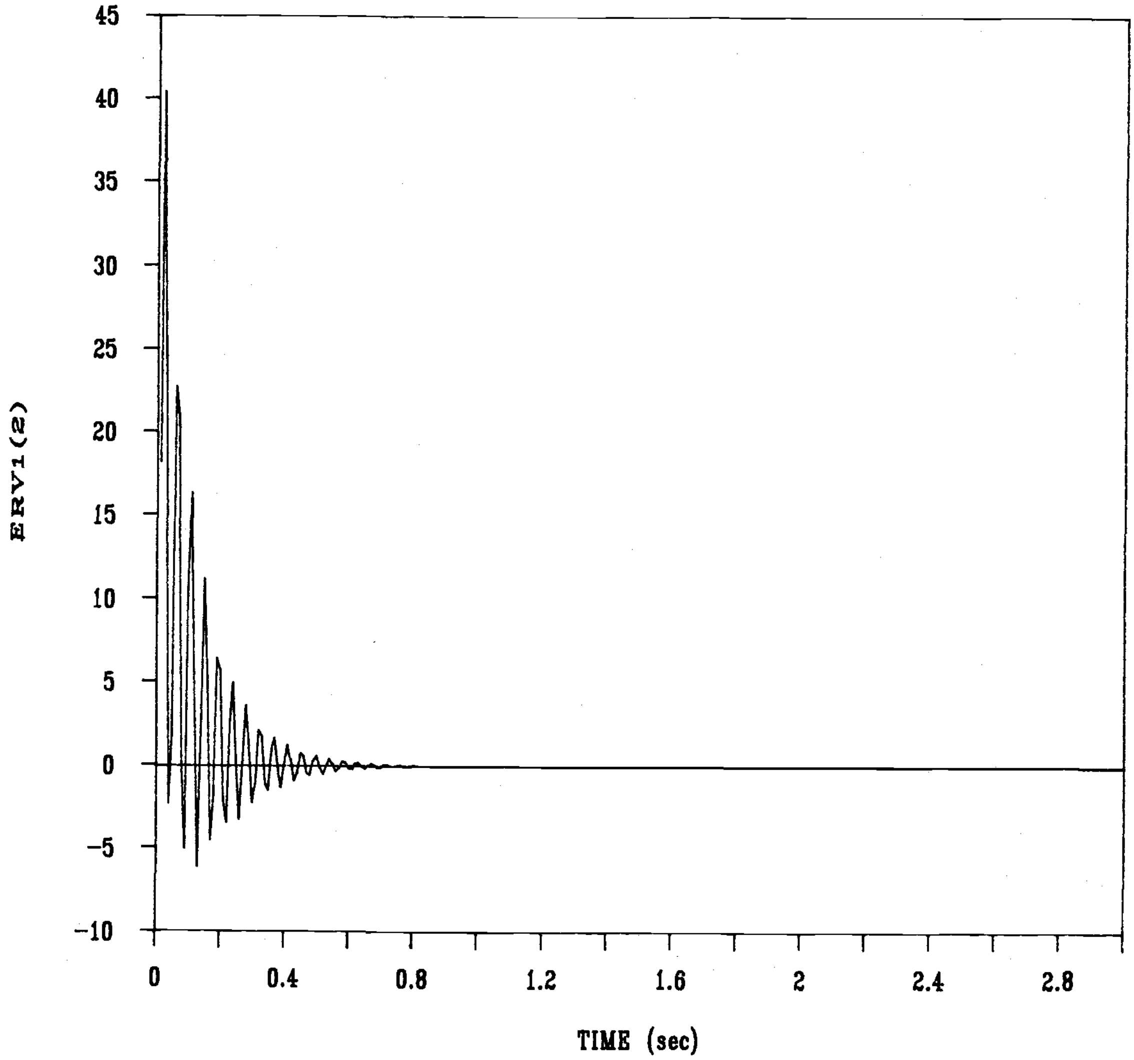


Figure 4.20 Transition of Position Error in Principal State for Mode (2)

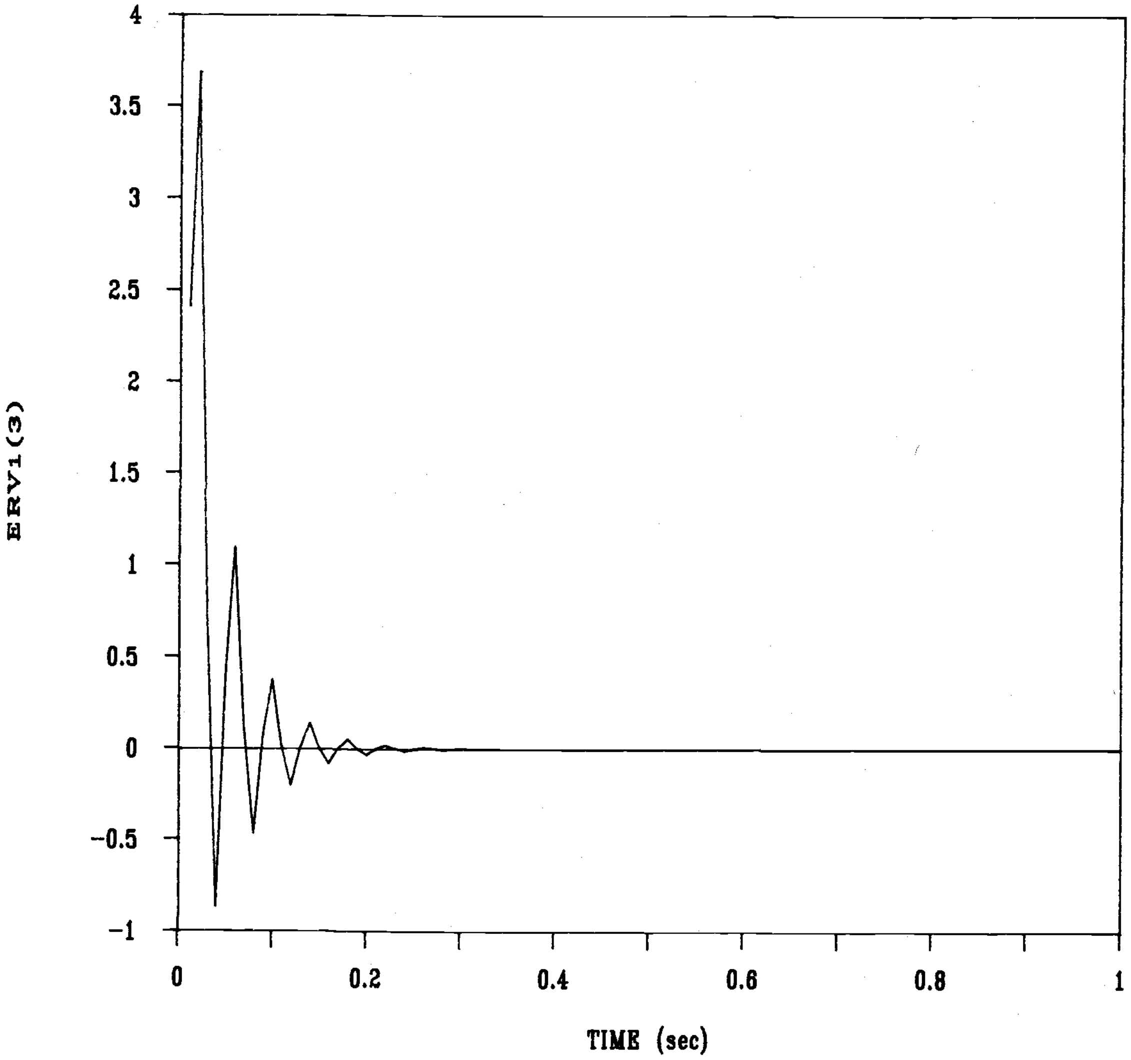


Figure 4.21 Transition of Position Error in Principal State for Mode (3)

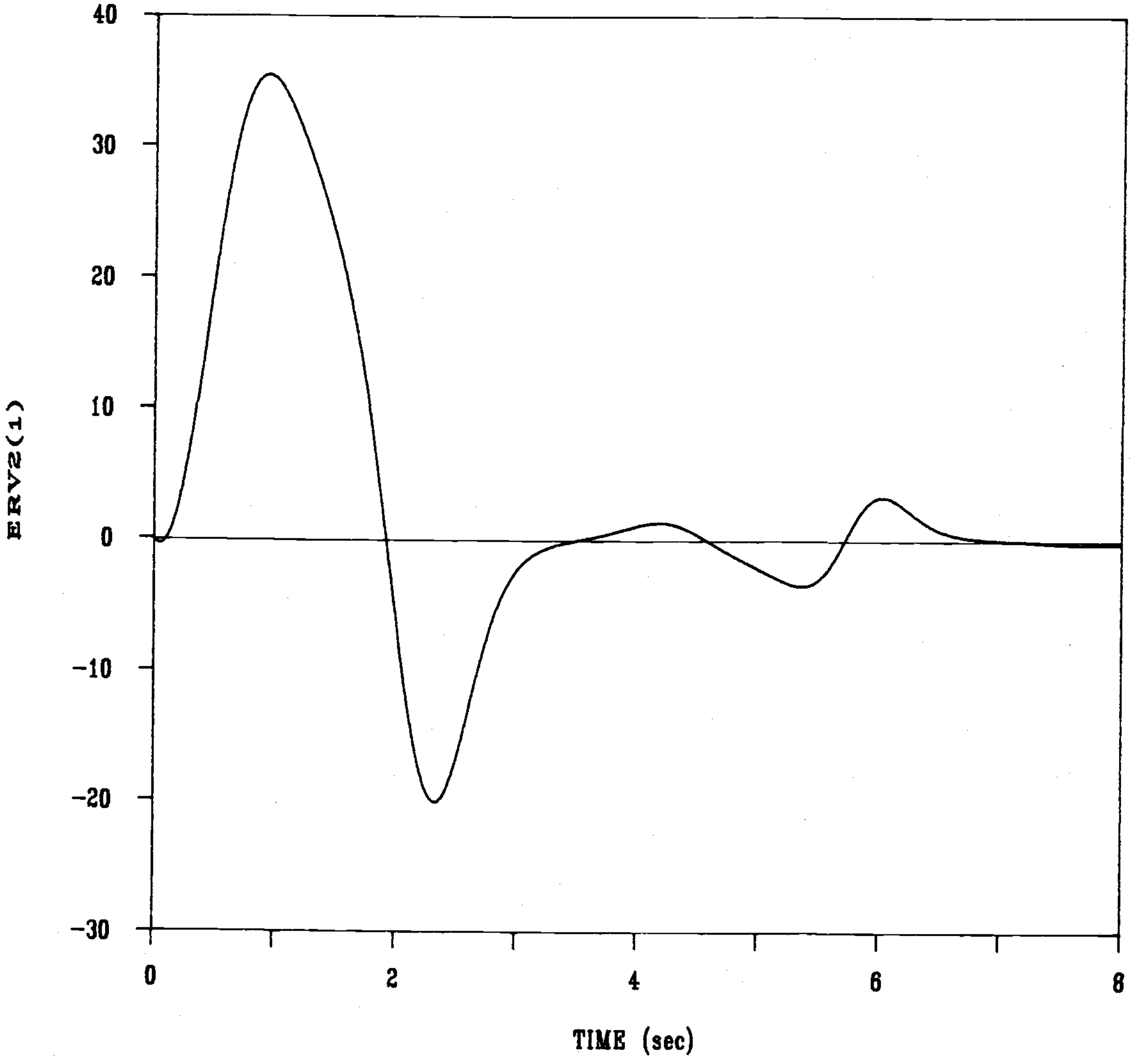


Figure 4.22 Transition of Velocity Error in Principal State for Mode (1)

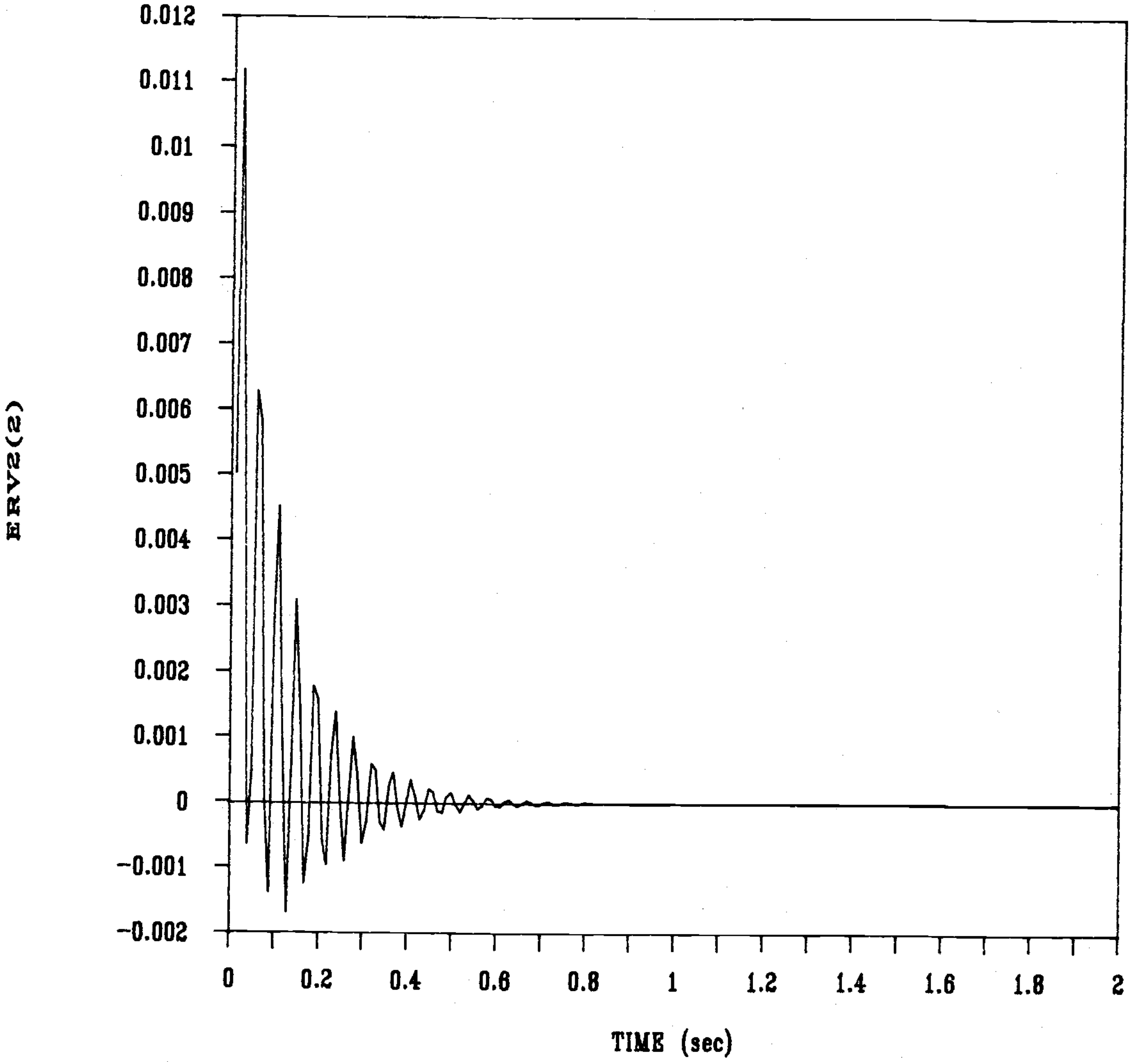


Figure 4.23 Transition of Velocity Error in Principal State for Mode (2)

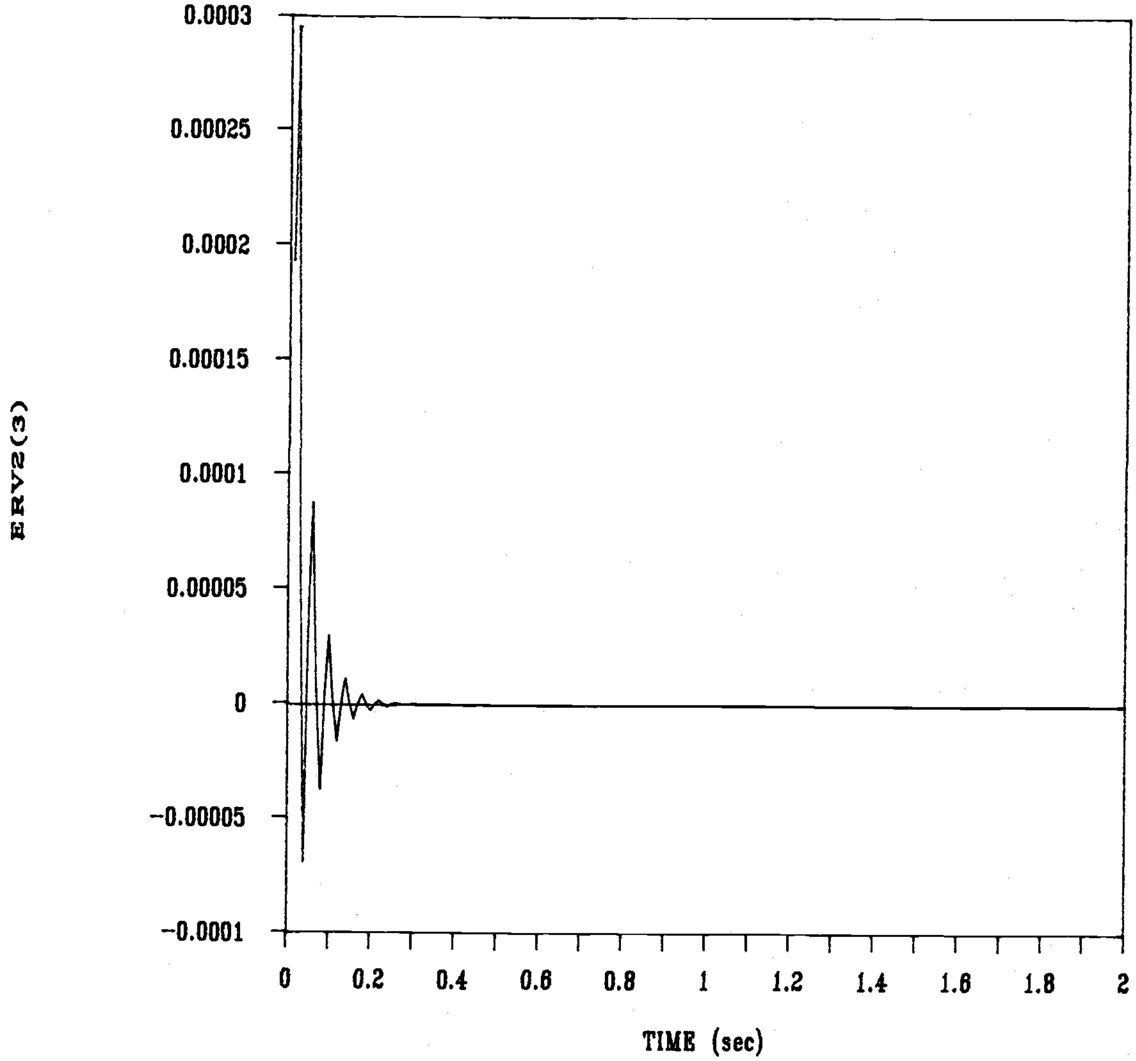


Figure 4.24 Transition of Velocity Error in Principal State for Mode (3)

