PL-TR--95-1010

PL-TR--
95-1010

THE DESIGN OF NEURAL CONTROLLER FOR FLEXIBLE MULTIBODY SYSTEMS

Gary G. Yen
Moon K. Kwak

The University of New Mexico
Department of Electrical and Computer Engineering
Room 110, EECE Building
Albuquerque, NM 87131-1356

March 1995

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

PHILLIPS LABORATORY
Space and Missiles Technology Directorate
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

PL-TR--95-1010

This final report was prepared by the University of New Mexico, Albuquerque, New Mexico, under Contract F29601-94-K-0024, Job Order 8809TMGY with Phillips Laboratory, Kirtland Air Force Base, New Mexico. The Laboratory Project Officer-in-Charge was Ken Qassim (VTS).

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been authored by a contractor of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the material contained herein, or allow others to do so, for the United States Government purposes.

If your address has changed, if you wish to be removed from the mailing list, or if your organization no longer employs the addressee, please notify PL/VTS, Kirtland AFB, NM 87117-5776, to help maintain a current mailing list.

This technical report has been reviewed and is approved for publication.

Ken Qassim

KEN QASSIM
Project Officer

FOR THE COMMANDER

L Kevin Slimak

L KEVIN SLIMAK
Chief, Division

HENRY L. PUGH, JR., Colonel, USAF
Director, Space and Missiles
Technology Directorate

DO NOT RETURN COPIES OF THIS REPORT UNLESS CONTRACTUAL OBLIGATIONS OR NOTICE ON A SPECIFIC DOCUMENT REQUIRES THAT IT BE RETURNED.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)

2. REPORT DATE
March 1995

3. REPORT TYPE AND DATES COVERED
Final; October 94 - December 94

4. TITLE AND SUBTITLE
THE DESIGN OF NEURAL CONTROLLER FOR FLEXIBLE MULTIBODY SYSTEMS

5. FUNDING NUMBERS
C: F29601-94-K-0024
PE: 62601F
PR: 8809
TA: TM
WU: GY

6. AUTHOR(S)
Gary G. Yen and Moon K. Kwak

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of New Mexico
Department of Electrical Engineering
Albuquerque NM 87131-1356

8. PERFORMING ORGANIZATION REPORT NUMBER
PL-TR--95-1010

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Phillips Laboratory
Structures and Controls Division
Kirtland AFB NM 87117-5776

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*) A distributive neural control system is advocated for flexible multibody structures. The proposed neural controller is designed to achieve trajectory slewing of structural member as well as vibration suppression for precision pointing capability. The motivation to support such an innovation is to pursue a real-time implementation of a robust and fault tolerant structural controller. The proposed control architecture which takes advantage of the geometric distribution of piezoceramic sensors and actuators has provided a tremendous freedom from computation complexity. In the spirit of model reference adaptive control, we utilize adaptive time-delay radial basis function networks as a building block to allow the neural network to function as an indirect closed-loop controller. The horizon-of-one predictive controllers cooperatively regulates the dynamics of the nonlinear structure to follow the prespecified reference models asymptotically. The proposed control strategy is validated in the experimental facility, called the Planar Articulating Control Experiment which consists of a two-link flexible planar structure constrained to move over a granite table. This paper addresses the theoretical foundation of the architecture and demonstrates its applicability via a realistic structural test bed.

14. SUBJECT TERMS
Flexible Multibody, Structural Modeling, Adaptive Neural Control, Model Reference Adaptive Control

15. NUMBER OF PAGES
44

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

Tables of Contents

LIST OF FIGURES iv

CHAPTER ONE FLEXIBLE MULTIBODY SYSTEMS 1

CHAPTER TWO MODELING OF FLEXIBLE MULTIBODY SYSTEMS 4
2.1 Equation of Motion 4
2.2 Perturbation Method 8
2.3 Zero-Order Kinematical Synthesis 9
2.4 First-Order Kinematical Synthesis 11

CHAPTER THREE NUMERICAL RESULTS AND DISCUSSIONS 14

CHAPTER FOUR NEURAL NETWORK IN CONTROL 19
4.1 Back-Propagation Network 22
4.2 Gaussian Function Network 26
4.3 Adaptive Time-Delay RBF Network 28
4.4 Simulation Study 30

CHAPTER FIVE CONCLUSIONS 34

List of Figures

FIGURE 1.1: PACE Test Article 2

FIGURE 2.1: Flexible Body 5

FIGURE 3.1: Zero-Order Displacements 15

FIGURE 3.2: First-Order Displacements 16

FIGURE 3.3: Time Histories of Angular Displacements 17

FIGURE 3.4: Time Histories of Tip Displacements 18

FIGURE 4.1: BOXES and Neural Network Approximations of Control Law 20

FIGURE 4.2: Vibration Suppression Learning Control Architecture 21

FIGURE 4.3: Trajectory Maneuvering Learning Control Architecture 21

FIGURE 4.4: A Functional Diagram for a Single Neuron 23

FIGURE 4.5: A Multi-layer Back-Propagation Network 24

FIGURE 4.6: A Comparison of Pattern Classification Problem 27

FIGURE 4.7: Simulation Flowchart for Gnu-C Routines 31

FIGURE 4.8: Open-Loop Response of PACE Test Article 32

FIGURE 4.9: Closed-Loop Response of PACE Test Article 33

Chapter 1

Flexible Multibody Systems

The Air Force remains active in space systems, and hardware, such as space robots, rotorcraft and spacecraft. They consist of subsystems which can be described as flexible multibodies. The dynamics and control of flexible multibody systems has been of interest for many years (Refs. 1-16). Identifying, modeling and controlling such systems using various theories with confidence has also become an important issue. At present, a real need exists for the validation and comparison of various modeling and control theories based on an actual hardware experiment. However, compared to theoretical developments and number of computer programs available, experimental verification has never been conducted. To this end, the Phillips Laboratory at Kirtland AFB has constructed a flexible multibody structure called PACE(Planar Articulating Controls Experiment) which consists of 2 flexible beams connected in series with motors at both the hub and the elbow joint (Fig. 1.1).

In recent years, the interest has broadened so as to include maneuvering and control. This narrows the choice of formulations significantly, as the formulation must be consistent with the control task. A set of equations of motion suited for the control task can be formulated by means of Lagrange's equations for flexible bodies in terms of quasi-coordinates. The advantage of this approach is that it yields equations in terms of body axes, which are the same axes as those used to express control forces and torques. In using the approach of Ref. 12 to derive equations of motion for a chain of flexible multibody systems, it is convenient to adopt a kinematical procedure permitting the expression of the velocity vector of a nominal point in a typical body in terms of the velocity vector of the preceding body in the chain. The resulting differential equations are nonlinear and hybrid (Ref. 13), where the term "hybrid" implies that the equations for the rigid-body translations and rotations are ordinary differential equations and those for the elastic motions are partial differential equations. Because maneuvering and control design in terms of hybrid equations is not feasible, the partial differential equations must be transformed into sets of ordi-

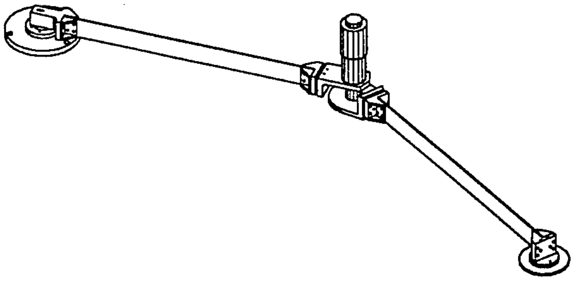

Figure 1.1: PACE Test Article

nary differential equations by means of a discretization-in-space procedure, such as the finite element method (Ref. 14) or a Rayleigh-Ritz based substructure synthesis (Ref. 15). The resulting formulation consists of a high-order set of nonlinear ordinary differential equations. A common approach to control design requires the solution of a two-point boundary-value problem, which is not feasible for high-order systems, so that a different approach is advisable.

The nonlinearity enters into the differential equations through the rigid-body motions. Indeed, the elastic motions tend to be small. In view of this, it appears natural to conceive of a perturbation approach whereby the rigid-body motions can be regarded as being of zero-order in magnitude and the elastic motions as being of first-order in magnitude. This approach permits dividing the problem into a low-dimensional set of nonlinear zero-order equations for the rigid-body motions and a high-dimensional set of linear first-order equations for the elastic motions and the perturbations in the rigid-body motion, where the order is to be taken in a perturbation sense. Note that, because the zero-order solution enters into the first-order equations as a known function of time, the first-order equations represent a time-varying system. Moreover, the system is subjected to persistent disturbances. The perturbation approach just described, first proposed in Ref. 16, was used in Refs. 17-21 to maneuver and control flexible spacecraft.

The kinematical synthesis of Refs. 16-21 works quite well in the case in which the number of bodies in the chain is relatively small. When the number of bodies is larger than three, difficulties can be expected, so that a different approach is advisable. In Refs. 14-19, the kinematical synthesis was implemented before the derivation of the equations of motion was carried out. In this paper, we consider a procedure whereby the equations of motion are derived first for each individual flexible body. Then, the sets of equations for the individual bodies are assembled into a global set by invoking the kinematical relations described above. In the process, the redundant coordinates and velocities resulting from considering the individual bodies separately are eliminated. It is convenient to carry out the kinematical synthesis on the zero-order problem and first-order problem separately. Implementation of the kinematical synthesis is based on recursive relations that lend themselves to ready computer coding. The resulting zero- and first-order global sets of equations are particularly suited for maneuvering and control design, respectively.

The zero-order nonlinear equations govern the maneuver as if the system consisted of articulated rigid bodies where the maneuver amounts to driving the system from an initial state to a final state. The equations can be solved open-loop or closed-loop. For minimum-time maneuvering, the control law is bang-bang. The simplest approach is to carry out the maneuver by means of actuators that impart predetermined motions to the substructures relative to one another. The first-order equations govern the elastic vibrations and the perturbations in the rigid-body motions. They contain the zero-order solution as a known function of time. As a result, the system is time-varying. Moreover, it is subjected to persistent disturbances caused by the maneuver. The process can be likened to that in which the system must follow a reference state. In this case, the reference state is defined by the rigid-body maneuvering, which is characterized by zero elastic states. Then, the first-order equations are simply the equations in terms of the difference between the actual states and the reference states, where this difference can be identified as perturbations in the state variables. Control of the first-order system is carried out closed-loop and includes a disturbance accommodation procedure. The latter depends on the type of disturbance, which in turn depends on the type of maneuver performed.

As a numerical example, PACE with direct DC motors is considered, in which bang-bang control voltages are applied to shoulder and elbow motors. Numerical results show the effectiveness of the new algorithm developed in this paper.

Chapter 2

Modeling of Flexible Multibody Systems

2.1 Equation of Motion

Our interest lies in deriving equations of motion capable of describing the problem of maneuvering and control of structures in the form of an articulated chain of 2 flexible substructures. To this end, we propose to use a kinematical synthesis procedure whereby the motion is referred to sets of local body axes embedded in the undeformed substructures and the motion of one substructure is described in terms of the motion of the preceding substructure in the chain. This kinematical procedure obviates the need for constraint equations.

The approach used in this report is to derive equations of motion for the individual substructures separately and then impose kinematical relations of the type described earlier to obtain system's equations of motion. Although the approach is used for the case of a two-link flexible body system, the approach can be extend to the case of arbitrary N flexible multibody systems.

