The last term of Eq. (32) is not considered because the vector state x is replaced by the estimated vector state $\hat{x}$. Thus, we have

$$\hat{x} = \begin{vmatrix} y \\ \dot{y} \\ a_{Mn} \\ a_{Tn} \end{vmatrix}; \quad F = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & -1/T & 0 \\ 0 & 0 & 0 & -1/\theta \end{vmatrix}; \quad G = \begin{vmatrix} 0 \\ 0 \\ 1/T \\ 0 \end{vmatrix} \tag{33}$$

The differential equation (22) for D gives

$$\begin{vmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_4 \\ \dot{d}_4 \end{vmatrix} = - \begin{vmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & -1/T & 0 \\ 0 & 1 & 0 & -1/\theta \end{vmatrix} \begin{vmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{vmatrix} \quad D_{tf}^T = [1,0,0,0] \tag{34}$$

The solution for $D(\tau)$ in this case may be written from Eq. (34) as

$$D^T = [1,\ \tau,\ T^2(1 - e^{-\tau/T}) - T\tau,\ \theta^2(e^{-\tau/\theta} - 1) + \theta\tau] \tag{35}$$

where τ is the time to go defined as

$$\tau = t_f - t \tag{36}$$

Using Eqs. (19) and (35), we may express P as

$$P = \begin{vmatrix} 1 & \tau & P_{13} & P_{14} \\ \tau & \tau^2 & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{31}^2 & P_{41}P_{31} \\ P_{41} & P_{42} & P_{41}P_{31} & P_{41}^2 \end{vmatrix} \tag{37}$$

where

$$\begin{aligned} P_{13} = P_{31} &= T^2(1 - e^{-\tau/T}) - T\tau \\ P_{23} = P_{32} &= \tau[T^2(1 - e^{-\tau/T}) - T\tau] \\ P_{14} = P_{41} &= \theta^2(e^{-\tau/\theta} - 1) + \theta\tau \\ P_{24} = P_{42} &= \tau[\theta^2(e^{-\tau/\theta} - 1) + \theta\tau] \end{aligned} \tag{38}$$

Using Eqs. (27) and (36) and

$$a = 1; \quad r = 3; \quad g_3 = \frac{1}{T}; \quad d_3 = T^2(1 - e^{-\tau/T}) - T\tau$$

we have

$$f(\tau) = \frac{1}{C_1} - \frac{1}{T^2}\int_\tau^0 [T^2(1 - e^{-\tau/T}) - T\tau]^2 \, d\tau \tag{39}$$

or

$$f(\mu) = \frac{1}{C_1}\left[1 + C_1T^3\left(\mu - \mu^2 + \frac{\mu^3}{3} - 2\mu e^{-\mu} + \frac{1 - e^{-2\mu}}{2}\right)\right] \tag{40}$$

where

$$\mu = \frac{\tau}{T} \tag{41}$$

The optimal control law from Eq. (29) is then

$$\begin{aligned} u = -\frac{1}{T^2}C(\mu)\{y + \tau\dot{y} + [T^2(1 - e^{-\tau/T}) - T\tau]a_{Mn} \\ + [\theta^2(e^{-\tau/\theta} - 1) + \theta\tau]a_{Tn}\} \end{aligned} \tag{42}$$

where

$$C(\mu) = \frac{1 - e^{-\mu} - \mu}{(1/C_1T^3) + \mu - \mu^2 + \tfrac{1}{3}\mu^3 - 2\mu e^{-\mu} + \tfrac{1}{2}(1 - e^{-2\mu})} \tag{43}$$

and the predicted miss distance becomes

$$\begin{aligned} M = y + \tau\dot{y} + [T^2(1 - e^{-\tau/T}) - T\tau]a_{Mn} \\ + [\theta^2(e^{-\tau/\theta} - 1) + \theta\tau]a_{Tn} \end{aligned} \tag{44}$$

Conclusions

This Note has presented the analytic derivation of the optimal closed-loop guidance law for a finite-bandwidth homing missile intercepting a maneuvering target and has provided a simple analytic means of solving the matrix Riccati differential equation. The resulting optimal guidance law agrees with the result determined by Hammond and includes the result obtained by Cottrell and Asher. The major contribution of this Note lies in the analytic solution of the matrix Riccati differential equation.