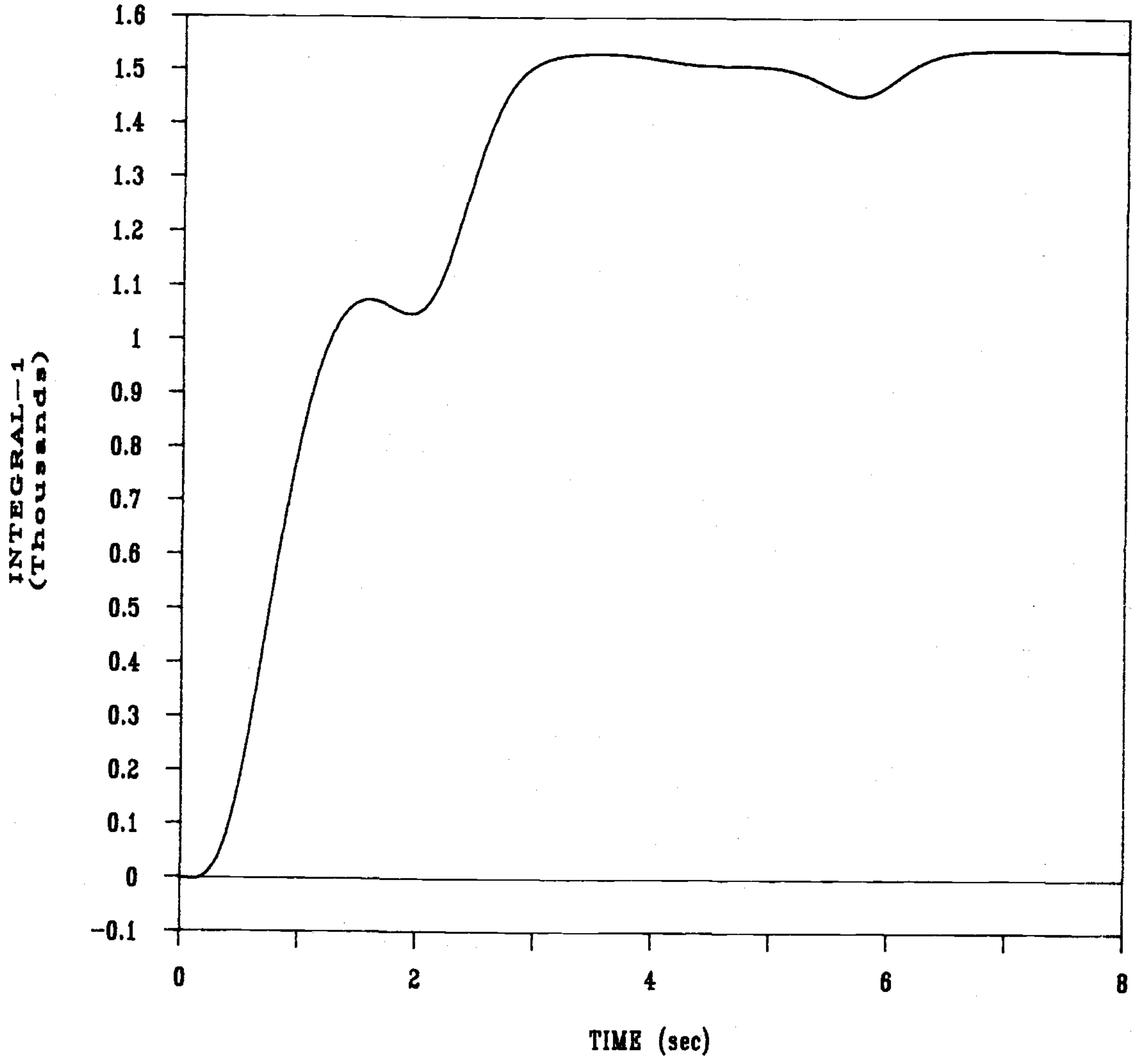


Figure 4.25 Integral Adaptation History
for Mode (1)

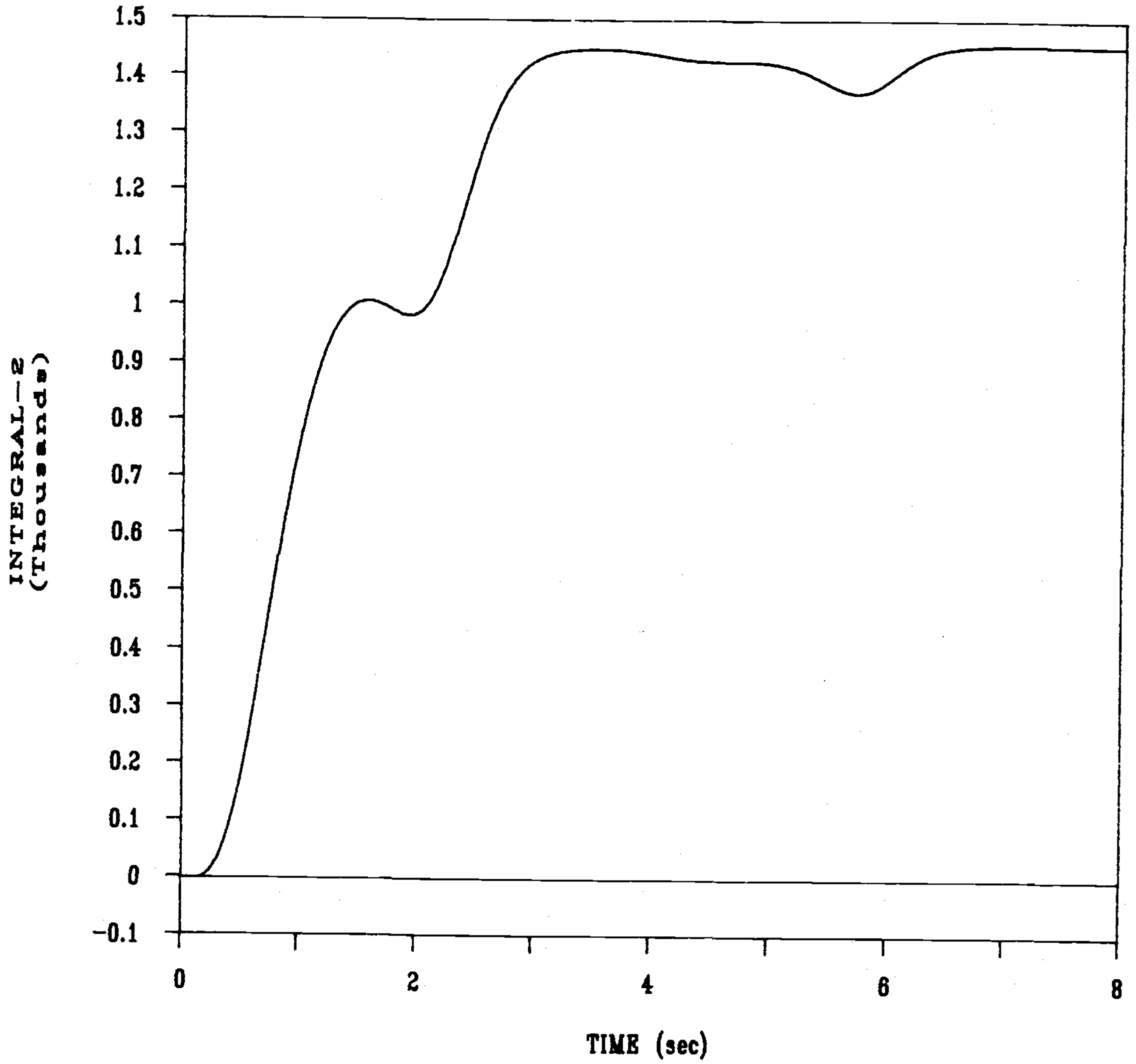


Figure 4.26 Integral Adaptation History
for Mode (2)

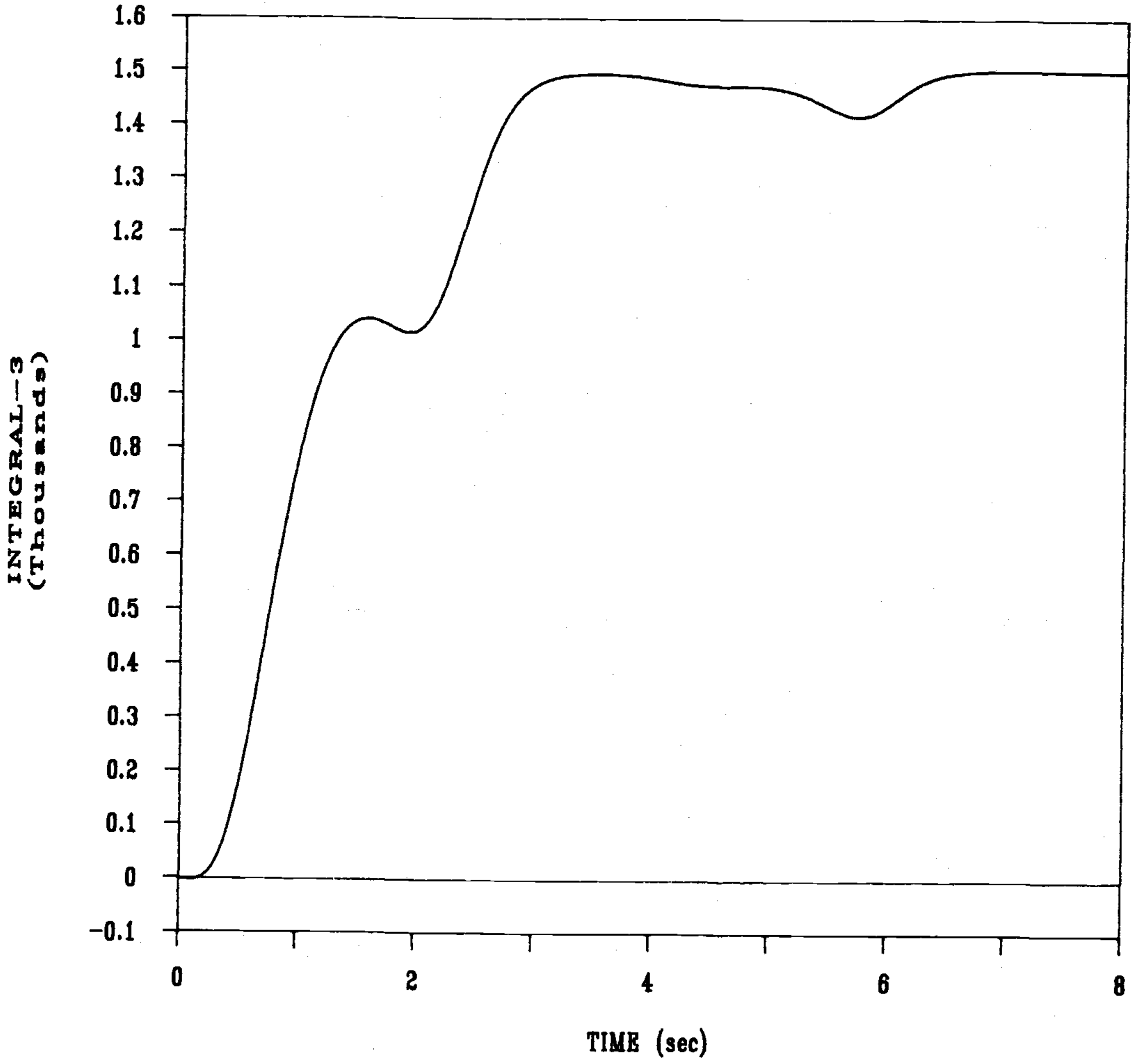


Figure 4.27 Integral Adaptation History for Mode (3)

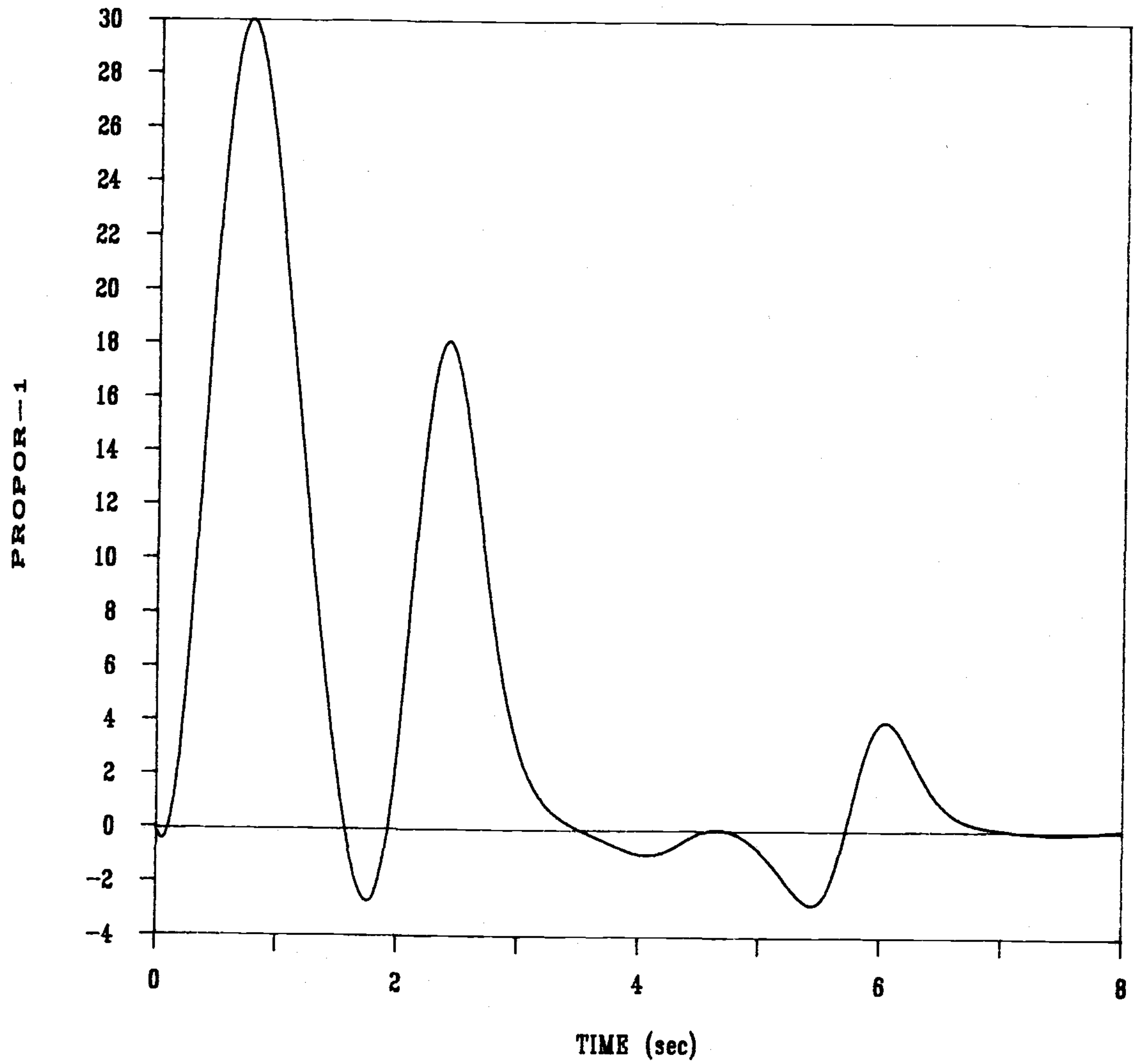


Figure 4.28 Proportional Adaptation History for Mode (1)