Let us consider a typical flexible substructure moving on a horizontal surface (Fig. 2.1) and introduce the inertial axes XY with the origin at O and a set of body axes $x_s y_s$ with the origin at S and embedded in the undeformed substructure. Then, we can write the position vector of a typical point in the substructure with the spatial coordinates given symbolically as follows:

$$\vec{W}_s = \hat{n}^T(R_s + C_s^T u_s) = \hat{n}^T W_s \tag{2.1}$$

where $\hat{n} = [\vec{I}\ \vec{J}]^T$ represents the column matrix consisting of the unit vectors in X- and Y-directions corresponding to inertial coordinates, $C_s = C(\theta_s)$ is the matrix of the direction cosines which is given by

$$C_s = \begin{bmatrix} c\theta_s & s\theta_s \\ -s\theta_s & c\theta_s \end{bmatrix} \tag{2.2}$$

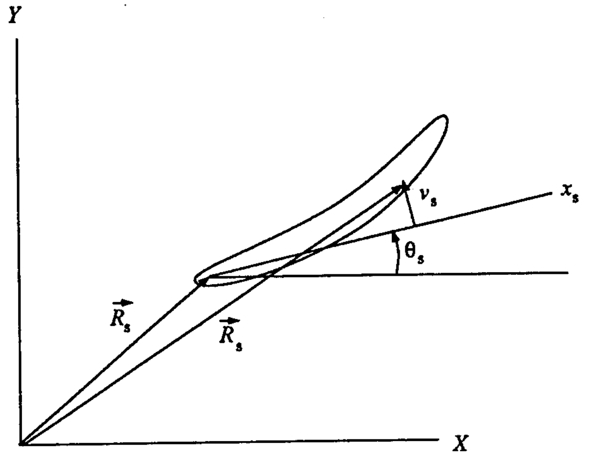

Figure 2.1: Flexible Body

In addition, R_s is the radius vector from I to S, and u_s includes the radius vector from S to a typical point in s and the elastic displacement vector of the same point relative to the body axes $x_s y_s$, respectively. Thus, we can write

$$R_s = \left\{ \begin{array}{c} R_{sx} \\ R_{sy} \end{array} \right\} \qquad u_s = \left\{ \begin{array}{c} x_s \\ v_s \end{array} \right\} \tag{2.3}$$

where v_s represents the elastic displacement. Note that R_s is in terms of inertial coordinates and x_s and v_s are in terms of components along the body axes. We propose to derive the equations of motion by the approach of Ref. 1. In fact, the equations of motion for the individual bodies are identical to the hybrid equations derived here, so that here we merely present the pertinent material.

The Lagrangian formulation requires the kinetic energy, which in turn requires velocities. Assuming that axes $x_s y_s$ rotates with the angular velocity $\dot{\theta}_s$ relative to the inertial axes, the velocity vector of a typical point in s can be easily derived. Since we are concerned with the kinetic energy, let us derive the velocity squared. Hence, using Eq. (2.1), we can obtain

$$\begin{aligned} \frac{1}{2}\dot{\vec{W}}_s \cdot \dot{\vec{W}}_s &= \frac{1}{2}\dot{W}_s^T \dot{W}_s \\ &= \frac{1}{2}\dot{R}_s^T \dot{R}_s - \dot{\theta}_s \dot{R}_s^T D_s^T u_s + R_s^T C_s^T \dot{u}_s \\ &\quad + \frac{1}{2}\dot{\theta}_s^2 u_s^T u_s - \dot{\theta}_s^2 u_s^T \tilde{I}_s \dot{u}_s - \frac{1}{2}\dot{u}_s^T \dot{u}_s \end{aligned} \tag{2.4}$$

where

$$D_s = \begin{bmatrix} s\theta_s & -c\theta_s \\ c\theta_s & s\theta_s \end{bmatrix}, \quad \tilde{I} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.5}$$

in which $c\theta = \cos\theta$ and $s\theta = \sin\theta$.

For chains of flexible substructures such as those under consideration here, no closed-form solution of the hybrid differential equations is possible, so that we must be content with an approximate solution. This implies discretization in space of the partial differential equations, which amounts to replacing the partial differential equations by sets of ordinary differential equations. The discretization can be carried out by means of the finite element method of the classical Rayleigh-Ritz method. In either case, we express the elastic displacement vector in the following form:

$$u_s = \begin{bmatrix} x_s & 0 \\ 0 & \Phi_s(x_s) \end{bmatrix} \begin{Bmatrix} 1 \\ q_s \end{Bmatrix} \tag{2.6}$$

where $\Phi_s(x_s)$ is the vector of admissible functions and q_s is the vector of generalized coordinates. We will assume that the admissible functions introduced in Eq. (2.5) automatically satisfy the geometric boundary conditions. In addition, the control design for the hybrid equation is not feasible, so that the use of the discretization approach can be justified.

Inserting Eq. (2.5) into Eqs. (2.4) and integrating over the length results in

$$\begin{aligned} \mathcal{T} &= \frac{1}{2}m_s\dot{R}_s^T\dot{R}_s - \dot{\theta}_s\dot{R}_s^T D_s^T(LS_s + N\overline{\Phi}_s q_s) + \dot{R}_s^T C_s^T N\overline{\Phi}_s\dot{q}_s \\ &\quad + \frac{1}{2}I_s\dot{\theta}_s^2 + \dot{\theta}_s\widetilde{\Phi}_s\dot{q}_s + \frac{1}{2}\dot{q}_s^T M_s\dot{q}_s \end{aligned} \tag{2.7}$$

where

$$m_s = \int_0^{\ell_s} \overline{m}_s dx_s \qquad S_s = \int_0^{\ell_s} \overline{m}_s x_s dx_s \qquad I_s = \int_0^{\ell_s} \overline{m}_s x_s^2 dx_s \tag{2.8}$$

in which $\overline{m}_s$ is the mass density per unit length. In addition,

$$\overline{\Phi}_s = \int_0^{\ell_s} \overline{m}_s \Phi_s dx_s \qquad \widetilde{\Phi}_s = \int_0^{\ell_s} \overline{m}_s x_s \Phi_s dx_s \tag{2.9}$$

$$M_s = \int_0^{\ell_s} \overline{m}_s \Phi_s^T \Phi_s dx_s \qquad L = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad N = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.10}$$

Moreover, the potential energy can be expressed in the following form:

$$\mathcal{V}_s = [u_s, u_s] \tag{2.11}$$

where [ ] denoted symbolically an energy integral and is a measure of the strain energy. Since we are concerned with the beam bending vibration, the elastic potential takes the form

$$\mathcal{V}_s = \frac{1}{2}\int_0^L EI_s \left(\frac{\partial^2 u_s(x,t)}{\partial x_s^2}\right)^2 dx_s \tag{2.12}$$

Inserting Eq. (2.6) into Eq. (2.12), we obtain the potential energy in the discretized form:

$$\mathcal{V}_s = \frac{1}{2} q_s^T K_s q_s \tag{2.13}$$

where

$$K_s = \int_0^L EI_s \frac{d^2\Phi}{dx_s^2}^T \frac{d^2\Phi}{dx_s^2} dx \tag{2.14}$$

represents the substructure stiffness matrix.

The virtual work for the system can be expressed in the following form:

$$\delta\mathcal{W} = T_s \delta\theta_s \tag{2.15}$$

where T_s is the torque applied to the origin of the body axes $x_s y_s$.

Hence, the Lagrange's equations of motion for the system shown can be written as

$$\frac{d}{dt}\left(\frac{\partial\mathcal{T}}{\partial\dot{R}_s}\right) = 0 \tag{2.16}$$

$$\frac{d}{dt}\left(\frac{\partial\mathcal{T}}{\partial\dot{\theta}_s}\right) - \frac{\partial\mathcal{T}}{\partial\dot{\theta}_s} = T_s \tag{2.17}$$

$$\frac{d}{dt}\left(\frac{\partial\mathcal{T}}{\partial\dot{q}_s}\right) - \frac{\partial\mathcal{T}}{\partial q_s} + \frac{\partial\mathcal{V}}{\partial q_s} = 0 \tag{2.18}$$

Inserting Eqs. (2.7), (2.13) and (2.15) into Eqs. (2.16), (2.17) and (2.18). we can obtain

$$\begin{aligned} m_s\ddot{R}_s - \ddot{\theta}_s D_s^T(LS_s + N\overline{\Phi}_s q_s) \\ +C_s^T N\overline{\Phi}_s \ddot{q}_s - \dot{\theta}_s^2 C_s^T(LS_s + N\overline{\Phi}_s q_s) - 2\dot{\theta}_s D_s^T N\overline{\Phi}_s \dot{q}_s = 0 \end{aligned} \tag{2.19}$$

$$-\ddot{R}_s^T D_x^T(LS_s + N\overline{\Phi}_s q_s) + I_s\ddot{\theta}_s + \widetilde{\Phi}_s\ddot{q}_s = T_s \tag{2.20}$$

$$\overline{\Phi}_s^T I_2^T C_s \ddot{R}_s + \widetilde{\Phi}_s^T\ddot{\theta}_s + M_s\ddot{q}_s + K_s q_s = 0 \tag{2.21}$$

The above equations represent the nonlinear ordinary differential equations for the single substructure. Imposing the kinematical relations to the above equations to link two adjacent substructures is not an easy task. In addition, the control design for the nonlinear system is not easy. Thus, we propose the perturbation method in the next section to ease the kinematical synthesis and numerical calculations.

2.2 Perturbation Method

Designing the maneuvering and control for articulated systems of substructures is very difficult, especially if the design is to be optimal in some fashion. The difficulty can be traced to the fact that the system is nonlinear and of high order. The nonlinearity can be attributed to the rigid-body motions and the high order to the elastic motions. The perturbation approach is based on the simple observation that rigid-body motions tend to be large compared to the elastic motions. Consistent with this, let us assume that the translations and rotation of the body can be divided into zero-order terms and first-order terms in magnitude. Elastic displacements are assumed to be small so that the generalized displacements associated with elastic motion can be regarded as a first-order term. Thus, we may write

$$R_s = R_{s0} + R_{s1} \qquad \theta_s = \theta_{s0} + \theta_{s1} \qquad T_s = T_{s0} + T_{s1} \tag{2.22}$$

Inserting Eq. (2.22) into trigonometric functions yields the following relations.

$$c\theta_s = c\theta_{s0} - s\theta_{s0}\theta_{s1} \qquad s\theta_s = s\theta_{s0} + c\theta_{s0}\theta_{s1} \tag{2.23}$$

which lead to

$$C_s = C_{s0} - D_{s0}\theta_{s1} \qquad D_s = D_{s0} + C_{s0}\theta_{s1} \tag{2.24}$$

Introducing Eqs. (2.22) and (2.24) into Eqs. (2.19), (2.20) and (2.21), we can obtain the zero-order equations of motion

$$m_s\ddot{R}_{s0} - D_{s0}^T L S_s \ddot{\theta}_{s0} - \dot{\theta}_{s0}^2 C_{s0}^T L S_s = 0 \tag{2.25}$$

$$-S_s L^T D_{s0}\ddot{R}_{s0} + I_s\ddot{\theta}_{s0} = T_{s0} \tag{2.26}$$

as well as the first-order equations of motion

$$\begin{aligned} m_s\ddot{R}_{s1} - S_s D_{s0}^T L\ddot{\theta}_{s1} + C_{s0}^T N\overline{\Phi}_s q_s - 2\dot{\theta}_{s0}C_{s0}^T L S_s\dot{\theta}_{s1} - 2\dot{\theta}_{s0}D_{s0}^T N\overline{\Phi}_s\dot{q}_s \\ -(\ddot{\theta}_{s0}C_{s0}^T - \dot{\theta}_{s0}^2 D_{s0})LS_s\theta_{s1} - (\ddot{\theta}_{s0}D_{s0}^T + \dot{\theta}_{s0}^2 C_{s0})N\overline{\Phi}_s q_s = 0 \end{aligned} \tag{2.27}$$

$$-S_s L^T D_{s0}\ddot{R}_{s1} + I_s\ddot{\theta}_{s1} + \widetilde{\Phi}_s\ddot{q}_s - S_s L^T C_{s0}\ddot{R}_{s0}\theta_{s1} - \ddot{R}_{s0}^T D_{s0}^T N\overline{\Phi}_s q_s = T_{s1} \tag{2.28}$$

$$\begin{aligned} \overline{\Phi}_s^T N^T C_{s0}\ddot{R}_{s1} \quad &+ \quad \widetilde{\Phi}_s\ddot{\theta}_{s1} + M_s\ddot{q}_s - \overline{\Phi}_s^T L_s^T D_{s0}\ddot{R}_{s0}\theta_{s1} + K_s q_s \\ &= \quad -\ddot{\theta}_{s0}\widetilde{\Phi}_s^T - \overline{\Phi}_s^T N^T C_{s0}\ddot{R}_{s0} \end{aligned} \tag{2.29}$$

In matrix form, we have

$$\mathcal{M}_{s0}\ddot{x}_{s0} = f_{s0} + d_{s0} \tag{2.30}$$

$$\mathcal{M}_{s1}\ddot{x}_{s1} + \mathcal{C}_{s1}\dot{x}_{s1} + \mathcal{K}_{s1}x_{s1} = f_{s1} + d_{s1} \tag{2.31}$$

where

$$x_{s0} = \begin{bmatrix} R_{s0} \\ \theta_{s0} \end{bmatrix} \qquad f_{s0} = \begin{bmatrix} 0 \\ T_{s0} \end{bmatrix} \qquad d_{s0} = \begin{bmatrix} \dot{\theta}_{s0}^2 S_s C_{s0}^T L \\ 0 \end{bmatrix} \tag{2.32}$$

$$x_{s1} = \begin{bmatrix} R_{s1} \\ \theta_{s1} \\ q_s \end{bmatrix} \qquad f_{s1} = \begin{bmatrix} 0 \\ T_{s1} \\ 0 \end{bmatrix} \tag{2.33}$$

$$d_{s1} = \begin{bmatrix} 0 \\ 0 \\ -\ddot{\theta}_{s0}\widetilde{\Phi}_s^T - \overline{\Phi}_s^T N^T C_{s0}\ddot{R}_{s0} \end{bmatrix} \tag{2.34}$$

and

$$\mathcal{M}_{s0} = \begin{bmatrix} m_s & -S_s D_{s0}^T L \\ -S_s L^T D_{s0} & I_s \end{bmatrix} \tag{2.35}$$

$$\mathcal{M}_{s1} = \begin{bmatrix} m_s & -S_s D_{s0}^T L & C_{s0}^T N\overline{\Phi}_s \\ -S_s L^T D_{s0} & I_s & \widetilde{\Phi}_s \\ \overline{\Phi}_s^T N^T C_{s0} & \widetilde{\Phi}_s^T & M_s \end{bmatrix} \tag{2.36}$$

$$\mathcal{C}_{s1} = \begin{bmatrix} 0 & -2\dot{\theta}_{s0}^T & -2\dot{\theta}_{s0} D_{s0}^T N\overline{\Phi}_s \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.37}$$

$$\mathcal{K}_{s1} = \begin{bmatrix} 0 & -(\ddot{\theta}_{s0} C_{s0}^T - \dot{\theta}_{s0}^2 D_{s0}) L S_s & -(\ddot{\theta}_{s0} D_{s0}^T + \dot{\theta}_{s0}^2 C_{s0}) N\overline{\Phi}_s \\ 0 & -S_s L^T C_{s0}\ddot{R}_{s0} & -\ddot{R}_{s0} D_{s0}^T N\overline{\Phi}_s \\ 0 & -\overline{\Phi}_s^T N^T D_{s0}\ddot{R}_{s0} & K_s \end{bmatrix} \tag{2.38}$$

The main advantage of using the perturbation method is in that the zero-order equations can be solved independently of the first-order equations. The zero-order equations are nonlinear but in low order. Once the zero-order equations are solved, the solution of the zero-order equations enters into the first-order equations and complete the numerical calculations. So far, we are concerned with the single substructure. In the next section, we will see how to assemble individual equations into a global equations of motion by means of the kinematical synthesis.

2.3 Zero-Order Kinematical Synthesis

The derivation of the equations for the individual substructures was carried out in Sec. 2.2 as if the substructures were free to move relative to one another. In reality, however, they are all part of an interacting system of flexible bodies, so that the motions of the various substructures are coupled. Indeed, because the various substructures are hinged to one another, the motion of a given hinge is accounted for several times, once for each substructure sharing the hinge. As a result, the motions defined by the independent equations of motion derived in Sec. 2.2 contain redundant coordinates. In this regard, we note that hinges constrain the translational motions but leaves the rotational motions free. The kinematical synthesis to be introduced shortly is designed to eliminate the surplus coordinates. We propose to carry out the kinematical synthesis for the zero-order and first-order equations separately. Then, we can conclude that the zero-order rigid-body displacements and velocities of the substructures are related as follows:

$$R_{s0} = \sum_{i=1}^{s-1} C_{i0}^T L\ell_i \tag{2.39}$$

$$\theta_{s0} = \theta_{10} + \sum_{i=1}^{s-1} \beta_{i+1,0} \tag{2.40}$$

The first and second derivatives with respect to time can be easily obtained.

$$\dot{R}_{s0} = -\sum_{i=1}^{s-1} D_{i0}^T L\ell_i \dot{\theta}_{i0} \tag{2.41}$$

$$\dot{\theta}_{s0} = \dot{\theta}_{10} + \sum_{i=1}^{s-1} \dot{\beta}_{i+1,0} \tag{2.42}$$

$$\ddot{R}_{s0} = -\sum_{i=1}^{s-1} D_{i0}^T L\ell_i \ddot{\theta}_{i0} - \sum_{i=1}^{s-1} C_{i0}^T L\ell_i \dot{\theta}_{i0}^2 \tag{2.43}$$

$$\ddot{\theta}_{s0} = \ddot{\theta}_{10} + \sum_{i=1}^{s-1} \ddot{\beta}_{i+1,0} \tag{2.44}$$

Let us confine the case to a two-link flexible body system such as PACE. In this case, the origin of the first body is identical to the origin of the inertial axes and the body is rotating on a plane. The second body is connected to the first body through the pivot joint. The zero-order system amounts to the two-link rigid body system. Thus, the overall degree-of-freedom is two. The first variable can be chosen as an absolute angle of the first body with respect to the inertial axes and the second variable can be chosen as the relative angle between the body axes of the second body and the body axes of the first body. Thus, we may write for the global zero-order displacement as follows:

$$z_0 = [\theta_{10}\ \beta_{20}]^T \tag{2.45}$$

The acceleration vector of each body can be generally written as

$$\ddot{x}_{s0} = E_{s0}\ddot{z}_0 + d_{s0}\ , \quad s = 1, 2 \tag{2.46}$$

It can be readily seen that

$$E_{10} = \begin{bmatrix} 0_{21} & 0_{21} \\ 1 & 0 \end{bmatrix} , \; d_{10} = \begin{bmatrix} 0_{21} \\ 0 \end{bmatrix} \tag{2.47}$$

for the first body and

$$E_{20} = \begin{bmatrix} -D_{10}^T I_1 \ell_1 & 0_{21} \\ 1 & 1 \end{bmatrix} , \; d_{20} = \begin{bmatrix} -\dot{\theta}_{10}^2 C_{10}^T I_1 \ell_1 \\ 0 \end{bmatrix} \tag{2.48}$$

for the second body where 0_{nm} represents an $n \times m$ null matrix. Inserting Eqs. (2.47) and (2.48) into Eq. (2.30) and adding the results, we can obtain the global zero-order equation.

$$\mathcal{M}_0 \ddot{z}_0 = d_0 \tag{2.49}$$

where

$$\mathcal{M}_0 = \sum_{s=1}^{2} E_{s0}^T \mathcal{M}_{s0} E_{s0} \; , \; d_0 = \sum_{s=1}^{2} E_{s0}^T (f_{s0} - \mathcal{M}_{s0} d_{s0}) \tag{2.50}$$

It can be observed from Eq. (2.49) that the numerical simulation of the global zero-order equation can be carried out independently of the first-order equation so that the maneuvering control can be also designed independently. Hence, the maneuvering motions excite beam vibrations which in turn cause perturbed motions in rotational motions.

2.4 First-Order Kinematical Synthesis

The kinematical synthesis for the first-order equations is carried out in the same way as for the zero-order equations. The derivation of the constraint relations is tedious and is omitted here for brevity. The basic recursive relations are

$$R_{s1} = \sum_{i=1}^{s-1} \left[C_{i0}^T N \Phi_i(\ell_i) q_i - D_{i0}^T L \ell_i \theta_{i1} \right] \tag{2.51}$$

$$\theta_{s1} = \theta_{11} + \sum_{i=1}^{s-1} \left[\beta_{i+1,1} + \Phi_i'(\ell_i) q_i \right] \tag{2.52}$$

The first and second derivatives with respect to time are as follows:

$$\dot{R}_{s1} = \sum_{i=1}^{s-1} \left[C_{i0}^T N \Phi_i(\ell_i) \dot{q}_i - \dot{\theta}_{i0} D_{i0}^T N \Phi_i(\ell_i) q_i - D_{i0}^T L \ell_i \dot{\theta}_{i1} - \dot{\theta}_{i0} C_{i0}^T L \ell_i \theta_{i1} \right] \tag{2.53}$$

$$\dot{\theta}_{s1} = \dot{\theta}_{11} + \sum_{i=1}^{s-1} \left[\dot{\beta}_{i+1,1} + \Phi_i'(\ell_i) \dot{q}_i \right] \tag{2.54}$$

$$\begin{aligned}\ddot{R}_{s1} \;=\; & \sum_{i=1}^{s-1} \Big[C_{i0}^T N \Phi_i(\ell_i) \ddot{q}_i - 2\dot{\theta}_{i0} D_{i0}^T N \Phi_i(\ell_i) \dot{q}_i - \ddot{\theta}_{i0} D_{i0}^T N \Phi_i(\ell_i) q_i \\ & - D_{i0}^T L \ell_i \ddot{\theta}_{i1} - 2\dot{\theta}_{i0} C_{i0}^T L \ell_i \dot{\theta}_{i1} - \ddot{\theta}_{i0} C_{i0}^T L \ell_i \theta_{i1} \Big] \end{aligned} \tag{2.55}$$

$$\ddot{\theta}_{s1} = \ddot{\theta}_{11} + \sum_{i=1}^{s-1} \Big[\ddot{\beta}_{i+1,1} + \Phi_i'(\ell_i) \ddot{q}_i \Big] \tag{2.56}$$

If we confine ourselves to the two-link flexible body system as we did in the previous section, the substructure first-order displacement vector can be expressed in terms of the global first-order displacement vector.

$$x_{s1} = E_{s1} z_1 \ , \quad s = 1, 2 \tag{2.57}$$

where

$$z_1 = [\theta_{11} \; q_1 \; \beta_{21} q_2]^T \tag{2.58}$$

It can be readily seen that

$$E_{11} = \begin{bmatrix} 0_{21} & 0_{2n} & 0_{21} & 0_{2n} \\ 1 & 0_{1n} & 0 & 0_{1n} \\ 0_{n1} & I_{nn} & 0_{n1} & 0_{nn} \end{bmatrix} \tag{2.59}$$

for the first body and

$$E_{21} = \begin{bmatrix} -D_{10}^T I_1 \ell_1 & C_{10}^T I_2 \phi_1(\ell_1) & 0_{21} & 0_{2n} \\ 1 & \phi_1'(\ell_1) & 0 & 0_{1n} \\ 0_{n1} & 0_{nn} & 0_{n1} & I_{nn} \end{bmatrix} \tag{2.60}$$

for the second body where I_{nn} represents an $n \times n$ identity matrix. Inserting Eqs. (2.59) and (2.60) into Eq. (2.31) and adding the results, we can obtain the global first-order equation.

$$\mathcal{M}_1 \ddot{z}_1 + \mathcal{C}_1 \dot{z}_1 + \mathcal{K}_1 z_1 = d_1 \tag{2.61}$$

where

$$\mathcal{M}_1 = \sum_{s=1}^{2} E_{s1}^T \mathcal{M}_{s1} E_{s1} \ , \quad \mathcal{C}_1 = \sum_{s=1}^{2} E_{s1}^T (2\mathcal{M}_{s1} \dot{E}_{s1} + \mathcal{C}_{s1} E_{s1}) \tag{2.62}$$

$$\mathcal{K}_1 = \sum_{s=1}^{2} E_{s1}^T (\mathcal{M}_{s1} \ddot{E}_{s1} + \mathcal{C}_{s1} \dot{E}_{s1} + \mathcal{K}_{s1} E_{s1}) \ , \quad d_1 = \sum_{s=1}^{2} E_{s1}^T d_{s1} \tag{2.63}$$

It can be observed from Eqs. (2.61), (2.62) and (2.63) that the zero-order solution enters into the coefficient matrices of the global first-order equation so that the global first-order equation is in fact the time-varying linear second-order differential equation. Numerical simulation is carried out in a way that first we solve the global zero-order equation and then insert the zero-order solution to the coefficients of the global first-order equation and solve the global first-order equation. It should be noted here that the actual rotational motions are the sum of two solutions as indicated by Eq. (2.22).