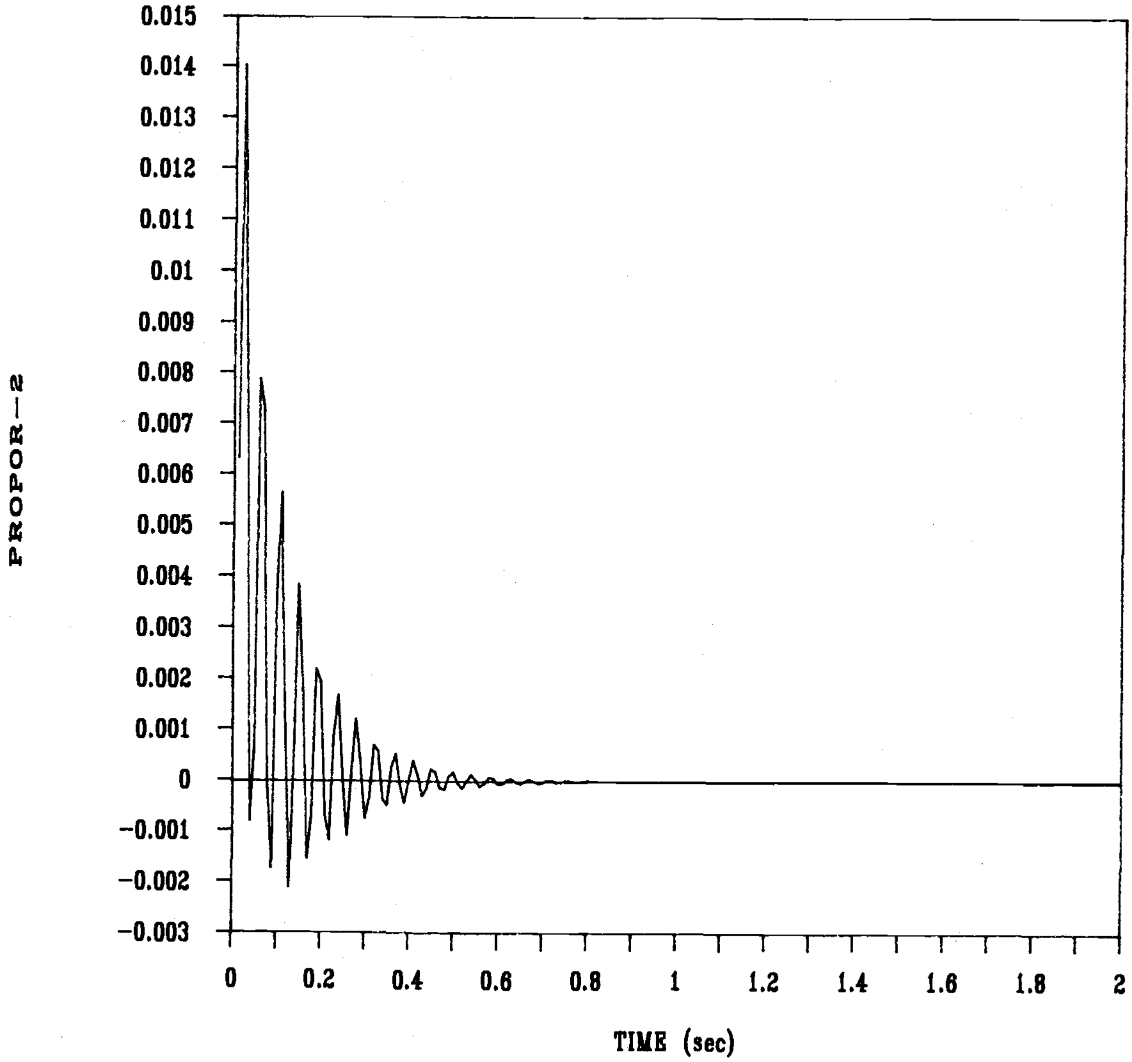


Figure 4.29 Proportional Adaptation History for Mode (2)

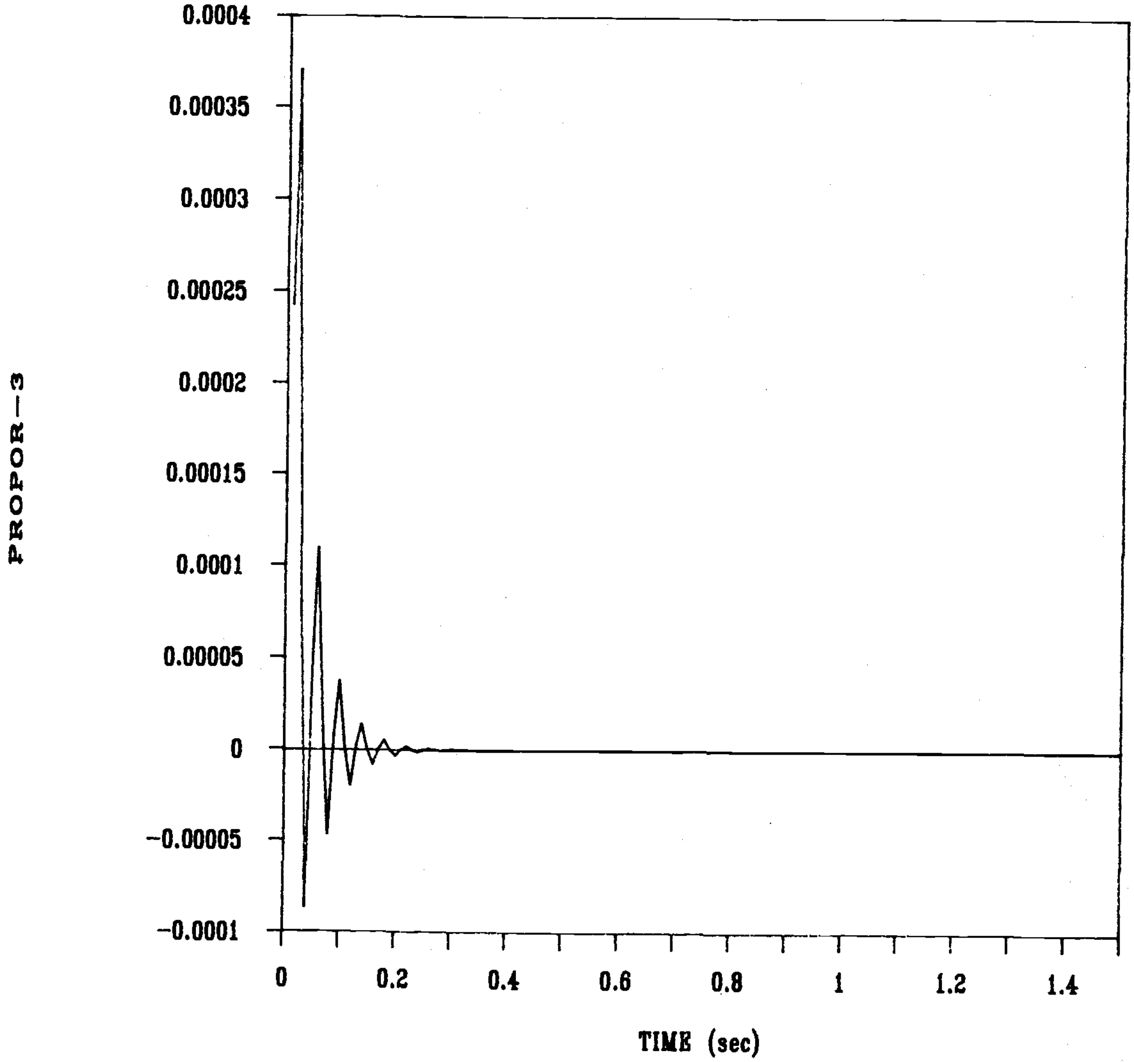


Figure 4.30 Proportional Adaptation History for Mode (3)