Chapter 3

Numerical Results and Discussions

A computer program based on the algorithm developed here was written using MATLAB. We consider the PACE shown in Fig. 1.1 as a numerical example. Each body represents a uniform slender beam with tip masses at both ends. The maneuver consists of bang-bang controls applied to each DC motors. As a result, the beams undergo bending deformation alone.

Figures 3.1 through 3.4 show time histories of the response of the system. The time integration of equations was carried out by means of the wilson θ method. Figure 3.1 shows zero-order response of the system. It can be seen from Fig. 3.1 that the rigid-body rotations are made through the torques applied to the shoulder and the elbow. Figure 3.2 shows first-order rotational responses of the system, which are in fact the per tubed motions due to the elastic vibrations of each beam. Thus, the actual rotational responses of the system are the combination of Figs. 3.1 and 3.2 as shown in Fig. 3.3.

Figure 3.4 shows time responses of tip displacements of each beam. As can be seen from Fig. 3.4, the fundamental modes are mostly excited by the maneuvering motions. However, dynamic characteristics of the whole system becomes more complicated than a single-body system since there are interactions between two bodies. It should be mentioned here that an efficient numerical integration scheme should be developed to cope with the nonlinearity of the system equations.

There remains a question regarding what kind of data is good for neural network training. Since there is no available guideline on this, we generated the data set by applying random signals to the shoulder and elbow motors, which results in random motions. It should be mentioned here that based on the past PACE experiments the motions of PACE are not repeatable which may cause serious problems in neural network control since neural networks are simply the

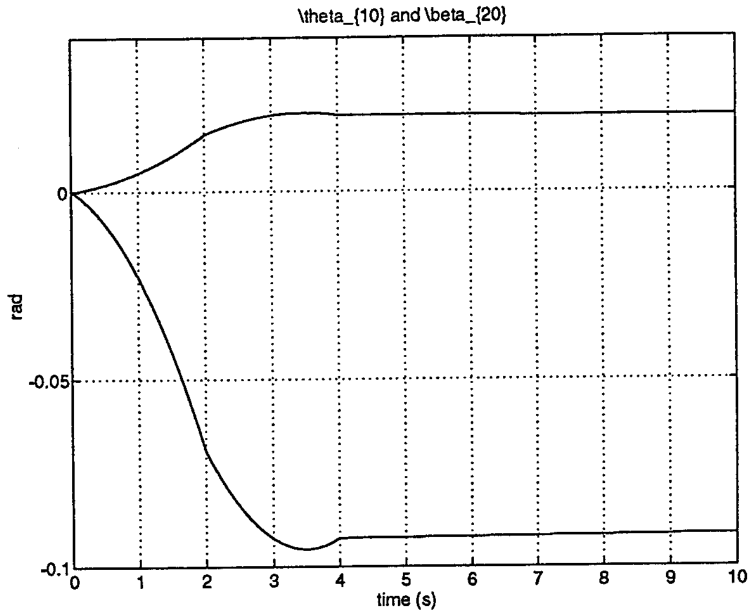

Figure 3.1: Zero-Order Displacements

mapping of the input and output. When there exists uncertainty, there is no definite answer on the stability of neural network control. We will have ample opportunities to explore the neural network control for the new field such as control of flexible multibody systems.

The first part of this report is concerned with the development of a mathematical formulation capable of treating the problem of maneuvering and control of flexible multibody systems. The formulation is based on equations of motion in terms of coordinates derived for each substructure independently of the other substructures. The individual substructures are made to act as a single structure by means of a consistent kinematical synthesis. The net effect of this synthesis is the elimination of the redundant coordinates resulting from the original treatment of the substructures as if they were independent.

In most problems of interest, the elastic motions are small compared to the maneuvering motions. In recognition of this, a perturbation approach is developed whereby the rigid-body maneuvering of the system defines the zero-order problem and the elastic motions and perturbations from the rigid-body maneuvering define the first-order problem., where the term "order" refers to magnitude in a perturbation sense. The kinematical synthesis mentioned earlier

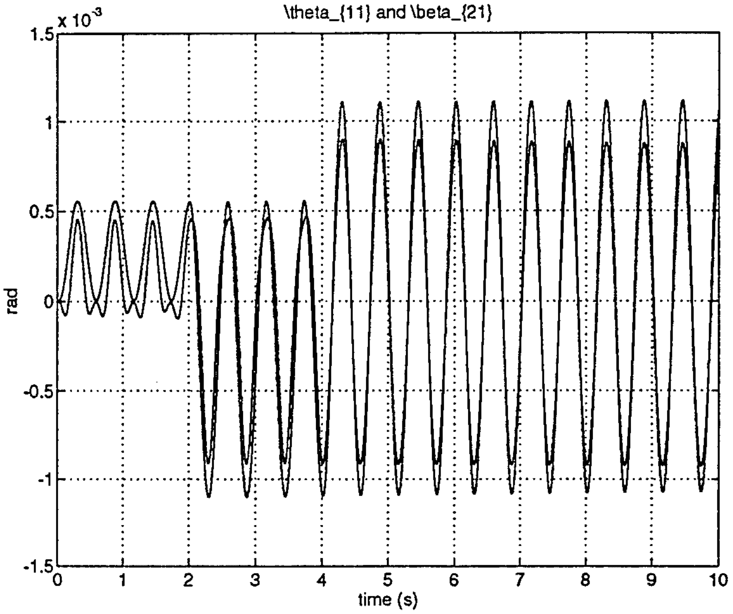

Figure 3.2: First-Order Displacements

is applied to both the zero-order problem and the first-order problem. The zero-order problem for the rigid-body maneuvering is nonlinear and of relatively low dimension and can be solved independently of the first-order problem. On the other hand, the zero-order solution induces time-varying coefficients and persistent disturbances in the first-order problem. The system of equations describing the first-order problem is linear, time-varying, and of high dimension. The equations are in terms of components about body axes, which makes them ideal to control design. The formulation lends itself to ready computer implementation.

As a numerical example, PACE test article is maneuvered according to a bang-bang control law by means of torques applied to the shoulder and the elbow. The numerical example shows the effectiveness of the algorithm developed in this report as well as the applicability of the equations to the maneuvering and control of flexible multibody systems.

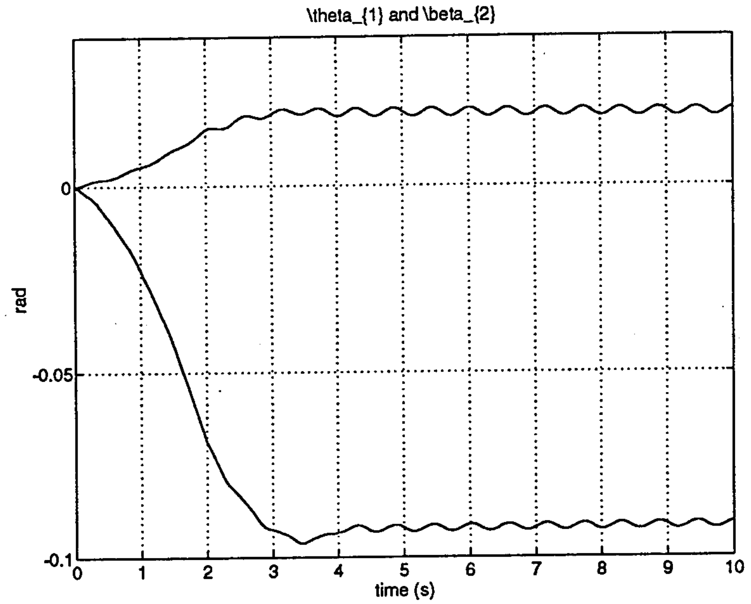

Figure 3.3: Time Histories of Angular Displacements

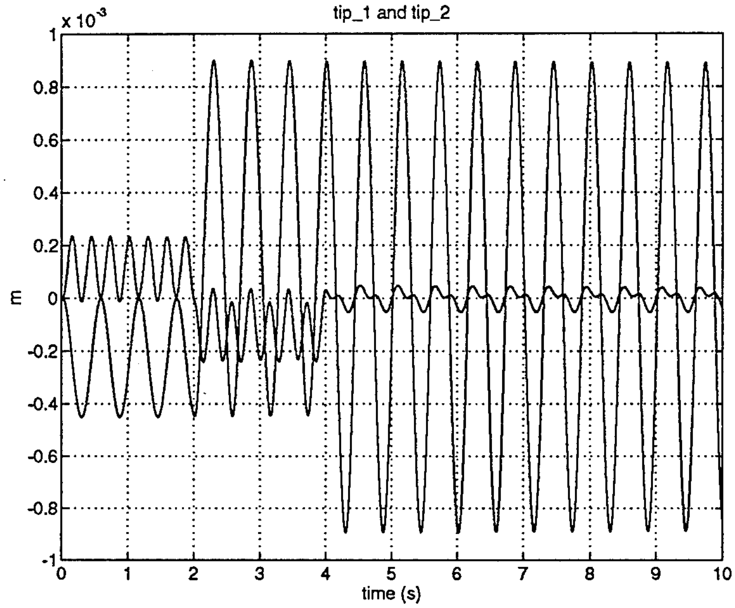

Figure 3.4: Time Histories of Tip Displacements

Chapter 4

Neural Network in Control

Contemporary control synthesis algorithms (e.g., robust, adaptive, and optimal controls) face severe limitations for some of the more challenging realistic systems as summarized in (Ref. 22). In particular, modern space structures are made of light-weight composites and equipped with piezoelectric or piezoceramic sensors and actuators. These flexible space structures, which are likely to be highly nonlinear with time-varying structural parameters and poorly modeled dynamics pose serious difficulties for all currently advocated methods. The difficulties arise in the characteristics of control system designs from a broad spectrum of aerospace applications; e.g., surveillance satellites, military robots or space vehicles. Existing design techniques often rely on the assumption of a good dynamic model containing identified system parameters. Furthermore, these iterative and computationally expensive approaches often require a prior fixed design constraints, where the loading and material properties need to be specified in advances. Consequently, design procedures to achieve the desired stability, robustness, and dynamic response for large space structures with unknown parameters are incomplete.

The ultimate autonomous control for flexible space structures, intended to maintain acceptable performance over an extended operating range without external intervention, can be especially difficult to achieve due to factors such as high dimensionality, multiple inputs and outputs, complex performance criteria, operational constraints, imperfect measurements, as well as the inevitable failures of various actuators, sensors, or other components. It is impractical to expect the existence of a high fidelity model prior to deployment of such complex structures. Therefore, the controller needs either to be exceptionally robust or adaptable after deployment. Also, catastrophic changes to the structural parameters, due to component failures, unpredictable uncertainties, and environmental influences, require that the controller be reconfigurable (i.e., it refers to the ability of reconfiguring the controller when the plant dynamics are subjected to catastrophic changes.).

Artificial neural networks, based on simplified models of the human brain, have been shown with promising capability for data manipulation. They have proven to provide an efficient means of learning concepts from past experience, abstracting features from uncorrelated data, and generalizing solutions from unforeseen inputs (Refs. 23-24). Unlike conventional digital computers which require storing program representing the processing steps need to be followed exactly, artificial neural networks are composed of highly interconnected processing elements which are analogous to the most elementary functions of the biological counterparts. It is also believed that such networks are extremely robust with respect to malfunctions of individual devices because of the distributed storage and information flow. They are in contrast to today's high speed Von-Neumann computers, which heavily rely on the perfect functioning of each device. In addition, these types of large-scale dynamic systems possess inherent potential for parallel computation because of assumed highly interconnected topology. Indeed, artificial neural networks have been successfully applied to identification and control of many dynamical systems, including autonomous space and underwater vehicles (Refs. 25-26), nuclear power plants (Ref. 27), chemical process facilities (Ref. 28), and manufacturing production lines (Ref. 29).

The ability of artificial neural networks to approximate arbitrary continuous functions (e.g., nonlinear plant dynamics and complex control law) provides an efficient mechanism for identification and control of large space structures. Back-propagation (BP) networks (Ref. 30) have recently been demonstrated to have the desired functional approximation capability with an arbitrary degree of accuracy (Ref. 31). Although back-propagation learning rule proves its effectiveness in many instances, it usually takes excessive time to train the neural networks and the networks may get trapped into local minima.

In this report, we utilize an efficient paradigm for identification and control of flexible space structures by incorporating the adaptive time-delay version of radial basis function (RBF) networks (Ref. 32). Radial basis function network with one hidden layer has been shown to be a universal approximator (Ref. 33). These model-free neural network paradigms (i.e., BP and RBF) are more effective in solving control problems than conventional learning control approaches, e.g., the BOXES algorithm which partitions the control law in the form of a look-up table (Ref. 34). Figure 4.1 shows the realizations of control laws via the BOXES approach (a staircase approximation) and a neural network approach (a smooth approximation).

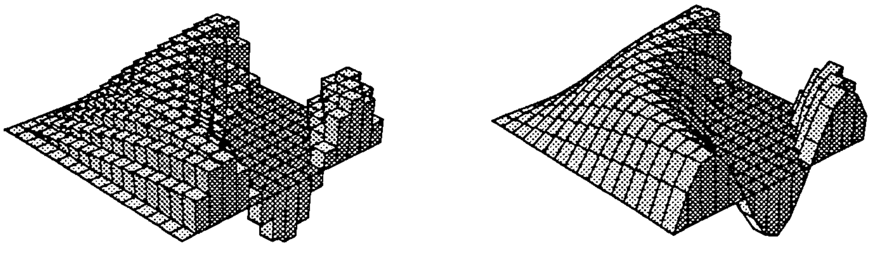

Figure 4.1: BOXES and Neural Network Approximations of Control Law

Our goal is to approach structural autonomy by extending the control system's operating envelope, which has traditionally required vast memory usage. Artificial neural networks, on the other hand, overcome common memory intensive problems and yet provide a sufficiently generalized solution space. In the vibration suppression problems, we utilize the adaptive time-delay radial basis function network as a building block to allow the control system to function as an indirect closed-loop controller. Prior to training the compensator, a neural identifier based on an ARMA model is utilized to identify the open-loop system. The horizon-of-one predictive controller regulates the dynamics of the nonlinear plant to follow a prespecified reference system asymptotically as depicted in Fig. 4.2 (i.e., model reference adaptive control architecture). As far as trajectory slewing problems are concerned, the generalized learning controller synthesized by the adaptive time-delay radial basis function network compensates the nonlinear large space structure in a closed-loop fashion as given in Fig. 4.3 (i.e., TDL: tapped delay line).

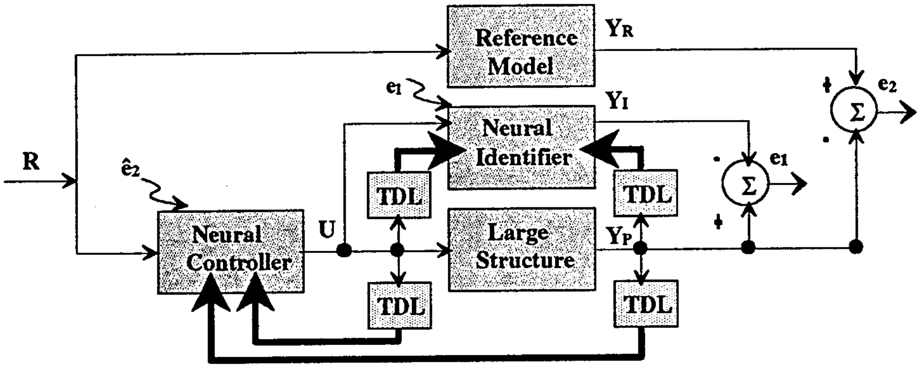

Figure 4.2: Vibration Suppression Learning Control Architecture

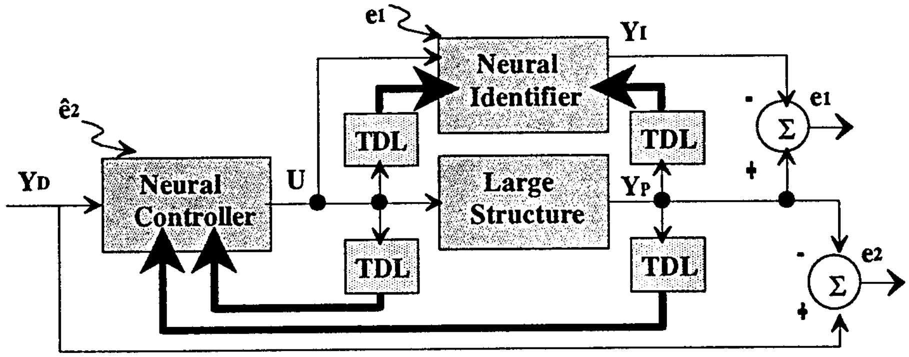

Figure 4.3: Trajectory Maneuvering Learning Control Architecture

The function of the neural controller is to map the states of the system into corresponding control actions in order to force the plant dynamics to match a certain output behavior which is specified either by the reference model (Y_R) or command output (Y_D). However, we cannot apply the optimization procedure (e.g., gradient descent or conjugate gradient) to adjust the weights of neural controller because the desired outputs for neural controller are not available. In (Ref. 35), a specialized learning algorithm which treats the plant as an additional unmodifiable layer of network is proposed. The output error, e_2, is back-propagated through the plant to derive the controller output error $\hat{e}_2$, and can be approximated by

$$\frac{\partial Y_{P,i}}{\partial u_j} \approx \frac{Y_{P,i}(\overline{u} + \delta u_j \hat{i}_j) - Y_{P,i}(\overline{u})}{\delta u_j}, \tag{4.1}$$

where $Y_{p,i}$ is the ith component of plant output, and u_j is the jth component of plant input. However, the authors fail to suggest a reliable way to compute $\hat{e}_2$. In (Ref. 36), the inverse Jacobian of the plant is used to estimate $\hat{e}_2$ at each weight update, which results in a complicated and computational expensive learning procedure. Moreover, since the plant is often not well-modeled because of modeling uncertainties, the exact partial derivatives cannot be determined. In (Ref. 37), a 'dynamic sign approximation' is utilized to determine the direction of the error surface, assuming the qualitative knowledge of the plant. This is not necessarily the case in space structure applications which are often equipped with highly correlated parameters. To achieve the true gradient descent of the square of the error, we use 'dynamic back propagation' (Ref. 38) to accurately approximate the required partial derivatives. A single-layer adaptive time-delay RBF network is first trained to identify the open-loop system. The resulting neural identifier then serves as extended unmodifiable layers to train the compensator (i.e., another single-layer adaptive time-delay RBF network). If the structural dynamics are to change as a function of time, the back-up neural identifier would require the learning algorithm to periodically update the network parameters accordingly.

In Section 4.1, back-propagation network is first outlined providing an underlined issue pertaining to the learning algorithm. Advanced from these derivations, the interconnecting topology and learning algorithm of radial basis function network are introduced in Section 4.2. This is followed by discussing an adaptive time-delay radial basis function network in Section 4.3 which provides an effective mechanism to capture most of the spatiotemporal interactions among the structural members. Simulation study of PACE test article is performed in Section 4.4 to demonstrate the feasibility and flexibility of the present architecture.

4.1 Back-Propagation Network

A given back-propagation network (Ref. 30) can be completely described by its interconnecting topology, neuronic characteristics, and learning rule. The individual processing unit performs its computations based only on local information. This biologically inspired mechanism provides the desired collective computational properties. Network with sigmoidal processing units has recently been demonstrated to have the desired functional approximation capabilities with an arbitrary degree of accuracy (Ref. 31). This implies that back-propagation network is dense on *C(D)*, where *C(D)* denotes the set of continuous functions in the closed and bounded set *D*. Based on this mathematical foundation, back-propagation networks have been successfully applied in various problems; from military signaturing to medical diagnosis; and from speech recognition and synthesis to robot and autonomous vehicle controls.

The output of the *i*th sigmoidal neuron in the *k*th layer of the network is defined by

$$u_i^k = \sum_{j=1}^{N^{k-1}} W_{ij}^k v_j^{k-1} + b_i^k, \quad i = 1,\ldots,N^k \tag{4.2a}$$

$$v_i^k = g_i^k(u_i^k), \quad i = 1,\ldots,N^k \tag{4.2b}$$

where g_i^k: $(-\infty,\infty) \rightarrow (-1,1)$ is a sigmoidal function (i.e., continuously differentiable, monotonically increasing, and *g(0)=0*), N^{k-1} is the number of neurons in the (*k*-1)st layer, W_{ij}^k is an adjustable parameter representing the interconnection weight between the output of the *j*th neuron of the (*k*-1)th layer to the input of the *i*th neuron of the *k*th layer and b_i^k is the external bias of the *i*th neuron of the *k*th layer. Figure 4.4 presents a functional diagram for a single neuron.

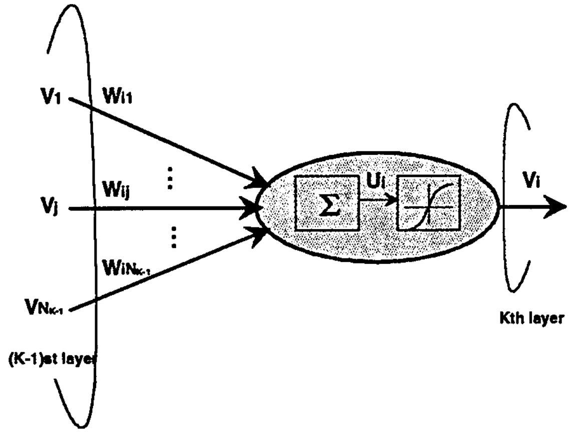

Figure 4.4: A Functional Diagram for a Single Neuron

The interconnection of these simple processing elements into a multilayer network given in Fig. 4.5 results in a general nonlinear mapping. Layer 1 is referred to as the input layer since the inputs directly effect these neurons. The last layer is referred to as the output layer, since the output of the network is the output of these neurons. For a single layered network, each processing unit is an output unit; thus if we know the desired output of the network we can calculate the output error for each processing unit. System dynamics for single layer perceptron network are given by

$$o_i = g_i(u_i) = g_i(\sum_{j=1}^{N^0} w_{ij} I_j + b_i), \quad i = 1, \ldots, N^1 \tag{4.3}$$

where I_j is the input to the network and N^0 is the number of input neurons. Assuming that our goal is to minimize the sum of the squares of the output errors described by $E = \frac{1}{2}\sum_{i=1}^{N^1}(d_i - o_i)^2$, where o_i is the actual output of the ith processing unit and d_i is the desired output of the ith processing unit, then gradient descents of E with respect to the interconnection weights, $w_{ij}, 1 \le i \le N^1, 1 \le j \le N^0$, and external biases $b_i, 1 < i < N^1$ are achieved by adapting each parameter according to equation (i.e., η is used for the learning rate.)

$$\Delta w_{ij} = -\eta(d_i - o_i)\frac{\partial g_i(u_i)}{\partial u_i} I_j. \tag{4.4a}$$

$$\Delta b_i = -\eta(d_i - o_i)\frac{\partial g_i(u_i)}{\partial u_i} \tag{4.4b}$$

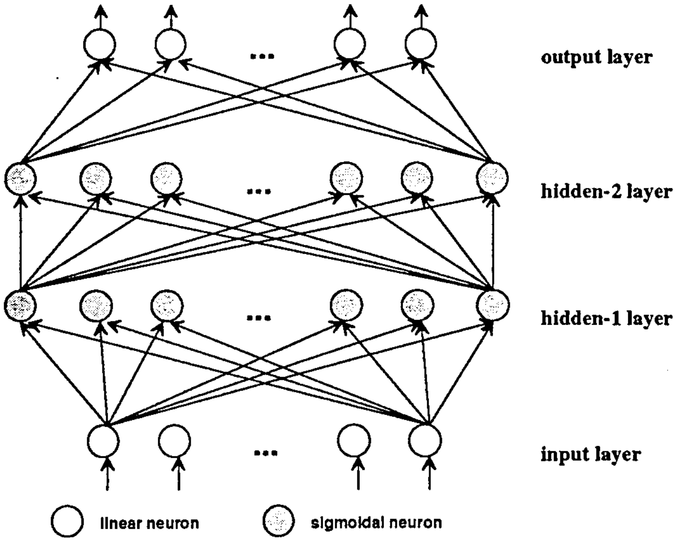

Figure 4.5: A Multi-layer Back-propagation Network

In a multilayer network, knowledge of a desired network output does not imply knowledge of the output error of each processing element within the network. The remaining layers of the network are referred to as hidden layers since the outputs of each neurons in these layers are hidden from the outputs of the network. The question is how to adapt the weights to the hidden layers when the desired output of the hidden layer is unknown. The solution, known as the back-propagation learning algorithm is outlined below.