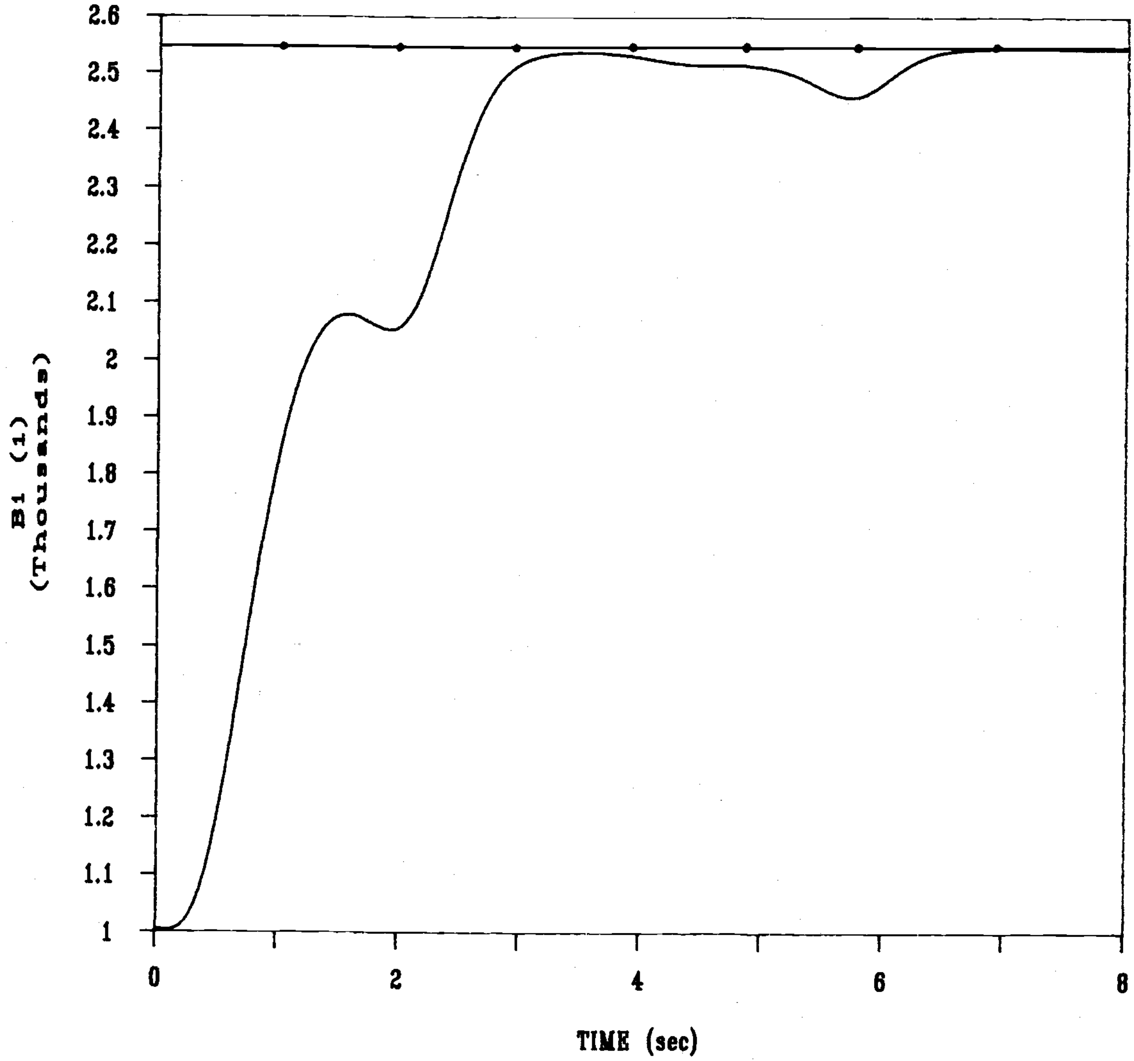


Figure 4.31 Parameter Convergence:
B_1 in Principal State

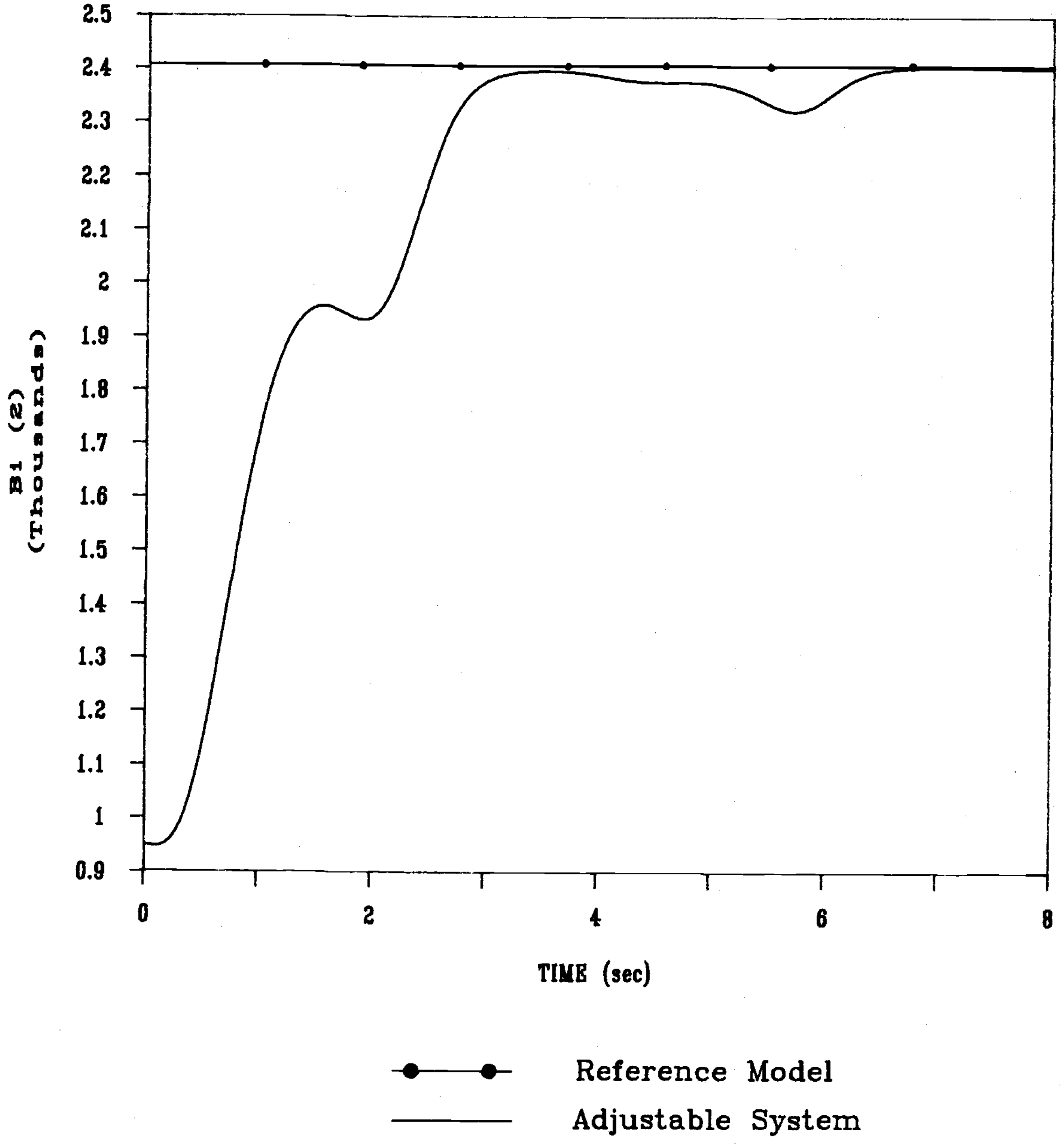


Figure 4.32 Parameter Convergence:
B_2 in Principal State

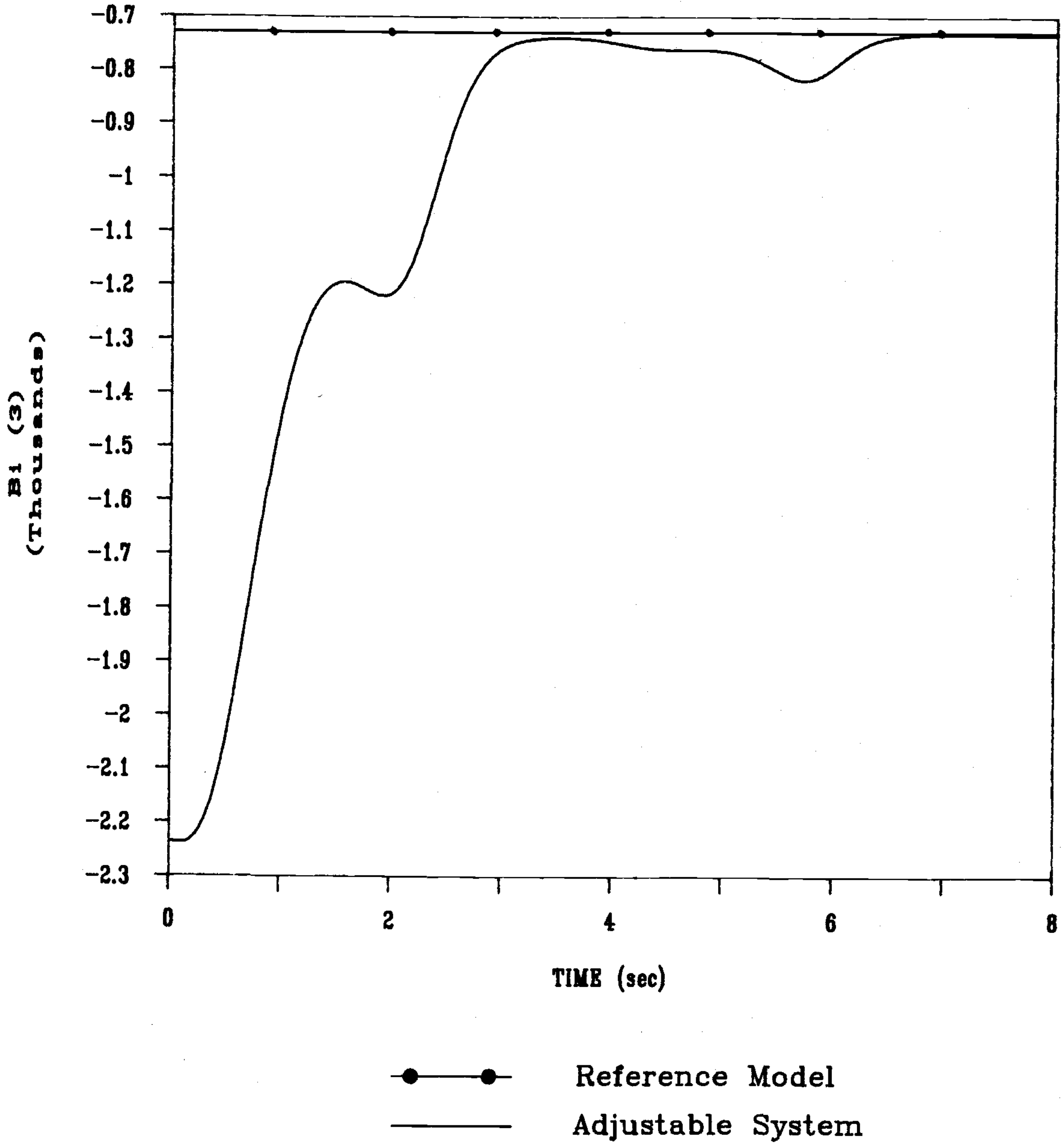


Figure 4.33 Parameter Convergence:
B_3 in Principal State

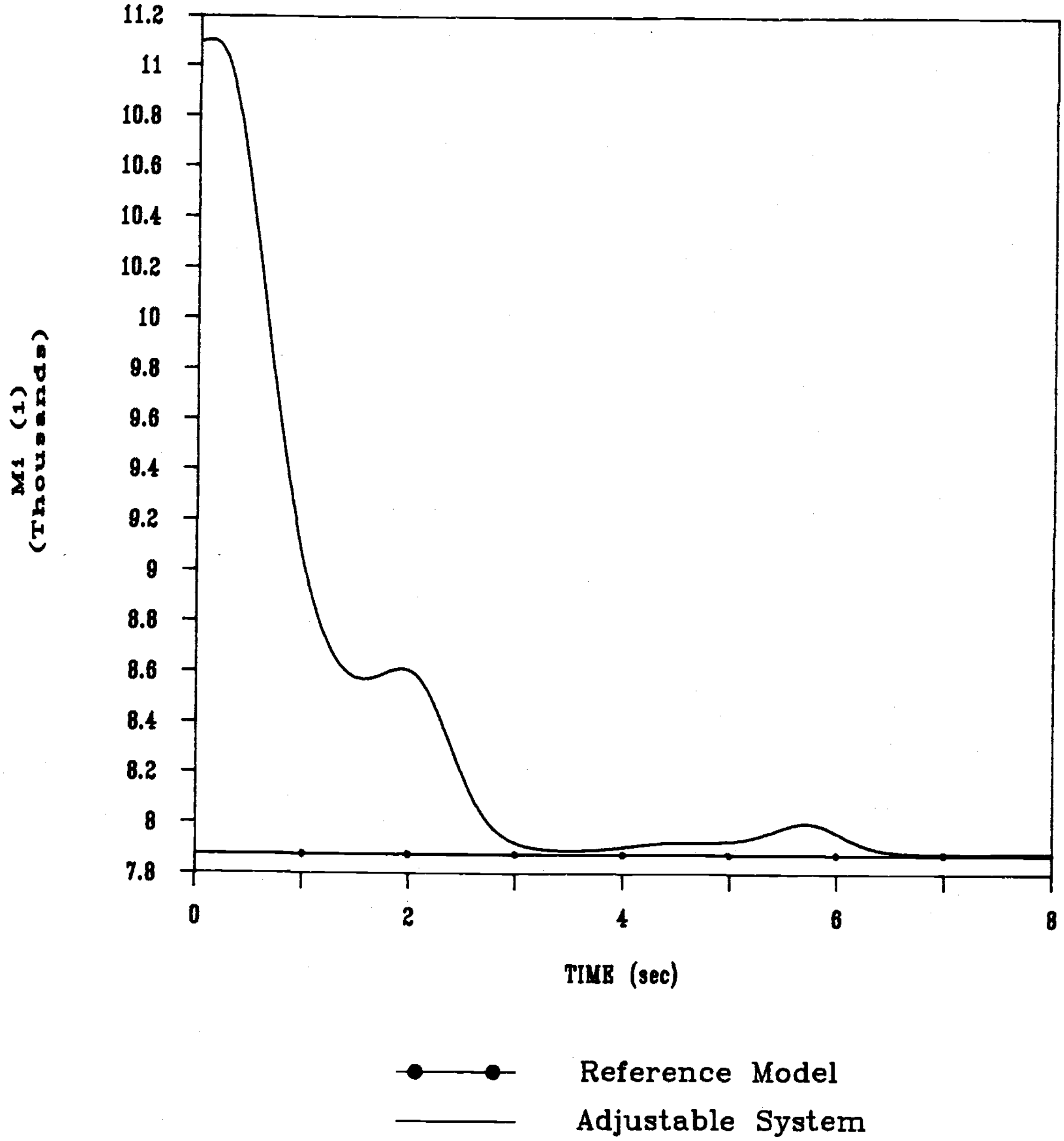


Figure 4.34 Parameter Convergence:
M_1 in Principal State

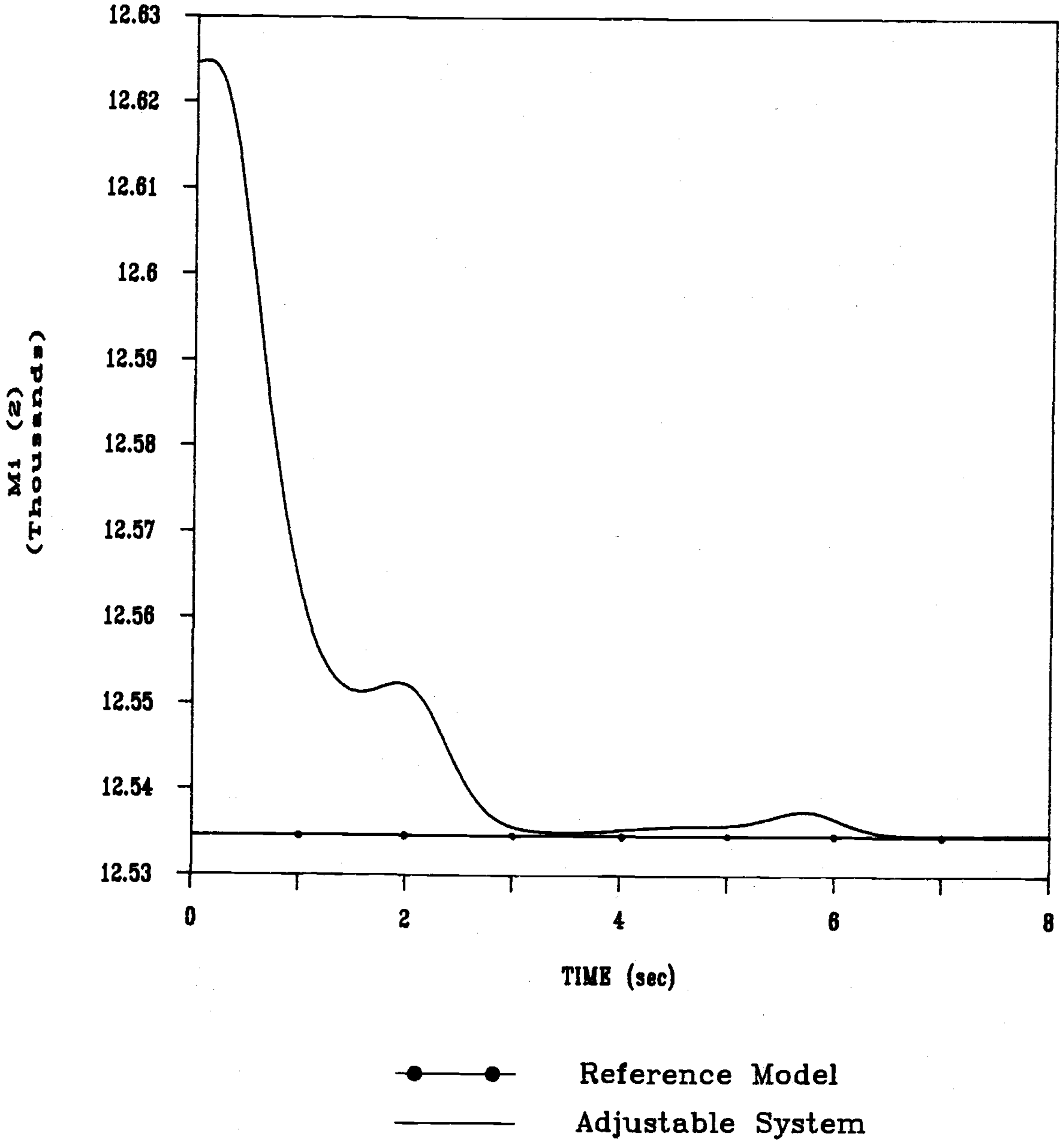


Figure 4.35 Parameter Convergence:
M_2 in Principal State

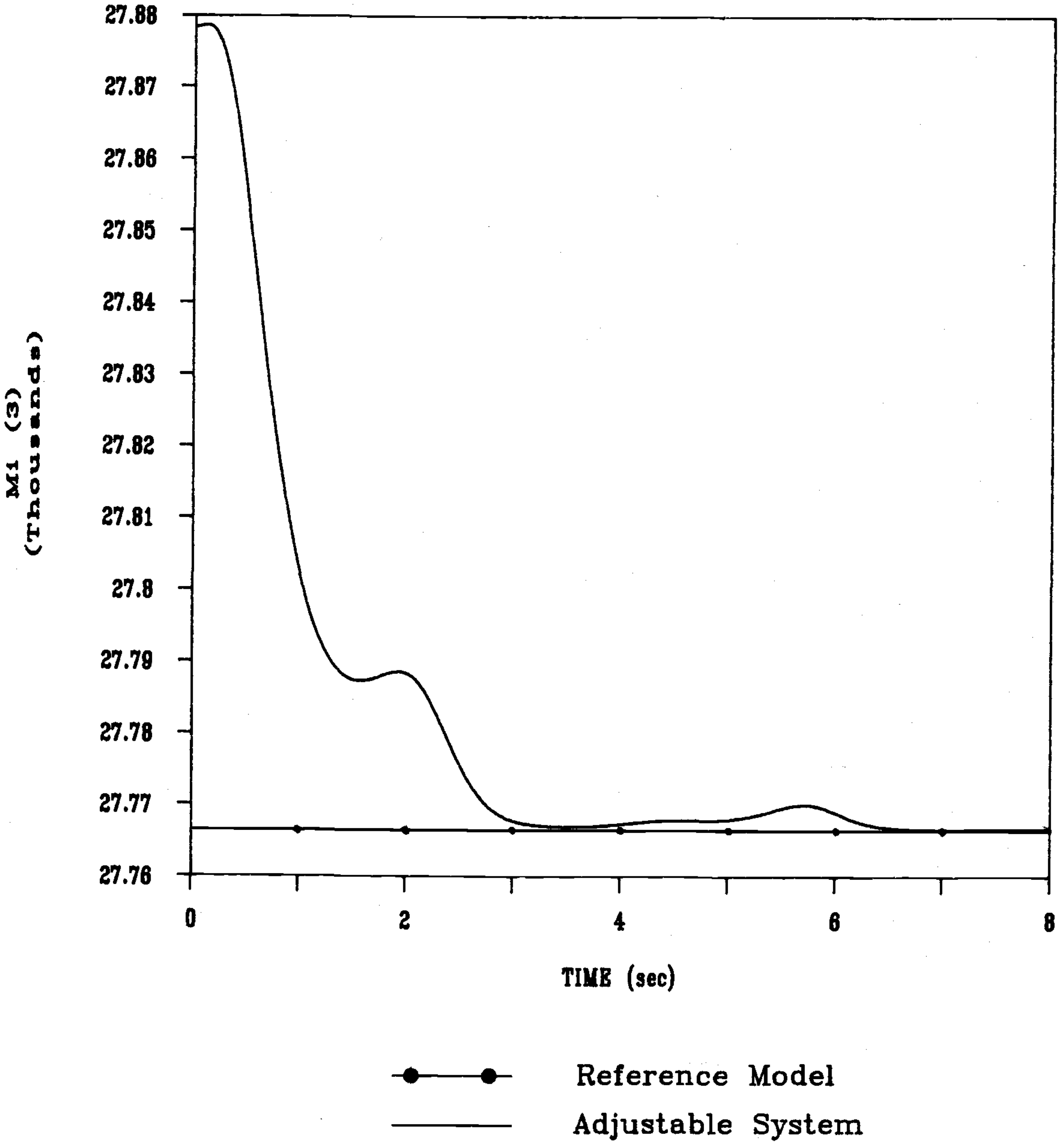


Figure 4.36 Parameter Convergence:
M_3 in Principal State

V. CONCLUSION

The assumed-mode method for the mathematical modeling of the flexible manipulator has been examined. The equations of motion in generalized coordinates for a one-link flexible manipulator have been obtained. Furthermore, based upon the assumption of Rayleigh damping factors, the equations of motion for a flexible manipulator with damping factors can be transformed into the principal coordinates. This procedure for the approximation of a flexible manipulator has been presented as an application of the model reference adaptive system.

The general concepts underlying the model reference adaptive system has been reviewed and a modified control scheme has been developed in order that mathematical representation obtained by the assumed-mode method may be utilized. The adaptation law can be transformed by the introduction of orthogonal matrix functions, allowing direct use of the equations of motion in principal coordinates for the flexible manipulator.

A computer simulation of the identification problem for a one-link flexible manipulator has been developed, in conjunction with adaptation laws that were determined for the integral and proportional mechanisms. The simulation results show that the adjustable system achieves convergence with the reference model. As the adaptation gains increased, the speed of adaptation also increased. The values of the integral adaptation blocks converged at certain values, while the proportional blocks died out over time. This indicates that the integral adaptation block memorizes the gains, which in turn diminishes the parameter errors between the reference model and the adjustable system. It can be observed from the simulation results that the higher-frequency modes are less effective in the control of adaptive system error than are the lower-frequency modes.

The use of dynamic strain gauges as sensing devices for feedback measurement provides is useful for the control of a flexible manipulator. However, it should be noted that the number of strain gauges fixed on the flexible manipulator must be the same as the number of mode functions utilized in the assumed-mode method for the generation of the equations of motion. This allows for ease of handling of the matrix problem.

Payload variations, presented as the tip mass, have been easily identified by the adaptive identifier based on the model reference adaptive system. Due to the dual characteristics of the model reference adaptive system, the fundamental concept derived from this method can be employed for identification of the reference model based upon a control system with distributed parameters. Further studies of this topic should yield a number of variations.

The extension of concepts discussed in this study to the control of a multi-link flexible manipulator is suggested. It is recommended that the use of the digital control units for a flexible manipulator, based upon the design of a discrete-time model reference adaptive system, be studied further. For the case of flexible manipulators with damping factors that cannot be represented in the form of Rayleigh damping, applications of the model reference adaptive system should be subjected to additional study.

BIBLIOGRAPHY

1. Anderson, B.D.O. and Vongpunitlerd, S., *Network Analysis and Synthesis : A Modern Systems Theory Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1973.

2. Anex, R.P., Jr. and Hubbard, M., "Modeling and Adaptive Control of a Mechanical Manipulator," *ASME Journal of Dynamic Systems, Measurement and control*, Vol. 108, June 1986, pp. 127-135.

3. Book, W.J., "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *The International Journal of Robotics Research*, Vol.3, Fall 1984, pp. 87-101.

4. Book, W.J., Maizza-Neto, O. and Whitney, D.E., "Feedback control of Two Beam, Two Joint Systems with Distributed Flexibility," *ASME Journal of Dynamic Systems, Measurement and control*, Vol. 97, Dec. 1975, pp. 424-431.

5. Book, W.J. and Majette, M., "Controller Design for Flexible, Distributed Parameter Mechanical Arms Via Combined State Space and Frequency Domain Techniques," *ASME Journal of Dynamic Systems, Measurement and control*, Vol. 105, March 1984, pp. 63-69.

6. Cannon, R.H., Jr. and Schmitz, E., "The Initial Experiments on the End-Point Control of a Flexible One-Link Robot," *The International Journal of Robotics Research*, Vol. 3, Fall 1984, pp. 62-75.

7. Chen, Y.H., "Adaptive Robust Model-Following Control and Application to Robot Manipulators," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 109, Sep. 1987, pp. 209-215.

8. Craig, J.J., *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, New York, 1988.

9. Craig, R.R., Jr., *Structural Dynamics: An Introduction to Computer Methods*, John Wiley & Sons, New York, 1981.

10. DeSilva, C.W. and Winssen, J.V., "Least Squares Adaptive Control for Trajectory Following Robots," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 109, June 1987, pp. 105-110.

11. Dubowsky, S. and DesForges, D.T., "The Application of Model-Referenced Adaptive Control to Robotic Manipulators," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 109, Sep. 1987, pp. 193-200.

12. Fletcher, C.A.J., *Computational Galerkin Methods, Springer-Verlag*, New York, 1984.

13. Fukuta, T. and Arakawa, A., "Modeling and Control Characteristic for a Two-Degrees-of-Freedom Coupling System of Flexible Robots Arms," *JSME International Journal*, Vol. 30, No. 267, 1987, pp. 1458-1464.

14. Harris, C.J. and Billings, S.A., *Self-Tuning and Adaptive Control: Theory and Applications*, Peter Peregrinus LTD, London, 1981.

15. Horowitz, R. and Tomizuka, M., " An Adaptive Control Scheme for Mechanical Manipulators-Compensation of Nonlinearity and Decoupling Control," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 108, June 1986, pp. 127-135.

16. Kim, B.K. and Shin, K.G., "An Adaptive Model Following Control of Industrial Manipulators," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-19, No. 6, Nov. 1983, pp. 805-813.

17. Koivo, A.J. and Guo, T., "Adaptive Linear Controller for Robotic Manipulators," *IEEE Transactions on Automatic Control*, Vol. AC-28, Feb. 1983, pp. 162-170.

18. Landau, I.D., "A Survey of Model Reference Adaptive Techniques-Theory and Applications," *Automatica*, Vol. 10, 1974, pp. 353-379.

19. Landau, I.D., *Adaptive Control: The Model Reference Approach*, Marcel Dekker, New York, 1979.

20. Leininger, G.G., "Self-Tuning Adaptive Control of Manipulators, *Proceedings of the International Meeting on Advanced Software in Robotics*, May 4-6, 1983, pp. 81-96.

21. Mees, A.I., *Dynamics of Feedback Systems*, John Wiley & Sons, New York, 1981.

22. Meirovitch, L., *Analytical Methods in Vibrations*, Macmillan, New York, 1967.

23. Meirovitch, L., *Elements of Vibration Analysis*, McGraw-Hill, New York, 1986.

24. Narenda, K.S. and Kudva, P., "Stable Adaptive Schemes for System Identification and Control-Part I," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. ASMC-4, 1974, pp. 542-551.

25. Narenda, K.S. and Kudva, P., "Stable Adaptive Schemes for System Identification and Control-Part II," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. ASMC-4, 1974, pp. 552-560.

26. Oden, J.T. and Reddy, J.N., *An Introduction to the Mathematical Theory of Finite Elements*, John Wiley & Sons, New York, 1976.

27. Ogata, K., *State Space Analysis of Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1967.

28. Padulo, L. and Arbib, M.A., *System Theory*, W. B. Saunders, Philadelphia, 1974.

29. Pestel, E.C. and Leckie, F.A., *Matrix Methods in Elastomechanics*, McGraw-Hill, New York, 1963.

30. Popov, V.M., "The Solution of a New Stability Problem for Controlled Systems," *Autom. Remote Control*, Vol. 24, No. 1, Jan. 1963, pp. 1-23.

31. Popov, V.M., *Hyperstability of Control Systems*, Springer-Verlag, New York, 1970.

32. Reinhardt, H.-J., *Analysis of Approximation Methods for Differential and Integral Equations*, Springer- Verlag, New York, 1985.

33. Safonov, M.G., *Stability and Robustness of Multivariable Feedback Systems*, MIT Press, 1980.

34. Seraji, H., "A New Approach to Adaptive Control of Manipulators," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 109, Sep. 1987, pp. 193-202.

35. Skaar, S.B. and Tucker, D., "Point Control of a One-Link Flexible Manipulator," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 53, March 1986, pp. 23-27.