As done for a single layer network we wish to adapt the interconnection weights to perform a gradient descent of the squared error at the output of the network. For processing unit in the output layer of the network, the weight adaptation equation is given by Eq. (4.4) with the network input I_j replaced by the output of the jth neuron of the previous layer v_j^{p-1},

$$\Delta w_{ij}^{p} = -\eta(d_i - o_i)\frac{\partial g_i^p(u_i^p)}{\partial u_i^p} v_j^{p-1} = \eta\,\delta_i^p\, v_j^{p-1}, \tag{4.5a}$$

$$\Delta b_i^p = -\eta(d_i - o_i)\frac{\partial g_i^p(u_i^p)}{\partial u_i^p} = \eta\delta_i^p \tag{4.5b}$$

where $\delta_i^p = -(d_i - o_i)\frac{\partial g_i^p(u_i^p)}{\partial u_i^p}$. The introduction of the δ parameter will simplify the presentation of the algorithm for the hidden layers. Each δ_i^k can be thought of as the output error backpropagated through the network to the input of the ith neuron of the kth layer (i.e., $\delta_i^k \equiv \frac{\partial E}{\partial u_i^k}$). Consider now a weight which does not connect to the output layer of the network, (i.e., w_{ij}^k, where $1 \le k < p$). In this case we will assume that $\delta_i^{k+1}$is known. From Eq. (4.5) this will be true for $k = p-1$. Using this information as a starting point, the algorithm presented below can be used to propagate the error, in an efficient manner, backwards through the network calculating the remaining δ_i^k's. The gradient of the square output error with respect to w_{ij}^k is described by

$$\Delta w_{ij}^k = \frac{\partial E}{\partial w_{ij}^k} = \eta\left(\sum_{s=1}^{N^{k+1}} \frac{\partial E}{\partial u_s^{k+1}}\frac{\partial u_s^{k+1}}{\partial u_i^k}\right)(\frac{\partial u_i^k}{\partial w_{ij}^k}), \tag{4.6a}$$

$$\Delta b_i^k = \frac{\partial E}{\partial b_i^k} = \eta\left(\sum_{s=1}^{N^{k+1}} \frac{\partial E}{\partial u_s^{k+1}}\frac{\partial u_s^{k+1}}{\partial u_i^k}\right)(\frac{\partial u_i^k}{\partial b_i^k}) \tag{4.6b}$$

where

$$\frac{\partial E}{\partial u_s^{k+1}} = \delta_s^{k+1}, \qquad \frac{\partial u_s^{k+1}}{\partial u_i^k} = \frac{\partial u_s^{k+1}}{\partial v_i^k}\frac{\partial v_i^k}{\partial u_i^k} = \frac{\partial g_i^k(u_i^k)}{\partial u_i^k} w_{si}^{k+1}, \tag{4.7a}$$

$$\frac{\partial u_i^k}{\partial w_{ij}^k} = v_j^{k-1}, \qquad \frac{\partial u_i^k}{\partial b_i^k} = 1 \tag{4.7b}$$

and N^{k+1} is the number of neurons in the (k+1)st layer of the network. Thus, the gradient calculation simplifies to

$$\Delta w_{ij}^k = \frac{\partial E}{\partial w_{ij}^k} = \eta \delta_i^k v_j^{k-1}, \tag{4.8a}$$

$$\Delta b_i^k = \frac{\partial E}{\partial b_i^k} = \eta \delta_i^k, \tag{4.8b}$$

$$\delta_i^k = \left(\sum_{s=1}^{N^{k+1}} \delta_s^{k+1} w_{si}^{k+1} \right) \frac{\partial g_i^k(u_i^k)}{\partial u_i^k}. \tag{4.9}$$

4.2 Gaussian Function Network

A radial basis function network (RBF) (Ref. 32) is a two layer neural network whose outputs form a linear combination of the basis functions derived from the hidden neurons. The basis function in the hidden layer produces a localized response to input stimulus as do locally-tuned receptive fields in our nervous systems (Ref. 39). Gaussian function network, a realization of RBF network using Gaussian kernels, is widely used in pattern classification and function approximation. The output of Gaussian neuron in the hidden layer is defined by

$$u_i^1 = \exp(-\frac{\left\| x - w_i^1 \right\|^2}{2\,\sigma_i^2}), \quad i = 1, \ldots, N^1 \tag{4.10}$$

where u_i^1 is the output of the ith neuron in the hidden layer, $x = \left[x_1, \ldots, x_{N^0} \right]^T$ is the input vector, $w_i^1 = \left[w_{i1}^1, \ldots, w_{iN^0}^1 \right]^T$ denotes the weighting vector for the ith neuron in the hidden layer (i.e., the center of the jth Gaussian kernel), σ_i^2 is the normalization parameter of the ith neuron (i.e., the width of the jth Gaussian kernel), and N^0 and

N^1 are the numbers of neurons in the input and hidden layers, respectively. Equation (4.10) produces a radially symmetric output with an unique maximum at the center and which drops off rapidly to zero at large radii. That is, it produces a significant nonzero response only when the input falls within a small localized region of the input space. The output layer equations are described by

$$y_i = w_i^{2^T} u^1, \quad i = 1,\ldots,N^2 \tag{4.11}$$

where y_i is the output of the ith neuron in the output layer, $w_i^2 = \left[w_{i1}^2,\ldots,w_{iN^1}^2\right]^T$ denotes the weighting vector for the ith neuron in the output layer, $u^1 = \left[u_1^1,\ldots,u_{N^1}^1\right]^T$ is the output vector from the hidden layer, and N^2 is the number of linear neurons in the output layer. The output layer neurons form a weighted linear combination of the outputs from the hidden layer.

Gaussian function networks have the property of undergoing local changes during training, unlike back-propagation networks which experience *global* weighting adjustments due to the characteristics of sigmoidal function. The localized influence of each Gaussian neuron allows the learning system to refine its functional approximation in a successive and efficient manner. Figure 4.6 provides a comparison of back-propagation network (at left) and RBF network (at right) in a pattern classification problem. Back-propagation network requires at least 4 hidden and 2 output sigmoid neurons, while Gaussian function network needs only one Gaussian hidden neurons and 2 output linear neurons to correctly classify two classes of patterns denoted by 'x' and 'o', respectively.

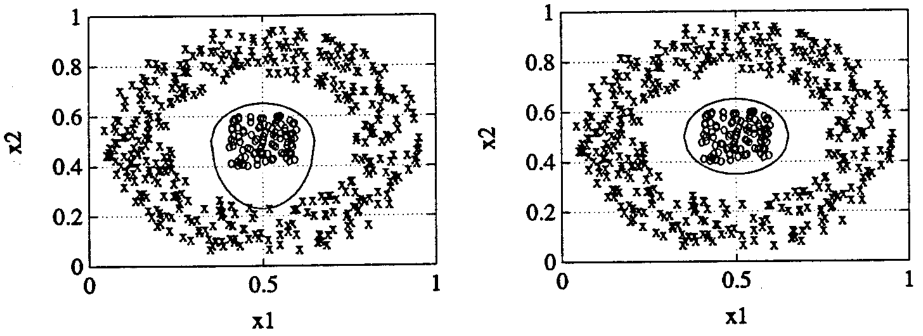

Figure 4.6: A Comparison of Pattern Classifications Problem.

The network effectively positions the Gaussian kernel at the center of the data, and then weights and thresholds it appropriately to produce the circular decision boundary. The hybrid learning algorithm (Ref. 32), which employs the K-means clustering for the hidden layer and the LMS for the output layer further ensures a faster convergence and often leads to better performance and generalization compared to back-propagation learning rule. The combination of locality of representation and linearity of learning offers tremendous computational efficiency in real-time adaptive control. In order to guarantee the generalization capability, a cross validation algorithm is used, and the number of training samples is ten times more than the theoretical VCdim bounds (Ref 40). K-means algorithm is perhaps the most widely known clustering algorithm because of its simplicity and its ability to produce good results. The normalization parameters, σ_i^2, are obtained once the clustering algorithm is complete. They represent a measure of the spread of the data associated with each cluster. The cluster widths are then determined by the average distance between the cluster centers and the training samples,

$$\sigma_i^2 = \frac{1}{m_i} \sum_{x \in \Theta_i} \left\| x - w_i^1 \right\|^2 , \quad i = 1, \ldots, N^1 \tag{4.12}$$

where Θ_i is the set of training patterns belonging to ith cluster and m_i is the number of samples in Θ_i. This is followed by applying a LMS algorithm to adapt the weights in output layer. The training set consists of input/output pairs but now the input patterns are processed by the hidden layer before being presented to the output layer.

4.3 Adaptive Time-delay RBF Network

Biological studies have shown that variable time-delays do occur along axons due to different conduction times and different lengths of axonal fibers. In addition, temporal properties such as temporal decays and integration occur frequently at synapses. Inspired by this observation, the time-delay back-propagation network was proposed in (Ref. 41) for solving the phoneme recognition problem. In this architecture, each neuron takes into account not only the current information from all neurons of the previous layer, but also a certain amount of past information from those neurons due to delay on the interconnections. However, a fixed amount of time-delay throughout the training process has limited the usage, possibly due to the mismatch of the temporal location in the input patterns. To overcome this limitation, Lin *et. al.* has developed an adaptive time-delay back-propagation network (Ref. 42) to better accommodate the varying temporal sequences, and to provide more flexibility for optimization tasks. In a similar spirit, the adaptive time-delay radial basis function network is proposed in this study to take full advantages of temporal pattern matching and learning/recalling speed.

A given adaptive time-delay radial basis function network can also be completely described by its interconnecting topology, neuronic characteristics, temporal delays, and learning rules. The individual processing unit performs its computations based only on local information. The output of the Gaussian neuron in the hidden layer is defined by

$$u_j^1 = \exp(-\frac{\left\| x - w_j^1 \right\|^2}{2\,\sigma_j^2}), \quad j=1,\ldots,N_1 \tag{4.13}$$

where u_j^1 is the output of the jth neuron in the hidden layer (denoted by the superscript 1), x is the input vector, w_j^1 denotes the weighting vector for the jth neuron in the hidden layer (i.e., the center of the jth Gaussian kernel), σ_j^2 is the normalization parameter of the jth neuron (i.e., the width of the jth Gaussian kernel), and N_1 is the number of neurons in the hidden layer. Equation (4.13) produces a radially symmetric output with a unique maximum at the center dropping off rapidly to zero for large radii. Inspired by the adaptive time-delay back-propagation network, the output equation of ATRBF networks is described by

$$y_j(t_n) = \sum_{i=1}^{N_1} \sum_{l=1}^{L_{ji}} w_{ji,l}^2 u_i^1\left(t_n - \tau_{ji,l}^2\right), \quad j=1,\ldots,N_2 \tag{4.14}$$

where $w_{ji,l}^2$ denotes the weight from the ith neuron in the hidden layer to the jth neuron in the output layer with the independent time-delay $\tau_{ji,l}^2$, $u_i^1(t_n - \tau_{ji,l}^2)$ is the output from the ith neuron in the hidden layer at time $t_n - \tau_{ji,l}^2$, L_{ji} denotes the number of delay connections between the ith neuron in the hidden layer and the jth neuron in the output layer. Shared with generic radial basis function networks, adaptive time-delay Gaussian function networks have the property of undergoing *local* changes during training, unlike adaptive time-delay back-propagation networks which experience *global* weighting adjustments due to the characteristics of sigmoidal functions. The localized influence of each Gaussian neuron allows the learning system to refine its functional approximation in a successive and efficient manner. The hybrid learning algorithm can also be used in the training of the adaptive time-delay radial basis function network. The adaptation of the output weights and time delays are derived based on error back-propagation to minimize the cost function,

$$E(t_n) = \frac{1}{2} \sum_{j=1}^{N_2} \left(d_j(t_n) - y_j(t_n)\right)^2, \tag{4.15}$$

where $d_j(t_n)$ indicates the desired value of the jth output neuron at time t_n. The weights and time-delays are updated step by step proportional to the opposite direction of the error gradient respectively,

$$\Delta w_{ji,l}^2 \equiv -\eta_1 \frac{\partial E(t_n)}{\partial w_{ji,l}^2}, \tag{4.16a}$$

$$\Delta \tau_{ji,l}^2 \equiv -\eta_2 \frac{\partial E(t_n)}{\partial \tau_{ji,l}^2}, \tag{4.16b}$$

where η_1 and η_2 are the learning rates. The mathematical derivation of this learning algorithm is straightforward. The learning rule can be summarized as follows.

$$\Delta w_{ji,l}^2 = \eta_1 \left(d_j(t_n) - y_j(t_n)\right) \sum_{l=1}^{L_{ji}} u_i^1\left(t_n - \tau_{ji,l}^2\right), \tag{4.17a}$$

$$\Delta \tau_{ji,l}^2 = -\eta_2 \left(d_j(t_n) - y_j(t_n)\right) \sum_{l=1}^{L_{ji}} w_{ji,l}^2 u_i^{1\prime}\left(t_n - \tau_{ji,l}^2\right). \tag{4.17b}$$

4.4 Simulation Study

The 5-second training data for system identification are generated directly from the modeling analysis in response to uniform random inputs. Due to the time limitation, at the current stage only the back-propagation network is used to validate the proposed architecture on this problem domain. Based on the PI's experience over the past two years, it is firmly believed that with adaptive time-delay radial basis function network, the network will converge at a faster rate, provided sufficient number of Gaussian neurons are used in the hidden layer. The motivation for exploring the back-propagation network in this study is to serve as a head start for a proper choice of the number of Gaussian neurons in the ATRBF network. The authors will continue to pursue in this issue.

Two sets of programs in MATLAB and gnu-C are developed for this study. They show different advantages in various aspects. The flowchart for C-simulation is given in Fig. 4.7. The dynamics of the plant are assumed unknown. System identification is simulated by a three layer back-propagation network. Two hidden layers with neurons 15 and 10, respectively are used to ensure the flexibility to approximate arbitrary non-convex regions. The output layer with linear activation function is used to scale-up to arbitrary range for the state variables spans. The control objective is to achieve trajectory slewing as well as vibration suppression along the motion.

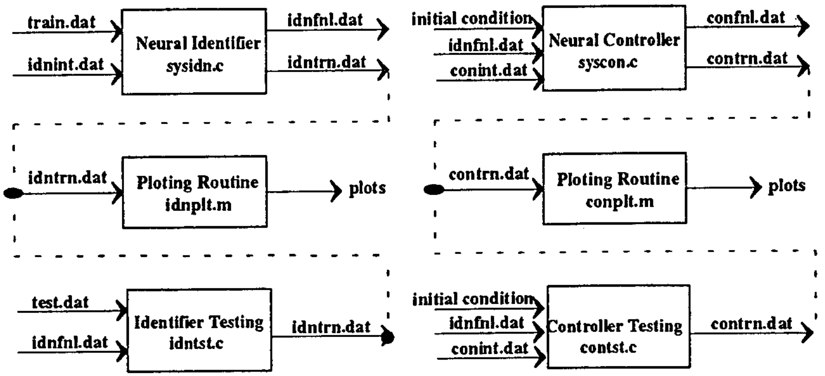

Figure 4.7: Simulation flowchart for gnu-C routines

Our strategy is to relax the vibration suppression task for the first few seconds. The RMS errors are accumulated only based on the trajectory following components, i.e., the first four components. Around 1.5 second, we begin to suppress the structural vibration for the last four components while maintaining the progress of trajectory following. However, by confining the desired responses to strictly zeros will deteriorate the network performance, even affecting the first four components. The way we proposed in this study is to setup an implicit exponential delaying envelop, so the network will smoothly catch up the requirements as we expected. Whenever the output falls within this exponential envelop, we will have the zero error, meaning no weight adjustments will be taken. This indeed shortens the training time significantly. Trajectory slewing/vibration suppression is performed by another back-propagation network with only one hidden layer (125 neurons). The output layer is equipped with hyperbolic tangent function for the activation function. The closed-loop controller regulates the dynamics of the PACE structure to follow the desired outputs as given below:

$$\theta_1(t)=\frac{\pi}{12}\left(1-\cos\frac{\pi t}{3}\right), \qquad \dot{\theta}_1(t)=\frac{\pi^2}{36}\sin\frac{\pi t}{3}, \qquad (4.18a)$$

$$\beta_2(t)=\frac{\pi}{24}\left(1-\cos\frac{\pi t}{3}\right), \qquad \dot{\beta}_2(t)=\frac{\pi^2}{72}\sin\frac{\pi t}{3}, \qquad (4.18b)$$

and tip displacements and their velocity are zeros, of course. Figure 4.8 presents the performance of the neural identifier with respect to all output variable respectively, in response to random inputs after training for 20000 trials. It took roughly a day to train it in a 50-MHz Pentium computer. Although the neural identifier learned to match the open-loop system in a reasonable time frame, the compensator took more than four days to converge to reasonable accuracy, mean square error 0.000562. Figure 4.9 displays the closed-loop performance with respect to all output variables, respectively, in response to an impulse. As shown in the figure, the neural regulator has learned to follow the specified trajectory and then damp out the structural vibration successfully.