36. Slotine, J.E. and Li, W., "On the Adaptive Control of Robot Manipulators," *The International Journal of Robotics Research*, Vol. 6, Fall 1987, pp. 49-59.

37. Stepanenko, Y., "Model Control of Fast Large-Scale Robot Motions," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 109, June 1987, pp. 80-87.

38. Sundararajum, N. and Montgomery, R.C., "Experiments Using Lattice Filters to Identify the Dynamics of a Flexible Beam," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 107, Sept. 1985, pp. 187-191.

39. Takegaki, M. and Arimoto, S., "An Trajectory Control of Manipulators," *International Journal of Control*, Vol. 34, No. 2, 1981, pp. 219-230.

40. Temple, G. and Bickley, W. G., *Rayleigh's Principle and Its Applications to Engineering*, Dover, New York, 1956.

41. Tomizuka, M. et al., "Implementation of Adaptive Techniques for Motion Control of Robotic Manipulators," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol 110, March 1988, pp. 62-69.

42. Vaha, P. and Halme, A., "Adaptive Digital Control for a Heavy Manipulators," *Proceedings of the International Meeting on Advanced Software in Robotics*, May 4-6, 1983, pp. 55-61.

43. Weeks, C.J., "Static Shape Determination and Control for Large Space Structures: I. The Flexible Beam," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 106, Dec. 1984, pp. 261-266.

44. Williams, D., *Dynamics Loads in Aeroplanes Under Given Impulsive Loads with Particular Reference to Landing and Gust Loads on a Large Flyingboat*, Great Britain RAE, Reports SME 3309 and 3316, 1945.

45. Zaluckey, A and Hardt, D.E., "Active Control of Robot Structure Deflections," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 106, March 1984, pp. 63-69.

46. Zienkiewicz, O.C. and Morgan, K., *Finite Elements and Approximation*, John Wiley & Sons, New York, 1983.

APPENDICES

APPENDIX A

Mathematical Representation of a Flexible Manipulator by the Assumed-Mode Method.

The use of Lagrange's equations of motion incorporated with the assumed-mode method, with admissible fuctions given by Equation (2.3.16), yields the equations of motion of a flexible manipulator in the form of Equation (4.3.2). Detailed expressions of elements of the [m], [k], and {b} matrices are represented in this appendix. Note that the mass per unit length of the manipulator is assumed as constant for purposes of simplification. The symbol z_i indicates the roots of Equation (2.3.17). The stiffness property, EI, is also considered as constant.

A.1 Elements of [m] matrix: m_{ij}

A.1.1 Case 1), when $i \neq j$,

$$m_{ij} = \int_0^L \rho A \; X_i(x) \; X_j(x) \; dx + m_o \; X_i(L) \; X_j(L) \; , \qquad (A.1.1)$$

or

$$m_{ij} = \rho A \; I_{ij} + m_o \; J_{ij} \; , \qquad (A.1.2)$$

where

$$I_{ij} = \int_0^L X_i(x) \; X_j(x) \; dx$$

$$J_{ij} = m_o \; X_i(L) \; X_j(L) \; . \qquad (A.1.3)$$

By letting

$$c_1 = \{\sinh(z_i)+\sin(z_i)\}\{\sinh(z_j)+\sin(z_j)\} ,$$

$$c_2 = \{\sinh(z_i)+\sin(z_i)\}\{\cosh(z_j)+\cos(z_j)\} ,$$

$$c_3 = \{\cosh(z_i)+\cos(z_i)\}\{\cosh(z_j)+\cos(z_j)\} ,$$

$$c_4 = \{\cosh(z_i)+\cos(z_i)\}\{\sinh(z_j)+\sin(z_j)\} ,$$

and

$$c_5 = 2 / (z_i^2 + z_j^2) , \qquad \text{(A.1.4)}$$

one obtains

$$\begin{aligned} I_{ij} = {} & (Lc_1/2)\{a_{11}-c_5\ a_{12}\} - (Lc_2/2)\{a_{21}-c_5\ a_{22}\} \\ & + (Lc_3/2)\{a_{31}-c_5\ a_{32}\} \\ & - (Lc_4/2)\{a_{41}-c_5\ a_{42}\} \end{aligned} \qquad \text{(A.1.4)}$$

and

$$J_{ij} = c_1\ a_{51} - c_2\ a_{52} + c_3\ a_{53} - c_4\ a_{54} , \qquad \text{(A.1.5)}$$

where

$$a_{11} = \frac{\sin(z_i-z_j)+\sinh(z_i-z_j)}{z_i-z_j} + \frac{\sin(z_i+z_j)+\sinh(z_i+z_j)}{z_i+z_j} , \qquad \text{(A.1.6)}$$

$$\begin{aligned} a_{12} = {} & z_i\{\cosh(z_j)\sin(z_i)+\sinh(z_i)\cos(z_j)\} \\ & + z_j\{\sinh(z_j)\cos(z_i) \\ & + \cosh(z_i)\sin(z_j)\} , \end{aligned} \qquad \text{(A.1.7)}$$

$$a_{21} = \frac{\cosh(z_j-z_i)-\cos(z_j-z_i)}{z_j-z_i} + \frac{\cosh(z_i+z_j)-\cos(z_i+z_j)}{z_i+z_j} , \qquad \text{(A.1.8)}$$

$$a_{22} = z_i\{\sinh(z_j)\sin(z_i)+\sinh(z_i)\sin(z_j)\}$$
$$+ z_j\{\cosh(z_j)\cos(z_i)$$
$$- \cosh(z_i)\cos(z_j)\} , \qquad (A.1.9)$$

$$a_{31} = \frac{\sin(z_i-z_j)-\sinh(z_i-z_j)}{z_i-z_j}$$
$$- \frac{\sin(z_i+z_j)-\sinh(z_i+z_j)}{z_i+z_j} , \qquad (A.1.10)$$

$$a_{32} = z_i\{\cosh(z_i)\sin(z_j)-\sinh(z_j)\cos(z_i)\}$$
$$+ z_j\{\cosh(z_j)\sin(z_i)$$
$$- \sinh(z_i)\cos(z_j)\} , \qquad (A.1.11)$$

$$a_{41} = \frac{\cosh(z_i-z_j)-\cos(z_i-z_j)}{z_i-z_j}$$
$$- \frac{\cosh(z_i+z_j)-\cos(z_i+z_j)}{z_i+z_j} , \qquad (A.1.12)$$

$$a_{42} = z_i\{\cosh(z_i)\cos(z_j)-\cosh(z_j)\cos(z_i)\}$$
$$+ z_j\{\sinh(z_i)\sin(z_j)$$
$$+ \sinh(z_j)\sin(z_i)\} , \qquad (A.1.13)$$

$$a_{51} = \cos(z_i)\{\cos(z_j)-\cosh(z_j)\}$$
$$- \cosh(z_i)\{\cos(z_j)-\cosh(z_j)\} , \qquad (A.1.14)$$

$$a_{52} = \cos(z_i)\{\sin(z_j)-\sinh(z_j)\}$$
$$- \cosh(z_i)\{\sin(z_j)-\sinh(z_j)\} , \qquad (A.1.15)$$

$$a_{53} = \sin(z_i)\{\sin(z_j)-\sinh(z_j)\}$$
$$- \sinh(z_i)\{\sin(z_j)-\sinh(z_j)\} , \qquad (A.1.16)$$

and

$$a_{54} = \sin(z_i)\{\cos(z_j)-\cosh(z_j)\}$$
$$- \sinh(z_i)\{\cos(z_j)-\cosh(z_j)\} . \qquad (A.1.17)$$

A.1.2 Case 2, when i = j

$$m_{ii} = \int_o^L \rho A \; X_i^2(x) \; dx + m_o \; X_i^2(L)$$

or

$$m_{ij} = \rho A \; I_{ii} + m_o \; J_{ii} \qquad (A.1.18)$$

where

$$I_{ii} = \int_o^L X_i^2(x) \; dx$$

$$J_{ii} = m_o \; X_i^2(L) \; . \qquad (A.1.19)$$

By letting

$$d_1 = \{\sinh(z_i) + \sin(z_i)\}^2 \; ,$$

$$d_2 = \{\sinh(z_i) + \sin(z_i)\}\{\cosh(z_i) + \cos(z_i)\} \; ,$$

and

$$d_3 = \{\cosh(z_i) + \cos(z_i)\}^2 \qquad (A.1.20)$$

one obtains

$$I_{ii} = \{Ld_1/(4z_i)\} \; a_{61}\} - (Ld_2/z_i) \; a_{62}\}$$
$$- \{Ld_3/(4z_i)\} \; a_{63}\} \qquad (A.1.21)$$

and

$$J_{ii} = d_1 \; a_{71} - 2 \; d_2 \; a_{72} + d_3 \; a_{73} \; , \qquad (A.1.22)$$

where

$$a_{61} = 4z_i + \sin(2z_i) - 4\sinh(z_i)\cos(z_i)$$
$$- 4\cosh(z_i)\sin(z_i)$$
$$+ 2\sinh(z_i)\cosh(z_i) \; , \qquad (A.1.23)$$

$$a_{62} = \{\sin(z_i) - \sinh(z_i)\}^2 \; , \qquad (A.1.24)$$

and

$$a_{63} = -\sin(2z_i) - 4\cosh(z_i)\sin(z_i) + 4\sinh(z_i)\cos(z_i) + 2\sinh(z_i)\cosh(z_i) \; . \qquad \text{(A.1.25)}$$

A.2 The Elements of [k] Matrix: k_{ij}

A.2.1 Case 1, when $i \neq j$

$$k_{ij} = \int_o^L EI\ X_i''(x)\ X_j''(x)dx \qquad \text{(A.2.1)}$$

or

$$k_{ij} = EI\ L_{ij} \qquad \text{(A.2.2)}$$

where

$$L_{ij} = \int_o^L X_i''(x)\ X_j''(x)\ dx \; . \qquad \text{(A2.3)}$$

Letting

$$z_i' = (z_i/L)^2 \; ,$$
$$z_j' = (z_j/L)^2 \; ,$$
$$c_1' = z_i'z_j'\ c_1 \; ,$$
$$c_2' = z_i'z_j'\ c_2 \; ,$$
$$c_3' = z_i'z_j'\ c_3 \; ,$$
$$c_4' = z_i'z_j'\ c_4 \; ,$$

and

$$c_6 = L\ /\ (z_i^2+z_j^2) \; , \qquad \text{(A.2.4.)}$$

one obtains

$$L_{ij} = c_1'\{(L/2)a_{11}+c_6\ a_{12}\} - c_2'\{(L/2)a_{21}+c_6\ a_{22}\} + c_3'\{(L/2)a_{31}+c_6a_{32}\} - c_4'\{(L/2)a_{41}+c_6a_{42}\} \qquad \text{(A.2.5)}$$

where c_i and a_{ij} are given by Equations (A.1.3) and (A.1.7) to (A.1.13).

A.2.2 Case 2, when i = j

$$k_{ij} = \int_0^L EI \{X_i''(x)\}^2 \, dx$$

or

$$k_{ij} = EI \, L_{ii} \qquad (A.2.6)$$

where

$$L_{ii} = \int_0^L \{X_i''(x)\}^2 \, dx \, . \qquad (A.2.7)$$

Letting

$$d_1' = (z_i/4)^4 \, d_1 \, ,$$

$$d_2' = 2 \, (z_i/4)^4 \, d_2 \, ,$$

$$d_3' = (z_i/4)^4 \, d_3 \, ,$$

and

$$d_4 = L/(4z_i) \, , \qquad (A.2.8)$$

one obtains

$$L_{ii} = (Ld_1'/d_4)a_{81} + 2d_2'a_{82} + d_3'a_{83} \, , \qquad (A.2.9)$$

where

$$a_{81} = 4z_i + \sin(2z_i) + 2\sinh(z_i)\cosh(z_i) + 4\sinh(z_i)\cos(z_i) + 4\cosh(z_i)\sin(z_i)\} \, , \qquad (A.2.10)$$

$$a_{82} = \{\sin(z_i) + \sinh(z_i)\}^2 \, , \qquad (A.2.11)$$

and

$$a_{83} = 2\sinh(z_i)\cosh(z_i) - \sin(2z_i) + 4\cosh(z_i)\sin(z_i) - 4\sinh(z_i)\cos(z_i) \, . \qquad (A.2.12)$$

A.3 Elements of {b} Matrix: b_i

$$b_i = \int_o^L \rho A \; x \; X_i(x) \; dx + m_o \; L \; X_i(L)$$

or

$$b_i = \rho A \; B_i + m_o \; L \; X_i \qquad (A.3.1)$$

where

$$B_i = \int_o^L x \; X_i(x) \; dx \; . \qquad (A.3.2)$$

Or

$$b_i = \rho A \; (L/z_i)^2 \; (a_{91}-a_{92}) + m_o \; L \; X_i(L) \qquad (A.3.3)$$

where

$$a_{91} = \{\sinh(z_i) + \sin(z_i)\} \; [\{\cosh(z_i) + \cos(z_i)\} - z_i\{\sinh(z_i)-\sin(z_i)\} - 2] \qquad (A.3.4)$$

and

$$a_{92} = \{\cosh(z_i) + \cos(z_i)\} \; [\{\sinh(z_i) + \sin(z_i)\} - z_i\{\cosh(z_i)+\cos(z_i)\}] \; . \qquad (A.3.5)$$

APPENDIX B

Computer Simulation Program *FLEX*

B.1 Algorithm

The fundamental algorithm of the simulation program *FLEX* is as follows:

1) Define the reference model and initial state of the adjustable system;
2) Calculate the generalized states of the reference model and the adjustable system and obtain the error vector;
3) Get filtered error vector;
4) Obtain the adaptive gains of the adaptation mechanism by the use of the error vector filtered through the compensator;
5) Adjust the adjustable system and define new parameters of the adjustable system; and
6) Repeat steps 2 through 5 of the above procedure.

B.2 Simulation Program *FLEX*

```
      PROGRAM FLEX
C *****************************************************************
C
C   Simulation Program for Control of a Flexible Manipulator
C      * Mathematical Model by Assumed-Mode Method
C      * Control Scheme by Model Reference Adaptive System
C      * Real Variables in Double-Precision
C
C *****************************************************************

      INTEGER RSIZE,PSIZE,ICHK,IGND,IPMT,IPVE,ICON,IPRT,ITF,INTL
      REAL*8 RMAS,PMAS,RMASS(8,8),PMASS(8,8),RSTIF(8,8)
      REAL*8 PSTIF(8,8),REVAL(8),PEVAL(8),REVEC(8,8),PEVEC(8,8)
      REAL*8 AX3(3,3),W(6),WK(6),ZW(18),RMIJ,RKIJ,REIG,RCONS(15)
      REAL*8 DX,DT,RT,RX,AX2(15,15),T,X,RTEVC(8,8),PTEVC(8,8)
      REAL*8 RSCR1(8,8),RSCR2(8,8),RSCR3(8,8),RSCR4(8,8)
      REAL*8 PSCR1(8,8),PSCR2(8,8),PSCR3(8,8),PSCR4(8,8)
      REAL*8 RMR(8),RKR(8),RBM(8),RSTA(8),RDSTA(8),ERVEC(8)
      REAL*8 PMR(8),PKR(8),PBM(8),PSTA(8),PDSTA(8),EDVEC(8)
      REAL*8 ERV2(8),PH1(8,8),PH2(8,8),PI1(8),PI2(8),PMINK(8,8)
      REAL*8 PHINT(8,8),THT,APNEW(8,8),BPNEW(8),PSTIN(8,8)
      REAL*8 PMNEW(8,8),PBI(8),RDFL,PDFL,ACN(4),PFEVC(8,8)
      REAL*8 PFEVL(8),PFMR(8),PFBM(8),PFSTA(8),PFDFL,RPER,RPFER
      REAL*8 RTIVC(8,8),RMRT(8),PLRT(8),PFRT(8),PH,ERV1(8)
      REAL*8 SCR,ADEVL(8),ADRM,ADRT(8),ADMAS(8),ADSTIF(8,8)
      REAL*8 RFTH,PLTH,PFTH,PERTH,FERTH,GERTH,PFDSTA(8)
      REAL*8 ZETA,ALPHA,RMLEN,PIINT(8)

C
C   Calculation of Roots of Characteristic Equations.
C

      PH = 3.14159265359
      ZETA = 0.05
      ALPHA = 2.5
      RMLEN = 2000.
      WRITE(*,*) ' Enter The Tip Mass of Ref. Model.'
      READ(*,*)  RMAS
      WRITE(*,*) ' Eneter The Tip Mass of Int. Model.'
      READ(*,*)  PMAS
      WRITE(*,*) ' ** RMAS & PMAS **'
      WRITE(*,*) RMAS, PMAS
      CALL ROOT(RMAS,RMRT)
      CALL ROOT(PMAS,PLRT)
      WRITE(*,*) '   *** RMRT ***'
      WRITE(*,*) (RMRT(I),I=1,8)
      WRITE(*,*) '   *** PLRT ***'
      WRITE(*,*) (PLRT(I),I=1,8)
```

```
C
C  Calculation of the Elements of [m] and [k] matrices.
C

      WRITE(*,*) '  Please Enter Adaptation Constant ACN(4)!'
      DO 1Ø ICON=1,4
        READ(*,*) ACN(ICON)
   1Ø CONTINUE
      WRITE(*,*) (ACN(ICON),ICON=1,4)
      RSIZE = 3
      PSIZE = 3
      CALL MK(RMAS,RMRT,RMASS,RSTIF,RSIZE)
      CALL MK(PMAS,PLRT,PMASS,PSTIF,PSIZE)
      ICHK = Ø
        WRITE(*,*) '  '
        WRITE(*,*) '  **  RMASS & PMASS  ** '
          CALL PRNTM(RMASS,RSIZE)
          CALL PRNTM(PMASS,RSIZE)
        WRITE(*,*) '  **  RSTIF & PSTIF  ** '
          CALL PRNTM(RSTIF,RSIZE)
          CALL PRNTM(PSTIF,RSIZE)
      CALL EIGMT(RSIZE,RMASS,RSTIF,ICHK,REVAL,REVEC)
        WRITE(*,*) '  **  REVEC & REVAL **'
          CALL PRNTM(REVEC,RSIZE)
          CALL PRNTV(REVAL,RSIZE)

      ICHK = 1
      CALL EIGMT(PSIZE,PMASS,PSTIF,ICHK,PEVAL,PEVEC)
        WRITE(*,*) '  **  PEVEC & PEVAL **'
          CALL PRNTM(PEVEC,PSIZE)
          CALL PRNTV(PEVAL,PSIZE)

      CALL CONS(RMRT,REVEC,REVAL,RMASS,RCONS,RSIZE)
      CALL CONS(PLRT,PEVEC,PEVAL,PMASS,PCONS,PSIZE)

C
C   Initilization fo Time Loop
C       DT : Time Interval
C

      DO 2Ø I = 1,PSIZE
        DO 3Ø J = 1,PSIZE
          PH1(I,J) = Ø.
          PMNEW(I,J) = PMASS(I,J)
          PFEVC(I,J) = PEVEC(I,J)
   3Ø   CONTINUE
        PI1(I)   = Ø.
        PFEVL(I) = PEVAL(I)
        PFRT(I)  = PLRT(I)
   2Ø CONTINUE
      CALL BIMT(RSIZE,RMAS,RMRT,RBM)
```

```
      CALL BIMT(PSIZE,PMAS,PLRT,PBM)
      DO 40 I = 1,PSIZE
        PSCR3(I,1) = PBM(I)
        PFBM(I)    = PBM(I)
   40 CONTINUE
      CALL INVERS(PMASS,PSCR4,PSIZE)
      CALL MULTI(PSCR4,PSTIF,PMINK,PSIZE,PSIZE,PSIZE)
      WRITE(*,*) '  ** PMINK ** '
      CALL PRNTM(PMINK,PSIZE)
      I = 1
      CALL MULTI(PSCR4,PSCR3,PSCR2,PSIZE,PSIZE,I)
      DO 50 J = 1,PSIZE
        PBI(J) = PSCR2(J,1)
   50 CONTINUE

      WRITE(*,*) '  **  RMASS & RSTIF  ** '
      CALL PRNTM(RMASS,RSIZE)
      CALL PRNTM(RSTIF,RSIZE)
      WRITE(*,*) '  **  PMASS & PSTIF  **'
      CALL PRNTM(PMASS,PSIZE)
      CALL PRNTM(PSTIF,PSIZE)
      WRITE(*,*) '  **  REVEC & REVAL **'
      CALL PRNTM(REVEC,RSIZE)
      CALL PRNTV(REVAL,RSIZE)
      WRITE(*,*) '  **  PEVEC & PEVAL **'
      CALL PRNTM(PEVEC,PSIZE)
      CALL PRNTV(PEVAL,PSIZE)
      WRITE(*,*) '  **  RBM & PBM  **'
      CALL PRNTV(RBM,RSIZE)
      CALL PRNTV(PBM,PSIZE)
      WRITE(*,*) '  **  PBI  **'
      CALL PRNTV(PBI,RSIZE)
C
C  Calculation of [m] & [k] matrices of Reference Model
C
      CALL TRANPS(REVEC,RTEVC,RSIZE,RSIZE)
      CALL MULTI(RTEVC,RMASS,RSCR1,RSIZE,RSIZE,RSIZE)
      CALL MULTI(RSCR1,REVEC,RSCR2,RSIZE,RSIZE,RSIZE)
      CALL TRANPS(PEVEC,PTEVC,PSIZE,PSIZE)
      CALL MULTI(PTEVC,PMASS,PSCR1,PSIZE,PSIZE,PSIZE)
      CALL MULTI(PSCR1,PEVEC,PSCR2,PSIZE,PSIZE,PSIZE)
      DO 60 I=1,RSIZE
        RMR(I)  = RSCR2(I,I)
        PFMR(I) = PSCR2(I,I)
   60 CONTINUE
      WRITE(*,*) ' **  RMR  **'
      CALL PRNTV(RMR,RSIZE)
      CALL INVERS(REVEC,RSCR1,RSIZE)
      CALL INVERS(RTEVC,RTIVC,RSIZE)
C
C  Configuration of Data Storage Files.
```

```
C
      OPEN(UNIT=1,FILE='MR1.DAT',STATUS='OLD')
      OPEN(UNIT=2,FILE='MR2.DAT',STATUS='OLD')
      OPEN(UNIT=3,FILE='MR3.DAT',STATUS='OLD')

      OPEN(UNIT=4,FILE='BM1.DAT',STATUS='OLD')
      OPEN(UNIT=5,FILE='BM2.DAT',STATUS='OLD')
      OPEN(UNIT=6,FILE='BM3.DAT',STATUS='OLD')

      OPEN(UNIT=7,FILE='DFL.DAT',STATUS='OLD')
      OPEN(UNIT=8,FILE='MAS.DAT',STATUS='OLD')
      OPEN(UNIT=9,FILE='DER.DAT',STATUS='OLD')

      OPEN(UNIT=1Ø,FILE='PI1.DAT',STATUS='OLD')
      OPEN(UNIT=11,FILE='PI2.DAT',STATUS='OLD')
      OPEN(UNIT=12,FILE='PI3.DAT',STATUS='OLD')

      OPEN(UNIT=13,FILE='TH1.DAT',STATUS='OLD')
      OPEN(UNIT=14,FILE='TH3.DAT',STATUS='OLD')

      OPEN(UNIT=15,FILE='EV1.DAT',STATUS='OLD')
      OPEN(UNIT=16,FILE='EV2.DAT',STATUS='OLD')

      T  = Ø.
      WRITE(*,*) ' ENTER DT.'
      READ(*,*) DT
      WRITE(*,*) ' ENTER FINAL INTERATION TIME, ITF.'
      WRITE(*,*) ' ITF = (Final Time) / DT.'
      READ(*,*) ITF
      WRITE(*,*) ' ENTER INTERVAL OF DATA SAVING, INTL.'
      READ(*,*) INTL

C             CCCCCCCCCCCCCCCCCCCCCC
C             CCCCC  Time Loop  CCCC
C             CCCCCCCCCCCCCCCCCCCCCC

      DO 3ØØ IGND = 1,ITF
        IPRT = IPRT + 1
        T = T + DT
        IF (IPRT.NE.INTL) GOTO 7Ø
        WRITE(*,*) '  ******************************************'
        WRITE(*,*) '     **  AT TIME =',T,' ** '
        WRITE(*,*) '   '
C
C  Calculation of [m] and [k] Matrices of Adjustable System
C
   7Ø   CALL TRANPS(PEVEC,PTEVC,PSIZE,PSIZE)
        CALL MULTI(PTEVC,PMNEW,PSCR1,PSIZE,PSIZE,PSIZE)
        CALL MULTI(PSCR1,PEVEC,PSCR2,PSIZE,PSIZE,PSIZE)
        CALL MULTI(PTEVC,PSTIF,PSCR3,PSIZE,PSIZE,PSIZE)
        CALL MULTI(PSCR3,PEVEC,PSCR4,PSIZE,PSIZE,PSIZE)
        CALL INVERS(PEVEC,PSCR1,PSIZE)
```

```
      DO 80 I=1,RSIZE
        PMR(I) = PSCR2(I,I)
        PKR(I) = PSCR4(I,I)
 80   CONTINUE

      DO 90 IV = 1,3
        WRITE(IV,*) T, RMR(IV),PMR(IV)
 90   CONTINUE

      WRITE(*,*) '  **  PMR **'
      CALL PRNTV(PMR,PSIZE)
      WRITE(*,*) '  ** RKR & PKR **'
      CALL PRNTV(RKR,RSIZE)
      CALL PRNTV(PKR,PSIZE)
      WRITE(*,*) '  ** PEVAL **'
      CALL PRNTV(PEVAL,PSIZE)

C
C Calculation of Error Vector
C

      CALL DST(RSIZE,RBM,RMR,RSTA,RDSTA,REVEC,REVAL,T)
      CALL DST(PSIZE,PBM,PMR,PSTA,PDSTA,PEVEC,PEVAL,T)
      CALL DST(PSIZE,PFBM,PFMR,PFSTA,PFDSTA,PFEVC,PFEVL,T)
      WRITE(*,*) '  **  RSTA & RDSTA  **'
        CALL PRNTV(RSTA,RSIZE)
        CALL PRNTV(RDSTA,RSIZE)
      WRITE(*,*) '  **  PSTA & PDSTA  **'
        CALL PRNTV(PSTA,PSIZE)
        CALL PRNTV(PDSTA,PSIZE)

C
C   Calculation of Deflections at Tip
C
      IF (IPRT.NE.INTL) GOTO 100
      CALL FCDFL(RSIZE,RMRT,RDFL,RMLEN,RSTA)
      CALL FCDFL(PSIZE,PLRT,PDFL,RMLEN,PSTA)
      CALL FCDFL(PSIZE,PFRT,PFDFL,RMLEN,PFSTA)
      WRITE(*,*) ' ++++ RDFL, PDFL & PFDFL ++++'
      WRITE(*,*) RDFL,PDFL,PFDFL
      RPER  = RDFL - PDFL
      RPFER = RDFL - PFDFL
      WRITE(*,*) '                ',RPER,RPFER
      WRITE(7,*) RDFL,PDFL,PFDFL
      WRITE(9,*) T,RPER,RPFER

 100  DO 110 IER = 1,RSIZE
         RSCR2(IER,1) = RSTA(IER)
         RSCR3(IER,1) = RDSTA(IER)
         PSCR2(IER,1) = PSTA(IER)
         PSCR3(IER,1) = PDSTA(IER)
 110  CONTINUE
```

```
C
C    ETA-ERROR  = PSCR3
C    ETA-ERROR' = PSCR4
C    PETA-STATE = RSCR3
C
        IET = 1

        CALL MULTI(RSCR1,RSCR3,RSCR4,RSIZE,RSIZE,IET)
        CALL MULTI(RSCR1,RSCR2,RSCR3,RSIZE,RSIZE,IET)
        CALL MULTI(PSCR1,PSCR3,PSCR4,PSIZE,PSIZE,IET)
        CALL MULTI(PSCR1,PSCR2,PSCR3,PSIZE,PSIZE,IET)

        WRITE(*,*) '  **  ERVEC & EDVEC **'
          CALL PRNTV(ERVEC,PSIZE)
          CALL PRNTV(EDVEC,PSIZE)
C
C  V = [D]{e},  {V1} = [M:-1]*{e1} - {e2}
C               {V2} = - {e1} - {e2}
C  Del[A] = Int[Ph1] + [Ph2]
C               [Ph1] = [Fa2]{V2}{[Ga1]{Qp1}:Tranps}
C               [Ph2] = [Fa2']{V2}{[Ga1']{Qp1}:Tranps}
C        ANEW = Del[A] + [Mp:-1]*[Kp]
C  Del{B} = Int[Pi1] + [Pi2]
C               [Pi1] = [Fb2]{V2}Gb*THETA"(t)
C               [Pi2] = [Fb2']{V2}Gb'*THETA"(t)
C        BNEW = Del{B} + [Mp:-1]*{Bp}
C

        DO 120 IV = 1,RSIZE
          ERV1(IV) = (2*ZETA + ALPHA/(2*ZETA)) * REVAL(IV)
     &                  * (RSCR2(IV,1)-PSCR2(IV,1))
     &                  + RSCR3(IV,1) - PSCR3(IV,1)
          ERV2(IV) = RSCR2(IV,1) - PSCR2(IV,1)
     &                  + (ALPHA/(2*ZETA*REVAL(IV)))
     &                  * (RSCR3(IV,1) - PSCR3(IV,1))
  120   CONTINUE

        WRITE(*,*) '  **  ERV2  **'
        CALL PRNTV(ERV2,PSIZE)

        WRITE(15,*) ERV1(1),ERV1(2),ERV1(3)
        WRITE(16,*) ERV2(1),ERV2(2),ERV2(3)

C
C  Qp1 = PSTA
C    IF Fa2 = Ga1 = I, Fa2' = Ga1' = I
C
C  PH1 = {V2}*{Qp1:Tranps}
C     {V2} = ERV2
C
        CALL THETA(T,THT)
```

```
      RFTH = THT + (RDFL/RMLEN)
      PLTH = THT + (PDFL/RMLEN)
      PFTH = THT + (PFDFL/RMLEN)
      PERTH = RFTH - PLTH
      FERTH = RFTH - PFTH
      GERTH = RFTH - THT

      WRITE(13,*) T,THT
      WRITE(14,*) PERTH,FERTH,GERTH

      DO 13Ø IV = 1,PSIZE

        PHINT(IV,IV) = ACN(1) * ERV2(IV)
     &                        * PSCR2(IV,1)
          WRITE(*,*) '**  PHINT(',IV,JV,')=',PHINT(IV,JV)
        PH1(IV,IV) = PH1(IV,IV) + DT*PHINT(IV,IV)
          WRITE(*,*) '**  PH1(',IV,JV,')=',PH1(IV,JV)
        PH2(IV,IV) = ACN(2) * ERV2(IV) * PSCR2(IV,1)
          WRITE(*,*) '**  PH2(',IV,JV,')=',PH2(IV,JV)
        DO 14Ø JV = 1,PSIZE
          IF (IV.EQ.JV) THEN
            RSCR4(IV,JV) = PHINT(IV,JV)*DT + PH2(IV,JV)
          ELSE
            RSCR4(IV,JV) = Ø.
          ENDIF
14Ø     CONTINUE
           WRITE(*,*) 'ERV2(IV) =',ERV2(IV)
        PIINT(IV) = ACN(3) * ERV2(IV) * THT * DT
        PI1(IV) = PIINT(IV) + PI1(IV)
          WRITE(*,*) '** PI1(',IV,')=',PI1(IV)
        PI2(IV) = ACN(4)*ERV2(IV) * THT
          WRITE(*,*) '** PI2(',IV,')=',PI2(IV)
        PSCR3(IV,1) = PI1(IV) + PI2(IV)
13Ø   CONTINUE

         DO 15Ø IV = 1Ø,12
           JV = IV - 9
           WRITE(IV,*) T,PI1(JV),PI2(JV)
15Ø      CONTINUE

      DO 16Ø IV = 1,PSIZE
        PBI(IV) = PFBM(IV) + PSCR3(IV,1)
16Ø   CONTINUE

      CALL FINDM(PSIZE,PLRT,PBI,ADRM)

      CALL ROOT(ADRM,PLRT)
         IF (IPRT.NE.INTL) GOTO 17Ø
         WRITE(8,*) RMAS, ADRM
17Ø   CALL MK(ADRM,PLRT,PMNEW,PSTIF,PSIZE)
      CALL BIMT(PSIZE,ADRM,PLRT,PBM)
```

```
      DO 180 IV = 1,PSIZE
        PI1(IV) = PBM(IV) - PFBM(IV)
180   CONTINUE

         DO 190 IV = 4,6
           JV = IV - 3
           WRITE(IV,*) T,RBM(JV),PBM(JV)
190      CONTINUE

      IF (IPRT.NE.INTL) GOTO 200
      WRITE(*,*) '   *** ADRM =',ADRM, ' ***'

      WRITE(*,*) ' *=*=*  PMNEW *=*=*'
        CALL PRNTM(PMNEW,PSIZE)
      WRITE(*,*) '   ** RBM **'
        CALL PRNTV(RBM,RSIZE)
      WRITE(*,*) '   *=*=*  PBM *=*=*'
        CALL PRNTV(PBM,PSIZE)
      IPRT = 0
200   ICHK = 1
      CALL EIGMT(PSIZE,PMNEW,PSTIF,ICHK,PEVAL,PEVEC)
      WRITE(*,*) ' ***  PEVEC & PEVAL ***'
        CALL PRNTM(PEVEC,PSIZE)
        CALL PRNTV(PEVAL,PSIZE)