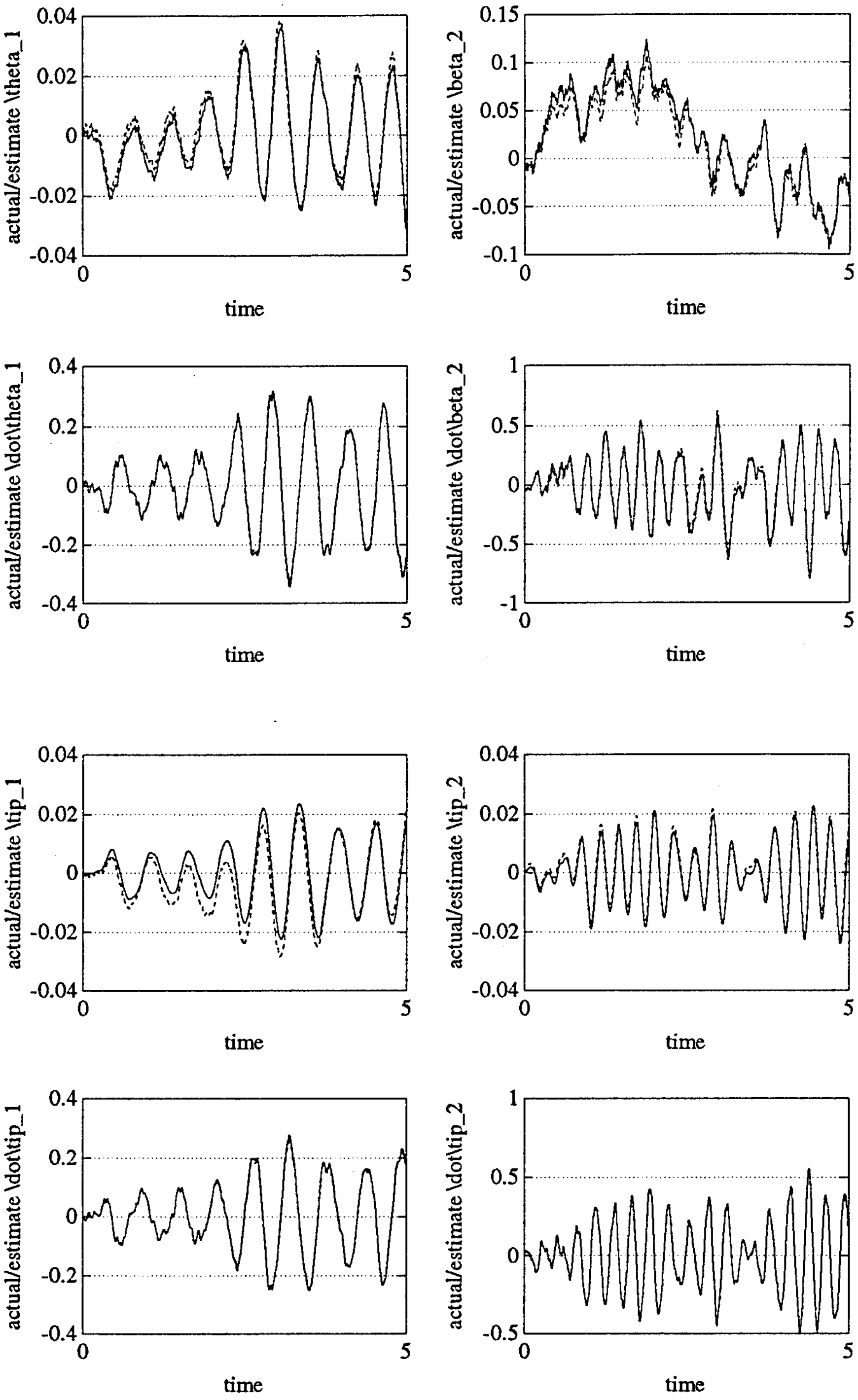

Figure 4.8: Open-loop Response of PACE test article

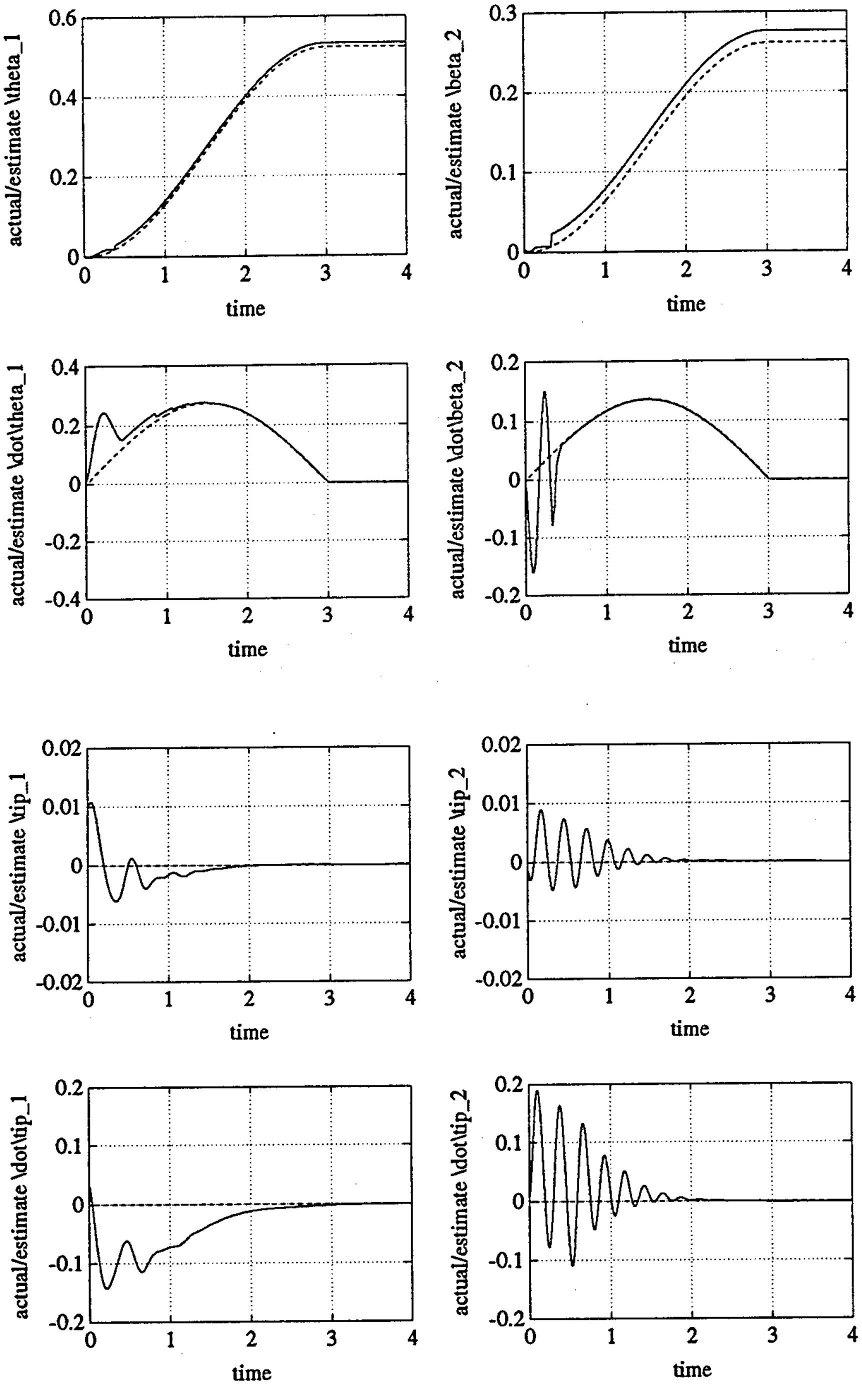

Figure 4.9: Closed-loop Response of PACE test article

Chapter 5

Conclusions

This paper is concerned with the development of a mathematical formulation capable of treating the problem of maneuvering and control of flexible multibody systems. The formulation is based on equations of motion in terms of quasi-coordinates derived for each substructure independently of the other substructures. The individual substructures are made to act as a single structure by means of a consistent kinematical synthesis. The net effect of this synthesis is the elimination of the redundant coordinates resulting from the original treatment of the substructures as if they were independent.

In most problems of interest, the elastic motions are small compared to the maneuvering motions. In recognition of this, a perturbation approach is developed whereby the rigid-body maneuvering of the system defines the zero-order problem and the elastic motions and perturbations from the rigid-body maneuvering define the first-order problem, where the term "order" refers to magnitude in a perturbation sense. The kinematical synthesis mentioned above is applied to both th zero-order problem and the first-order problem. The zero-order problem for the rigid-body maneuvering is nonlinear and of relatively low dimension, and it can be solved independently of the first-order problem. The control can be open loop or closed loop. On the other hand, the zero-order solution induces time-varying coefficients and persistent disturbances in the first-order problem. The system of state equations describing the first-order problem is linear, time-varying and of high dimension. The equations are in terms of component about body axes, which makes them ideally suited for control design. The control can be divided into two parts, one closed-loop part designed to control transient disturbances and another part designed to reject persistent disturbances. The formulation lends itself to ready computer implementation.

As a numerical example, a planar three-beam system is maneuvered according to a bang-bang control law by means of a torque applied to the center beam. Each beam is equipped with two actuators in charge of suppressing vibrations and perturbations in rigid-body motions; they exerted forces according to the direct velocity feedback control law. The numerical example shows the effectiveness of the algorithm developed in this paper as well as the applicability of the equations to the maneuvering and control of flexible multibody systems.

To support the above modeling efforts, an effective control algorithm needs to be design. We propose the use of a neural network-based control system for flexible multibody systems. The proposed neural controller is designed to achieve trajectory maneuvering of structural member as well as vibration suppression for precision pointing of space structures. To this end, a hybrid connectionist system is employed, which consists of a system identification neural network and a controller neural network. This leads to a robust and fault tolerant controller readily applicable to space structures of interest to the Air Force.

The proposed technology can directly be incorporated into various structure control designs for civilian and military satellites. In a similar spirit, the proposed architecture can be extended to dynamic controls of space engines, underwater vehicles, chemical processes, power plants, and manufacturing scheduling. The proposed technology will greatly reduce operational and developmental costs, and provide high performance, fault tolerant control systems in various application domains.

References

[1] M. K. Kwak and L. Meirovitch. A New Approach to the Maneuvering and Control of Flexible Multibody Systems. *Journal of Guidance, Control, and Dynamics*. **15(6)**. 1342-1353 (1992).

[2] M. K. Kwak, W. T. Schlaegel and A. Das. PACE: A Test Bed for the Dynamics and Control of Flexible Multibody Systems. *Proceedings of AIAA/ ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference*. 191-196 (1993).

[3] L. Meirovitch and A. L. Hale. Synthesis and Dynamic Characteristics of Large Structures with Rotating Substructures. *Proceedings of IUTAM Symposium on the Dynamics of Multibody System*. 231-244 (1978).

[4] J. M. Hollerbach. A Recursive Largrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity. *IEEE Transactions on Systems, Man and Cybernetics*. **10(11)**. 730-736 (1980).

[5] R. L. Houston. Flexibility Effects in Multibody System Dynamics. *Mechanics Research Communications*. **7(4)**. 261-268 (1980).

[6] R. L. Houston. Multi-body Dynamics Including the Effects of Flexibility and Compliance. *Computers and Structures*. **14(5)**. 443-451 (1981).

[7] W. J. Book. Recursive Largrangian Dynamics of Flexible Manipulator Arms. *International Journal of Robotics Research*. **3(3)**. 87-101 (1984).

[8] W. S. Yoo and E. J. Haug. Dynamics of Articulated Structures, Part I: Theory. *Journal of Structural Mechanics*. **14(1)**. 105-126 (1986).

[9] W. S. Yoo and E. J. Haug. Dynamics of Articulated Structures, Part II: Computer Implementation and Applications. *Journal of Structural Mechanics*. **14(2)**. 177-189 (1986).

[10] K. Changizi and A. A. Shabana. A Recursive Formulation for the Dynamic Analysis of Open Loop Deformable Multibody Systems. *Journal of Applied Mechanics*. **55**. 687-693 (1988).

[11] A. A. Shabana. On the Use of the Finite Element Method and Classical Approximation Techniques in the Non-linear Dynamics of Multibody Systems. *International Journal of Non-linear Mechanics*. **25(2)**. 153-162 (1990)

[12] L. Meirovitch. Hybrid State Equations for Flexible Bodies in terms of Quasi-Coordinates. *Journal of Guidance, Control and Dynamics*. **14(5)**. 1008-1013 (1991).

[13] L. Meirovitch and M. K. Kwak. State Equations for a Spacecraft with Maneuvering Flexible Appendages in terms of Quasi-Coordinates. *Applied Mechanics Reviews*. **42(11)**. 161-170 (1989).

[14] L. Meirovitch. *Computational Methods in Structural Dynamics*. Sijhoff & Noordhoff. Netherland. (1980).

[15] L. Meirovitch and M. K. Kwak. A Rayleigh-Ritz Based Substructure Synthesis Flexible Multi-Body Systems. *AIAA Journal*. **29(10)**. 1709-1719 (1991).

[16] L. Meirovitch and R. D. Quinn. Equations of Motion for Maneuvering Flexible Spacecraft. *Journal of Guidance, Control and Dynamics*. **10(5)**. 453-465 (1987).

[17] L. Meirovitch and R. D. Quinn. Maneuvering and Vibration control of Flexible Spacecraft. *Journal of Astronautical Science*. **35(3)**. 301-328 (1987).

[18] L. Meirovitch and M. K. Kwak. Dynamics and Control of a Spacecraft with Retargeting Flexible Antennas. *Journal of Guidance, Control, and Dynamics*. **13(2)**. 241-248 (1990).

[19] L. Meirovitch and M. K. Kwak. Control of a Spacecraft with Multi-Targeted Flexible Antennas. *Journal of the Astronautical Sciences*. **38(2)**. , 187-199 (1990).

[20] L. Meirovitch and M. K. Kwak. Control of Flexible Spacecraft with Time-Varying Configuration. *Journal of Guidance, Control, and Dynamics*. **15(2)**. 314-324 (1992).

[21] M. K. Kwak, K. K. Denoyer and D. Sciulli. Dynamics and Control of a Slewing Active Beam. *Proceedings of 9th VPI&SU Symposium on Dynamics and Control of Large Structures*. 191-194 (1993).

[22] D. A. White and D. A. Sofge. *Handbook of Intelligent Control- Neural, Fuzzy, and Adaptive Approaches*. Van Nostrand Reinhold. New York (1992).

[23] J. A. Hertz, A. S. Krogh and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley. Redwood City (1991).

[24] G. G. Yen. Learning, Forgetting, and Unlearning in Associative Memories: The Eigenstructure Method and The Pseudo Inverse Method. *Ph.D. dissertation*. Dept of Electrical and Computer Engineering. University of Notre Dame (1991).

[25] T. Troudet, S. Garg and W. C. Merrill. Neural Network Application to Aircraft Control System Design. *Proceedings of AIAA Guidance, Navigation, and Control Conference*. 993-1009 (1991).

[26] K. P. Venugopal, R. Sudhakar and A. S. Pandya. On-line Learning Control of Autonomous Underwater Vehicles Using Feedforward Neural Networks. *IEEE Journal of Oceanic Engineering*. **17(4)**. 308-319 (1992).

[27] Z. C. Guo and R. E. Uhrig. Using Modular Neural Networks to Monitor Accident Conditions in Nuclear Power Plants. *Proceedings of SPIE Technical Symposium on Intelligent Information Systems, Applications of Artificial Neural Networks III*. 505-516 (1992).

[28] K. Watanabe, I. Matsuura, M. Abe, M. Kubota and D. M. Himmelblau. Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks. *AIChE Journal*. **35(11)**. 1803-1812 (1989).

[29] C. S. Leem and S. E. Dreyfus. Learning Input Feature Selection for Sensor Fusion in Tool Wear Monitoring. *Proceedings of the Artificial Neural Networks in Engineering Conference*. 815-820 (1992).

[30] D. E. Rumelhart and J. L. McCelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. MIT. Cambridge (1986).

[31] K. Hornik, M. Stinchcombe and H. White. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*. **2(5)**. 359-366 (1989).

[32] J. Moody and C. J. Darken. Fast Learning in Networks of Locally-tuned Processing Units. *Neural Computation*. **1(2)**. 281-294 (1989).

[33] E. J. Hartman, J. D. Keeler and J. M. Kowalski. Layered Neural Networks with Gaussian Hidden Units as Universal Approximations. *Neural Computation*. **2(2)**. 210-215 (1990).

[34] D. Michie and R. A. Chambers. BOXES: An Experiment in Adaptive Control. in E. Dale and D. Michie editors. *Machine Intelligence*. 137-152 (1968).

[35] D. Psaltis, A. Sideris and A. A. Yamamura. A Multilayered Neural Network Controller. *IEEE Control Systems Magazine*. **8(3)**. 17-21 (1988).

[36] R. Elsey. A Learning Architecture for Control Based on Back-propagation Neural Network. *Proceedings of IEEE International Conference on Neural Networks*. 587-594 (1988).

[37] M. Saerens and A. Soquet. A Neural Controller. *Proceedings of IEE International Conference on Artificial Neural Networks*. 211-215 (1989).

[38] K. S. Narendra and K. Parthasarathy. Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks*. **1(1)**. 4-27 (1990).

[39] D. H. Cohen and S. M. Sherman. The Nervous System and its Components. in R. M. Berne and M. N. Levy editors. *Physiology*. 69-76 (1983).

[40] E. B. Baum and D. Haussler. What Size Net Gives Valid Generalization ?. *Neural Computation*. **1(1)**. 151-160 (1989).

[41] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang. Phoneme Recognition: Neural Networks Versus Hidden Markov Models. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. 107-110 (1988).

[42] D. T. Lin, J. E. Dayhoff and P. A. Ligomenides. Adaptive Time-delay Neural Network for Temporal Correlation and Prediction. *Proceedings of SPIE Conference on Biological, Neural Net, and 3-D Methods*. 170-181 (1992).