300 CONTINUE

      ENDFILE 1
      CLOSE(1)
      ENDFILE 2
      CLOSE(2)
      ENDFILE 3
      CLOSE(3)
     ENDFILE 4
     CLOSE(4)
     ENDFILE 5
     CLOSE(5)
     ENDFILE 6
     CLOSE(6)
      ENDFILE 7
      CLOSE(7)
      ENDFILE 8
      CLOSE(8)
      ENDFILE 9
      CLOSE(9)
     ENDFILE 10
     CLOSE(10)
     ENDFILE 11
     CLOSE(11)
     ENDFILE 12
     CLOSE(12)
      ENDFILE 13
      CLOSE(13)
```

```
       ENDFILE 14
       CLOSE(14)
      ENDFILE 15
      CLOSE(15)
      ENDFILE 16
      CLOSE(16)

      STOP
      END



         SUBROUTINE MK(RMK,MKRT,TMASS,TSTIF,IDIM)
C
C   Calculation of Components of [m] & [k] Matrices
C     Input : RMK(mass), MKRT, IDIM(dim)
C     Output: TMASS = [m], TSTIF = [k]
C
      INTEGER IDIM
      REAL*8 MKRT(8),TMASS(8,8), TSTIF(8,8),RMM,RKK,RMK

      DO 5ØØ II=1,IDIM
        DO 51Ø JJ=1,IDIM
          IF (II.EQ.JJ) THEN
            CALL MII(RMK,MKRT,RMM,II)
            CALL KII(MKRT,RKK,II)
          ELSE
            CALL MIJ(RMK,MKRT,RMM,II,JJ)
            CALL KIJ(MKRT,RKK,II,JJ)
          ENDIF
          TMASS(II,JJ) = RMM
          TSTIF(II,JJ) = RKK
 51Ø    CONTINUE
 5ØØ  CONTINUE
      RETURN
      END



         SUBROUTINE EIGMT(IESZ,EMASS,ESTIF,ICH,EIGVAL,EIGVEC)
C
C   Calculation of Eigen Values and Vectors.
C     Input : IESZ(dim),EMASS,ESTIF,ICH
C     Output: EIGVEC,EIGVAL
C
      INTEGER IESZ,IA,IZ,N,IJOB,ICH,IM,IER,ISW
      REAL*8 EMASS(8,8),ESTIF(8,8),AX1(8,8),AX2(8,8)
      REAL*8 AXR(3,3),WR(6),WKR(6),ZWR(18)
      REAL*8 AXP(3,3),WP(6),WKP(6),ZWP(18)
      REAL*8 EIGVAL(8),EIGVEC(8,8),REIG,OCHK,SML
```

```
      IA    =  IESZ
      IZ    =  IESZ
      N     =  IESZ
      IJOB =  1
          WRITE(*,*) ' EMASS =',((EMASS(IJI,JIJ),IJI=1,IESZ),
     &                                  JIJ=1,IESZ)
      CALL INVERS(EMASS,AX1,IESZ)
      CALL MULTI(AX1,ESTIF,AX2,IESZ,IESZ,IESZ)
C
C  Call Library (IMSL: "EIGRF")
C
      DO 53Ø I = 1,IESZ
        DO 52Ø J = 1,IESZ
          IF (ICH.EQ.Ø) THEN
            AXR(I,J) = AX2(I,J)
          ELSE
            AXP(I,J) = AX2(I,J)
          ENDIF
  52Ø   CONTINUE
  53Ø CONTINUE

      IF (ICH.EQ.Ø) THEN
        CALL EIGRF(AXR,N,IA,IJOB,WR,ZWR,IZ,WKR,IER)
      ELSE
        CALL EIGRF(AXP,N,IA,IJOB,WP,ZWP,IZ,WKP,IER)
      ENDIF

      IM  =  1
      DO 54Ø IK = 1,IESZ
        IJ = IK*2
        II = IJ-1
        IF(ICH.EQ.Ø) THEN
          REIG = DABS(WR(II))
        ELSE
          REIG = DABS(WP(II))
        ENDIF
        EIGVAL(IM) = DSQRT(REIG)
        IM = IM + 1
  54Ø CONTINUE

      IMP = Ø
      DO 56Ø IL = 1,IESZ
        DO 55Ø IM = 1,IESZ
          IJ = 2*(IM+IMP)
          II = IJ-1
          IF (ICH.EQ.Ø) THEN
            EIGVEC(IM,IL) = ZWR(II)
          ELSE
            EIGVEC(IM,IL) = ZWP(II)
          ENDIF
  55Ø   CONTINUE
        IMP = IMP + IESZ
```

```
  560 CONTINUE

      IM = IESZ - 1
      DO 570 II = 1,IM
        OCHK = EIGVAL(II)
        ISW = II
        DO 580 IJ = II+1,IESZ
          SML = EIGVAL(IJ)
          IF (OCHK.LE.SML) GOTO 580
            ISW = IJ
            OCHK = EIGVAL(ISW)
  580   CONTINUE
        IF(ISW.EQ.II) GOTO 570
        EIGVAL(ISW) = EIGVAL(II)
        EIGVAL(II)  = OCHK
        DO 590 IJ = 1, IESZ
          OCHK = EIGVEC(IJ,II)
          EIGVEC(IJ,II)  = EIGVEC(IJ,ISW)
          EIGVEC(IJ,ISW) = OCHK
  590   CONTINUE
  570 CONTINUE

      RETURN
      END



         SUBROUTINE BIMT(IBDM,RMBI,BIRT,BMT)
C
C  Calculation of [b] Matrix.
C
      INTEGER  IBDM
      REAL*8 BMT(8),BIRT(8),SH,SN,CH,CN,A11,A12,A21,A22,A1
      REAL*8 RLEN, RHOA,AA,AB,RMBI,ZI

      RLEN = 2000.
      RHOA = (80.0D-3) / RLEN

      DO 595 IB = 1,IBDM
        ZI = BIRT(IB)
         SH = DSINH(ZI)
         SN = DSIN(ZI)
        CH = DCOSH(ZI)
        CN = DCOS(ZI)
         A1 = RHOA * (RLEN/ZI) * (RLEN/ZI)
        A11 = SH + SN
        A12 = CH + CN - ZI * (SH-SN) - 2.
        A21 = CH + CN
        A22 = SH + SN - ZI * (CH+CN)
        AA  = A1 * (A11*A12 - A21*A22)
        AB  = RMBI * RLEN * (A11*(CN-SH) - A21*(SN-SH))
        BMT(IB) = AA + AB
  595 CONTINUE
```

```
      RETURN
      END


      SUBROUTINE FINDM(FNSIZE,FNRT,FNBI,FNRM)
C
C  Caculation of Parameters Adjusted.
C    Input : FNSIZE(DIMENSION),FNRT(Zi)
C    Output: FNRM(MASS)
C
      INTEGER  FNSIZE
      REAL*8 FNRT(8),FSH,FSN,FCH,FCN,B11,B12,B21,B22,B1
      REAL*8 FLEN, RHOA,FAA,FAB,FNRM,FZI,FNBI(8),FNWRM

      FLEN = 2ØØØ.
      RHOA = (8Ø.ØD-3) / FLEN
      FNRM = Ø.
      DO 6ØØ IB = 1,FNSIZE
        FZI = FNRT(IB)
         FSH = DSINH(FZI)
         FSN = DSIN(FZI)
        FCH = DCOSH(FZI)
        FCN = DCOS(FZI)
         B1 = RHOA * (FLEN/FZI) * (FLEN/FZI)
        B11 = FSH + FSN
        B12 = FCH + FCN - FZI * (FSH-FSN) - 2.
        B21 = FCH + FCN
        B22 = FSH + FSN - FZI * (FCH+FCN)
        FAA = B1 * (B11*B12 - B21*B22)
        FAB = FLEN * (B11*(FCN-FSH) - B21*(FSN-FSH))
        FNWRM =  ((FNBI(IB)-FAA) / FAB)
        FNRM = FNWRM + FNRM
  6ØØ CONTINUE

      FNRM = FNRM / FNSIZE
      RETURN
      END


      SUBROUTINE DST(DSIZE,DBI,DMI,DSTA,DDSTA,DVEC,DVAL,TD)
C
C   Calculation of Matural Coordinates, DSTA & DDSTA with
C      Rayleigh Damping Factors.
C
C      Input : DSIZE(Dim), DBI(Bi Matrix), DMI(Mi Matrix),
C              DVEC(Modal Matrix), DVAL(Eigenvalues)
C      Output: DSTA(Position), DDSTA(Velocity)
```

```
C
      INTEGER DSIZE,ID
      REAL*8  TD,DVAL(8),DVEC(8,8),DBI(8),DMI(8),DSTA(8)
      REAL*8  DDSTA(8),DAA,DAB,DAC,DA1,DA2,DA3,DB1,DB2
      REAL*8  DC1,DC2,DC3,DC4,DC5,DDB1,DDB2,DDA1,DDA2
      REAL*8  DW,DZW,DZWT,ARG,DZETA,DPHI

      DPHI = 3.14159265359
      DZETA = Ø.Ø5

      DO 6Ø5 ID = 1,DSIZE
        ARG = 1 - DZETA*DZETA
        DW  = DVAL(ID)*DSQRT(ARG)
        DZW = DZETA * DVAL(ID)
        DB1 = 1 + DW
        DB2 = 1 - DW
        DZWT = DZW * TD
        DA1 = 1.5 * DPHI + TD
        DA2 = 1.5 * DPHI - DW*TD
        DA3 = 1.5 * DPHI + DW*TD
        DC1 = DEXP(-DZWT)
        DC2 = DB1/DZW
        DC3 = DB2/DZW
        DC4 = 1/DC2
        DC5 = 1/DC3
        DAC = DBI(ID)*DPHI/(DMI(ID)*DW)
        DAA = (DC3*DC3/(2*DB2))*DC1*(DC1*(DC5*DCOS(DA1)
     &          + DSIN(DA1)) - (DC5*DCOS(DA3) + DSIN(DA3)))
        DAB = (DC2*DC2/(2*DB1))*DC1*(DC1*(DC4*DCOS(DA1)
     &          + DSIN(DA1)) - (DC4*DCOS(DA2) + DSIN(DA2)))
        DSTA(ID) = DAC * (DAA-DAB)
        DDA1 = DC1*(-2*DZW*(DC5*DCOS(DA1) + DSIN(DA1))
     &          + (-DC5*DSIN(DA1) + DCOS(DA3)))
        DDA2 = DZW*(DC5*DCOS(DA3) + DSIN(DA3)) +
     &            DC5*DW*DSIN(DA3) - DW*DCOS(DA3)
        DDA = (DC3*DC3/(2*DB2)) * DC1 * (DDA1+DDA2)
        DDB1 = DC1*(-2*DZW*(DC4*DCOS(DA1)+DSIN(DA1))
     &           - DC4*DSIN(DA1) + DCOS(DA1))
        DDB2 = DZW*(DC4*DCOS(DA2) + DSIN(DA2)
     &           - DC4*DW*DSIN(DA2) + DW*DCOS(DA2))
        DDB = (DC2*DC2/(2*DB1)) * DC1 * (DDB1+DDB2)
        DDSTA(ID) = DAC * (DDA-DDB)
  6Ø5 CONTINUE

      RETURN
      END
```

```
          SUBROUTINE STAD(ISTA,ISDM,EBI,MRE,STA,EVEC,EVAL,ET)
C
C   Calculation of Natural Coordinates, STA (if ISTA=Ø)
C     or its Derivatives (if ISTA=1), without Damping.
C        Input : ISTA(check), ISDM(dim),EBI,MRE,EVEC,EVAL
C        Output: STA
C
      INTEGER ISTA,ISDM,IEE
      REAL*8 EBI(8),MRE(8),STA(8),ET,EA1,EA2,EA3
      REAL*8 EVAL(8),EVEC(8,8),SSCR1(8,8),SSCR2(8,8)
C
C  SSCR1i = (PHI*Bi)*SIN(1.5PHI+T)*COS(Wi*T) / (Mr*Wi*Wi)
C
      PHI = 3.1415927
      DO 61Ø IE = 1,ISDM
        EA1 = (PHI / (MRE(IE)*EVAL(IE)*EVAL(IE))) * EBI(IE)
        EA2 = 1.5*PHI + ET
        EA3 = EVAL(IE)*ET
        IF (ISTA.EQ.Ø) THEN
          SSCR1(IE,1) = EA1*DSIN(EA2)*(1-DCOS(EA3))
        ELSE
          SSCR1(IE,1) = EA1 * (DCOS(EA2)*(1-DCOS(EA3)) -
     &                EVAL(IE)*DSIN(EA2) * DSIN(EA3))
        ENDIF
  61Ø CONTINUE
C
C  CALCULATE THE VECTOR [PHI]*{ETA}
C
      IEE = 1
      CALL MULTI(EVEC,SSCR1,SSCR2,ISDM,ISDM,IEE)
      DO 615 IE = 1,ISDM
        STA(IE) = SSCR2(IE,1)
  615 CONTINUE
      RETURN
      END



          SUBROUTINE FCDFL(IFDM,FDRT,FDFL,XF,FETA)
C
C  Calculation of Deflection of Fexible Body, FDFL.
C     - XF   = x (POSITION)
C     - FETA = [PHI]*{ETA} AT TIME t (=ET) IN SUBROUTINE "ETAD".
C
      INTEGER IFDM,IFI
      REAL*8 FDRT(8),FDFL,XF,FSCR(8,8),FETA(8),XFD,FZN
      FDFL = Ø.
      DO 62Ø IFI = 1,IFDM
        FZN = FDRT(IFI)
        CALL XMODE(FZN,XF,XFD)
        FDFL = FDFL + XFD*FETA(IFI)
```

```
  62Ø CONTINUE
      RETURN
      END


      SUBROUTINE XMODE(ZN,XLEN,XMOD)
C
C  Calculation of Mode Function.
C
      REAL*8 ZN,ZLEN,XLEN,RLEN,A1,A2,A3,A4,XMOD
      RLEN   =  2ØØØ.
      ZLEN   =  ZN*XLEN / RLEN
        A1   =  DSINH(ZN) + DSIN(ZN)
        A2   =  DCOSH(ZN) + DCOS(ZN)
        A3   =  DCOS(ZLEN) - DCOSH(ZLEN)
        A4   =  DSIN(ZLEN) - DSINH(ZLEN)
      XMOD   = A1 * A3 - A2 * A4
      RETURN
      END


      SUBROUTINE CONS(CNRT,EIGVEC,EIGVAL,RMASS,RCONS,ICDM)
C
C  Calculation of Coefficients Vectors "RCONS(SIZE1)"
C    with Non-Zero Initial Conditions.
C
      INTEGER ICDM
      REAL*8 CNRT(8),EIGVEC(8,8),EIGVAL(8),RMASS(8,8)
      REAL*8 RCONS(8),ATL(8,8),BTL(8,8),RMC,A1,A2,A3,A4,A5
      REAL*8 RS1(8,8),RS2(8,8),RS3(8,8),RMRC(8),RLEN
      REAL*8 RHOA,ZI
      RMC    =  12.ØD-3
      RLEN =  2ØØØ.
      VI   =  1ØØØ.
      RHOA =  (8Ø.ØD-3) / RLEN
      DO 625 IC = 1,ICDM
        ZI   =  CNRT(IC)
        A1   =  DSINH(ZI) + DSIN(ZI)
        A2   =  DCOSH(ZI) + DCOS(ZI)
        A3   =  DSIN(ZI) - DSINH(ZI)
        A4   =  DCOS(ZI) + DCOSH(ZI) - 2.Ø
        A5   =  RLEN / ZI
        ATL(IC,1) = A5*(A1*A3+A2*A4)
        BTL(IC,1) = A1*(DCOS(ZI)-DCOSH(ZI))
     &               -A2*(DSIN(ZI)-DSINH(ZI))
  625 CONTINUE
      CALL TRANPS(EIGVEC,RS1,ICDM,ICDM)

      CALL MULTI(RS1,RMASS,RS2,ICDM,ICDM,ICDM)

      CALL MULTI(RS2,EIGVEC,RS3,ICDM,ICDM,ICDM)
```

```
      DO 630 IC = 1,ICDM
        RMRC(IC) = RS3(IC,IC)
  630 CONTINUE

      L = 1
      CALL MULTI(RS1,ATL,RS2,ICDM,ICDM,L)
      CALL MULTI(RS1,BTL,RS3,ICDM,ICDM,L)

      DO 640 IC = 1,ICDM
        WRITE(*,*) ' RMRC(',IC,')=',RMRC(IC)
        IF (RMRC(IC).EQ.0.) GOTO 650
        RCONS(IC) = (RHOA*RS2(IC,1)+RMC*RS3(IC,1))*VI /
     $              (RMRC(IC)*EIGVAL(IC))
        GOTO 660
  650   WRITE(*,*) ' RMRC(IC) IS "ZERO"'
  660   WRITE(*,*)' RCONS(',IC,')=',RCONS(IC)
  640 CONTINUE

      RETURN
      END


         SUBROUTINE MII(RM,MIRT,RMIJ,I)
C
C  Calculation of Elemetns of [m] when i=j.
C
      REAL*8 MIRT(8),RMIJ,AA
      REAL*8 RLEN,RHOA,RM,ZI,DZ,CI,SI,CHI,SHI,S2I
      REAL*8 D1,D2,D3,TR1,TR2,TR3,TB1,TB2,TB3,AII,BII
      RLEN = 2000.
      RHOA = 80.0D-3 / RLEN
        ZI = MIRT(I)
        DZ = 2.*ZI
        CI = DCOS(ZI)
        SI = DSIN(ZI)
        CHI = DCOSH(ZI)
        SHI = DSINH(ZI)
        S2I = DSIN(DZ)
        D1 = (SHI+SI)
        D2 = (CHI+CI)
        AA = .5 * RLEN / ZI
      TR1 = D1*D1*AA*(2.*ZI+.5*S2I-2.*(SHI*CI+CHI*SI)+SHI*CHI)
      TR2 = 2.*D1*D2*AA*(SI-SHI)*(SI-SHI)
      TR3 = D2*D2*AA*(SHI*CHI - 2.*(CHI*SI-SHI*CI)-.5*S2I)
        AII = TR1 - TR2 + TR3
      TB1 = D1*(CI-CHI) - D2*(SI-SHI)
        BII = TB1*TB1
      RMIJ = RHOA*AII + RM*BII

      RETURN
      END
```

```
      SUBROUTINE MIJ(RM,MJRT,RMIJ,I,J)
C
C  Calculation of Elements of [m] When i.NE.j
C
     REAL*8 MJRT(8),RMIJ,RLEN,RHOA,RM,ZI,ZJ,ZM,ZP,ZS
     REAL*8 CI,CJ,CM,CP,SI,SJ,SM,SP,CHI,CHJ,CHM,CHP
     REAL*8 SHI,SHJ,SHM,SHP,C1,C2,C3,C4,AIJ,BIJ
     REAL*8 TR11,TR12,TR21,TR22,TR31,TR32
     REAL*8 TR41,TR42,TB1,TB2,TB3,TB4
     RLEN = 2ØØØ.
     RHOA = 8Ø.ØE-3 / RLEN
       ZI = MJRT(I)
       ZJ = MJRT(J)
       ZM = ZI - ZJ
       ZP = ZI + ZJ
       ZS = ZI*ZI + ZJ*ZJ
     CI = DCOS(ZI)
     CJ = DCOS(ZJ)
     CM = DCOS(ZM)
     CP = DCOS(ZP)
     SI = DSIN(ZI)
     SJ = DSIN(ZJ)
     SM = DSIN(ZM)
     SP = DSIN(ZP)
     CHI = DCOSH(ZI)
     CHJ = DCOSH(ZJ)
     CHM = DCOSH(ZM)
     CHP = DCOSH(ZP)
     SHI = DSINH(ZI)
     SHJ = DSINH(ZJ)
     SHM = DSINH(ZM)
     SHP = DSINH(ZP)
       C1 = (SHI+SI) * (SHJ+SJ)
       C2 = (SHI+SI) * (CHJ+CJ)
       C3 = (CHI+CI) * (CHJ+CJ)
       C4 = (CHI+CI) * (SHJ+SJ)
     TR11 = ((SM+SHM)/ZM) + ((SP+SHP)/ZP)
     TR12 = 2.*(ZI*(CHJ*SI+SHI*CJ) + ZJ*(SHJ*CI+CHI*SJ)) / ZS
     TR21 = ((CM-CHM)/ZM) + ((CHP-CP)/ZP)
     TR22 = 2.*(ZI*(SHJ*SI+SHI*SJ) + ZJ*(CHJ*CI-CHI*CJ)) / ZS
     TR31 = ((SM-SHM)/ZM) - ((SP-SHP)/ZP)
     TR32 = 2.*(ZI*(CHI*SJ-SHJ*CI) + ZJ*(CHJ*SI-SHI*CJ)) / ZS
     TR41 = ((CHM-CM)/ZM) + ((CHP-CP)/ZP)
     TR42 = 2.*(ZI*(CHI*CJ-CHJ*CI) + ZJ*(SHI*SJ+SHJ*SI)) / ZS
       AIJ = .5*C1*RLEN*(TR11-TR12) - .5*C2*RLEN*(TR21-TR22)
    &          + .5*C3*RLEN*(TR31-TR32) - .5*C4*RLEN*(TR41-TR42)
     TB1 = CI*CJ - CI*CHJ - CHI*CJ + CHJ*CHI
     TB2 = CI*SJ - CI*SHJ - CHI*SJ + CHI*SHJ
     TB3 = SI*SJ - SI*SHJ - SHI*SJ + SHI*SHJ
     TB4 = SI*CJ - SI*CHJ - SHI*CJ + SHI*CHJ
```

```
        BIJ = C1*TB1 - C2*TB2 + C3*TB3 - C4*TB4
      RMIJ = RHOA*AIJ + RM*BIJ

      RETURN
      END



         SUBROUTINE KII(KIRT,RKIJ,I)
C
C   Calculation of Elements of [k] Matrix, When i=j.
C
      REAL*8 KIRT(8),RKIJ,RLEN,REI,ZI,Z2I,CI,SI,CHI,SHI
      REAL*8 S2I,ZI4,A1,A2,A3,DD,TR1,TR2,TR3
      RLEN = 2ØØØ.
      REI = (1.8D+8) * 315.
        ZI = KIRT(I)
        Z2I = 2.*ZI
        CI = DCOS(ZI)
        SI = DSIN(ZI)
        CHI = DCOSH(ZI)
        SHI = DSINH(ZI)
        S2I = DSIN(Z2I)
        ZI4 = (ZI/RLEN)**4
      A1 = ZI4*(SHI+SI)*(SHI+SI)
      A2 = ZI4*(CHI+CI)*(CHI+CI)
      A3 = 2.*ZI4*(SHI+SI)*(CHI+CI)
      DD = .5*RLEN / ZI
        TR1 = DD*(2.*ZI+.5*S2I+SHI*CHI+2.*SHI*CI+2.*CHI*SI)
        TR2 = DD*(SHI*CHI-.5*S2I+2.*CHI*SI-2.*SHI*CI)
        TR3 = DD*(SI+SHI)*(SI+SHI)
      RKIJ = (A1*TR1 + A2*TR2 - A3*TR3) * REI
      RETURN
      END



         SUBROUTINE KIJ(KJRT,RKIJ,I,J)
C
C  Calculation of Elements of [k] Matrix, When i.NE.j
C
      REAL*8 KJRT(8),RKIJ,RLEN,REI,RHLN,ZI,ZJ,ZM,ZP,ZLIJ
      REAL*8 CI,CJ,CM,CP,SI,SJ,SM,SP,CHI,CHJ,CHM,CHP
      REAL*8 SHI,SHJ,SHM,SHP,ZIS,ZJS,AS,A1,A2,A3,A4
      REAL*8 TR11,TR12,TR1,TR21,TR22,TR2,TR31,TR32,TR3
      REAL*8 TR41,TR42,TR4
      RLEN = 2ØØØ
      REI  = (1.8D+8) * 315.
      RHLN = 5ØØ. / 2.
      ZI = KJRT(I)
      ZJ = KJRT(J)
      ZM = ZI - ZJ
      ZP = ZI + ZJ
```

```
      ZLIJ = RLEN / (ZI*ZI + ZJ*ZJ)
        CI = DCOS(ZI)
        CJ = DCOS(ZJ)
        CM = DCOS(ZM)
        CP = DCOS(ZP)
        SI = DSIN(ZI)
        SJ = DSIN(ZJ)
        SM = DSIN(ZM)
        SP = DSIN(ZP)
        CHI = DCOSH(ZI)
        CHJ = DCOSH(ZJ)
        CHM = DCOSH(ZM)
        CHP = DCOSH(ZP)
        SHI = DSINH(ZI)
        SHJ = DSINH(ZJ)
        SHM = DSINH(ZM)
        SHP = DSINH(ZP)
      ZIS = (ZI/RLEN) ** 2
      ZJS = (ZJ/RLEN) ** 2
      AS = ZIS*ZJS
        A1 = AS * (SHI+SI) * (SHJ+SJ)
        A2 = AS * (CHI+CI) * (CHJ+CJ)
        A3 = AS * (CHI+CI) * (SHJ+SJ)
        A4 = AS * (SHI+SI) * (CHJ+CJ)
      TR11 = ((SM+SHM)/ZM) + ((SP+SHP)/ZP)
      TR12 = ZI*(CHJ*SI+SHI*CJ) + ZJ*(CHI*SJ+SHJ*CI)
        TR1 = RHLN*TR11 + ZLIJ*TR12
      TR21 = ((SM-SHM)/ZM) + ((SHP-SP)/ZP)
      TR22 = ZI*(CHI*SJ-SHJ*CI) + ZJ*(CHJ*SI-SHI*CJ)
        TR2 = RHLN*TR21 + ZLIJ*TR22
      TR31 = ((CHP-CP)/ZP) + ((CHM-CM)/ZM)
      TR32 = ZI*(CHI*CJ-CHJ*CI) + ZJ*(SHI*SJ+SHJ*SI)
        TR3 = RHLN*TR31 + ZLIJ*TR32
      TR41 = ((CM-CHM)/ZM) + ((CHP-CP)/ZP)
      TR42 = ZI*(SHI*SJ+SHJ*SI) + ZJ*(CHJ*CI-CHI*CJ)
        TR4 = RHLN*TR41 + ZLIJ*TR42
      RKIJ = (A1*TR1 + A2*TR2 - A3*TR3 - A4*TR4) * REI

      RETURN
      END



         SUBROUTINE INVERS(AINV,BINV,IDIM)
C
C  Calculation of Inverse Matrix Using Gauss Elim. Method.
C
      INTEGER IDIM,DIDIM,L
      REAL*8 AINV(8,8),BINV(8,8),AAIN(8,8)
C
C  SET THE IDENTITY MATRIX
```

```
C
      NP1 = IDIM + 1
      DIDIM = 2*IDIM

      DO 7ØØ II = 1,IDIM
        DO 71Ø JJ = NP1,DIDIM
          L = JJ - IDIM
          AAIN(II,L) = AINV(II,L)
          IF(L.EQ.II) THEN
            AAIN(II,JJ) = 1.
          ELSE
            AAIN(II,JJ) = Ø.
          ENDIF
  71Ø   CONTINUE
  7ØØ CONTINUE

C
C  CALCULATION OF INVERS
C
      DETER = 1.
      DO 72Ø K=1,IDIM
        DETER = DETER*AAIN(K,K)
        KP1 = K + 1
        KPF = 2*IDIM

        DO 73Ø J=KP1,KPF
          AAIN(K,J) = AAIN(K,J) / AAIN(K,K)
  73Ø   CONTINUE

        AAIN(K,K) = 1.
        DO 74Ø I=1,IDIM
          IF(I.EQ.K.OR.AAIN(I,K).EQ.Ø.) GOTO 74Ø
          DO 75Ø J=KP1,KPF
            AAIN(I,J) = AAIN(I,J) - AAIN(I,K) * AAIN(K,J)
  75Ø     CONTINUE
          AAIN(I,K) = Ø.
  74Ø   CONTINUE
  72Ø CONTINUE
C
C  INVERSE MATRIX IN "B"
C
      DO 76Ø I=1,IDIM
        DO 77Ø J=1,IDIM
          N = J + IDIM
          BINV(I,J) = AAIN(I,N)
  77Ø   CONTINUE
  76Ø CONTINUE

      RETURN
      END
```

```
      SUBROUTINE MULTI(A,B,T,IM,IN,IP)
C
C  Multiplication of Matrices [A(mxn)] & [B(nxp)].
C        --  Output: "T" MATRIX  --
C
      INTEGER IM,IN,IP
      REAL*8 A(8,8),B(8,8),T(8,8),TT(8,8)
C
C  TT(I,J) = Ø.Ø
C
      DO 78Ø I=1,IM
        DO 79Ø J=1,IP
          TT(I,J) = Ø.
 79Ø    CONTINUE
 78Ø  CONTINUE
C
C  MULTIPLICATION
C
      DO 8ØØ I=1,IM
        DO 81Ø J=1,IP
          DO 82Ø K=1,IN
            TT(I,J) = A(I,K)*B(K,J) + TT(I,J)
 82Ø      CONTINUE
          T(I,J) = TT(I,J)
 81Ø    CONTINUE
 8ØØ  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ADDI(A,B,T,IM,IN)
C
C  Addition of Matrices [A(mxn)] & [B(mxn)].
C        --  Output: "C" MATRIX --
C
      INTEGER IM,IN
      REAL*8 A(8,8),B(8,8),C(8,8)
      DO 83Ø I=1,IM
        DO 84Ø J=1,IN
          C(I,J) = A(I,J) + B(I,J)
 84Ø    CONTINUE
 83Ø  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SUBT(A,B,C,IM,IN)
C
C  Subtraction of Matrices [A(mxn)] - [B(mxn)].
C        --  Output: "C" MATRIX  --
C
      INTEGER IM,IN
      REAL*8 A(8,8),B(8,8),C(8,8)
      DO 85Ø I=1,IM
        DO 86Ø J=1,IN
          C(I,J) = A(I,J) - B(I,J)
  86Ø   CONTINUE
  85Ø CONTINUE
      RETURN
      END



      SUBROUTINE TRANPS(A,AT,IM,IN)
C
C   Transpose Matrix [A] into [AT]
C
      INTEGER IM,IN
      REAL*8 A(8,8),AT(8,8)
      DO 87Ø I=1,IM
        DO 88Ø J=1,IN
          AT(J,I) = A(I,J)
  88Ø   CONTINUE
  87Ø CONTINUE
      RETURN
      END



      SUBROUTINE THETA(TT,THTT)
C
C   THETA(t)   = PHI*[1+SIN(1.5*PHI+t)
C    THETA(t)" = - PHI*SIN(1.5*PHI+t)
C        Input : TT(time)
C        Output: THTT(2nd derivative of THETA(t))
C
      REAL*8 TT,THTT,PHI,THAR

      PHI  = 3.1415927
      THAR = 1.5*PHI + TT
      THTT = - PHI*DSIN(THAR)
      RETURN
      END
```

```
         SUBROUTINE PRNTM(PRMT,IPDM)
C
C   Print Matrix [PRMT]:
C
      INTEGER IPDM,IPTM,JPTM
      REAL*8 PRMT(8,8)

      WRITE(*,*) '          ----------------------------------'
      DO 9ØØ IPTM = 1,IPDM
        WRITE(*,*) '       ',(PRMT(IPTM,JPTM),JPTM=1,IPDM)
  91Ø   FORMAT('       ',3D2Ø.5)
  9ØØ CONTINUE
      WRITE(*,*) '          ----------------------------------'
      WRITE(*,*) '   '
      RETURN
      END


         SUBROUTINE PRNTV(PRVE,IPDV)
C
C   Print Vector {PRMT}:
C
      INTEGER IPDV,IPTV
      REAL*8 PRVE(8)

      WRITE(*,*) '          ----------------------------------'
        WRITE(*,*) '       ',PRVE(1),PRVE(2),PRVE(3)
  91Ø   FORMAT('       ',3D2Ø.5)
      WRITE(*,*) '          ----------------------------------'
      WRITE(*,*) '   '
      RETURN
      END


         SUBROUTINE ROOT(RTMS,PRRT)
C
C  Calculation of Roots of Chacteristic Equation.
C
      INTEGER SIZE1,FLAG1,FLAG2,I,J
      REAL*8 RLAA(8),PRRT(8),RBCOF(8),RTX,RTDX
      REAL*8 RCA,RTK,RTF,RLEN,YXT,RTDL,RTXL,RTXA,RTXB,XXX,RTT
      REAL*8 RTMS,RHOL
      RLEN    = 2ØØØ.
      RHOL    = 8Ø.ØD-3
```

```
C
C  DEFINE PARAMETERS
C
      SIZE1  =   8
      NN     =   1
      RTX    =   Ø.
      RTDX     =   Ø.1
      VI     = 1ØØØ.
      RCA    = SQRT(1.8D+8* 315. * 2ØØØ. / Ø.Ø8)
      RTK     = RTMS / RHOL
      RTF     = CHREQN(RTX,RTK)
      IF (RTF.LT.Ø.) THEN
          FLAG1  =  1
        ELSE
          IF (RTF.GT.Ø.) FLAG1 = 2
      ENDIF
  92Ø FORMAT( '    ',D15.8,'    ',D15.8)
  93Ø RTX   =  RTX + RTDX
      RTF = CHREQN(RTX,RTK)
      IF (RTF.LE.Ø.) FLAG2 = 1
      IF (RTF.GT.Ø.) FLAG2 = 2
      IF (FLAG1.EQ.FLAG2) THEN
        NN = NN
      ELSE
        XXX = RTX
        RTXA = RTX - RTDX
        CALL NEWTON(XXX,RTT,RTXA,RTK)
        PRRT(NN) = RTT
        RLAA(NN) = RCA * RTT*RTT / (RLEN*RLEN)
        NN = NN + 1
      ENDIF
      FLAG1 = FLAG2
      IF (NN.LE.SIZE1) GOTO 93Ø
      RETURN
      END



        REAL*8 FUNCTION CHREQN(CHZ,CHRK)
C
C  Computation of Characteristic Equation.
C
      REAL*8 CHZ,CHRK,CHTM
      CHTM  = DCOS(CHZ)*DSINH(CHZ) - DSIN(CHZ)*DCOSH(CHZ)
      CHREQN = DCOS(CHZ)*DCOSH(CHZ) + CHRK*CHZ*CHTM + 1
      RETURN
      END
```

```
          SUBROUTINE CHRDER(CDZD,CHRD,CDRK)
C
C  Computation of Derivative of Characteristic Equation.
C
      REAL*8 CDZD,CDRK,ZZ,ZX,CHRD
      ZZ =  DCOS(CDZD)*DSINH(CDZD) - DSIN(CDZD)*DCOSH(CDZD)
      ZX = -DSIN(CDZD)*DCOSH(CDZD) + DCOS(CDZD)*DSINH(CDZD)
      CHRD = ZX + CDRK*ZZ - 2.Ø*CDRK*CDZD*DSIN(CDZD)*DSINH(CDZD)
      RETURN
      END


          SUBROUTINE NEWTON(NTX,NTY,NTA,NTK)
C
C  Computation of Roots of Characteristic Equation
C         Using Newton Method.
C
      INTEGER II, FL1, WHAT,ITT
      REAL*8 NTX,NTY,NTA,SMLST,NTF1,NTF2,NTFT,NTFM
      REAL*8 NTFA,NTFB,CHECK,NTB,FLA,FLB
      REAL*8 NTZ,NTCD,NTK,NTXM,NTXN
      NTB    = NTX
      FL1   = Ø
      II    = Ø
      ITT   = 1Ø
      SMLST  =   .ØØØØØØ5
  94Ø NTZ = NTX
      CALL CHRDER(NTZ,NTCD,NTK)
      NTF2   =  NTCD
      NTF1   =  CHREQN(NTX,NTK)
      NTXN   = NTX - NTF1/NTF2
      CHECK = CHREQN(NTXN,NTK)
      IF (ABS(CHECK).LE.SMLST) THEN
        NTY    =  NTXN
        NTFT   =  CHREQN(NTY,NTK)
        FL1 =   1
      ELSE
        NTX    = NTXN
        IF (II.GT.ITT) THEN
          FL1   = 1
          WRITE(*,*) ' CHECK =',CHECK,' ACCURACY NOT IMPROVED!'
          WRITE(*,*) '  ** TYPE FOLLOWING NUMBER TO CONTINUE**'
          WRITE(*,*) '    ---> "1" TO CONTINUE IN NEWTON.    '
          WRITE(*,*) '    ---> "2" TO USE BI-SECTION METHOD.'
          WRITE(*,*) '    ---> "Ø" TO EXIT THE ITERATION.'
  95Ø     WHAT = Ø
          IF(WHAT.EQ.1) GOTO 96Ø
          IF(WHAT.EQ.2) GOTO 97Ø
          IF(WHAT.EQ.Ø) GOTO 98Ø
          WRITE(*,*)'   RETYPE THE NUMBER, PLEASE!!'
          GOTO 95Ø
```

```
  96Ø     ITT = ITT - 1
          FL1 = Ø
          GOTO 99Ø
C
C  TO COMPUTE THE ROOTS BY BI-SECTION METHOD.
C
  97Ø     NTFA  = CHREQN(NTA,NTK)
          NTFB  = CHREQN(NTB,NTK)
          IF (NTFA.LT.Ø.Ø) THEN
            FLA = Ø.
          ELSE
            FLA = 1.
          ENDIF
          IF (NTFB.LT.Ø.Ø) THEN
            FLB = Ø.
          ELSE
            FLB = 1.
          ENDIF
C
C  TO CALCULATE THE ROOTS.
C
          IF (FLA.EQ.FLB) THEN
            WRITE(*,*) ' ROOT DOES NOT EXIT BETWEEN',NTA,
     &                 ' AND',NTB
            FL1 = 1
          ELSE
            NTXM = (NTA+NTB) / 2.
            NTFM = CHREQN(NTXM,NTK)
            WRITE(*,*) '    *** NTXM =',NTXM,'  NTFM =',NTFM
            IF (NTFM.LT.Ø.Ø) THEN
              FLM = Ø.
            ELSE
              FLM = 1.
            ENDIF
            IF (FLM.NE.FLA) THEN
              NTB = NTXM
              NTFB = NTFM
            ELSE
              NTA = NTXM
              NTFA = NTFM
            ENDIF
            IF (ABS(NTFM).LT.SMLST) THEN
              NTXN = NTXM
              NTFT = NTFM
              FL1 = 1
            ELSE
              FL1 = 1
            ENDIF
          ENDIF

          WRITE(*,*)'  NTA =',NTA,'  NTFA =',NTFA
          WRITE(*,*)'  NTB =',NTB,'  NTFB =',NTFB
          WRITE(*,*)'*TYPE ANY TO CONTINUE (9Ø TO EXIT!).'
```

```
          IF (ANY.EQ.9Ø) FL1 = 1
          IF (FL1.EQ.Ø) GOTO 97Ø
          NTY  =  NTXN
          NTFT =  CHREQN(NTY,NTK)
          GOTO 99Ø
C
C  END OF BI-SECTION METHOD.
C
  98Ø     FL1 = 1
          NTY = NTXN
          NTFT = CHREQN(NTY,NTK)
  99Ø     FL1 = FL1
          ELSE
            FL1 = Ø
          ENDIF
        ENDIF
        II = II + 1
        IF(FL1.EQ.Ø) GOTO 94Ø
        RETURN
        END
```