References

[1]Cottrell, R. G., "Optimal Intercept Guidance for Short Range Tactical Missiles," *AIAA Journal*, Vol. 9, No. 7, 1971, pp. 1414, 1415.

[2]Asher, R. B., "Optimal Guidance with Maneuvering Targets," *Journal of Spacecraft and Rockets*, Vol. 11, No. 3, 1974, pp. 204-206.

[3]Hammond, J. K., "Optimal and Suboptimal Laws for Simple Homing Systems," Aeronautics and Astronautics, Rept. 337, Southampton, England, UK, March 1975.

[4]Bryson, A. E., and Ho, Y. C., *Applied Optimal Control Theory*, Wiley, New York, 1975, Chap. 5.

Estimation of Modal Parameters of Linear Structural Systems Using Hopfield Neural Networks

Sabri Cetinkunt* and Hsin-Tan Chiu†
University of Illinois at Chicago, Chicago, Illinois 60680

I. Introduction

ARTIFICIAL neural networks (ANNs) can be considered highly interconnected dynamic systems wherein the main building block is a neuron. Two major types of artificial neural network architectures being investigated are feedforward- and recurrent-type network architectures. Previous research in feedforward-type ANNs show advantages in this type of network's ability to learn a control function.[1-4] Networks can be trained by a teacher that may be implementing a linear or nonlinear control algorithm. In essence, a feedforward-type neural network can be trained to approximate a hypersurface between the inputs and outputs. The use of ANNs for estimation purposes is rarely studied. This may be due to the fact that system dynamics cannot be explicitly retrieved from feedforward network connection strengths because the network learns only mapping functions. Recurrent-type neural network architectures do not suffer from this drawback. Recurrent-

Received April 24, 1992; revision received May 25, 1993; accepted for publication July 20, 1993.

*Associate Professor, Department of Mechanical Engineering.
†Graduate Student, Department of Mechanical Engineering.

type neural networks (i.e., Hopfield nets[5,6]) are capable of solving optimization problems. The characteristic distinction between feedforward- and recurrent-type networks is that the former learns the mapping function by updating its connection weights whereas the latter seeks its minimum computational energy as the state of neurons evolve in time. Recently, a neural estimator network architecture was reported in Ref. 7, where the Hopfield model is implemented to estimate the parameters of a linear system by expressing the network energy function in terms of the estimation error.

In this Note, the Hopfield model neural network design reported in Ref. 7 is modified to estimate the modal parameters of a flexible beam system without prior knowledge, assuming that the system states and their time derivatives are available. The potential applications of this research include the modal parameter estimation of large space structures, lightweight manipulators, and high-performance spacecraft.

II. Neural Network Model

A. Hopfield Network Model

The dynamic equations of the analog Hopfield network model are described as (see Fig. 1)

$$C_i \mathrm{d}U_i/\mathrm{d}t = \sum_{j=1}^{N} T_{ij}V_j - U_i/R_i + I_i$$
$$V_i = g(\lambda_i U_i) \tag{1}$$
$$U_i = (1/\lambda_i)g^{-1}(V_i) \qquad 1 \le i \le N$$

where N is the total number of neurons, and T_{ij} is the connection strength between neurons i and j, C_i is the ith neuron capacitance, and R_i, I_i are the ith neuron impedance and bias input, respectively. The input-output relation between the ith neuron state U_i and the output V_i is determined by the activation function $g(x)$, which is a nonlinear sigmoidal function. The dynamics of Eq. (1) are influenced both by the learning rate λ_i and the activation function $g(x)$. For instance, in the high learning rate limit case $\lambda \to \infty$, the activation function $g(x)$ approaches a step function.

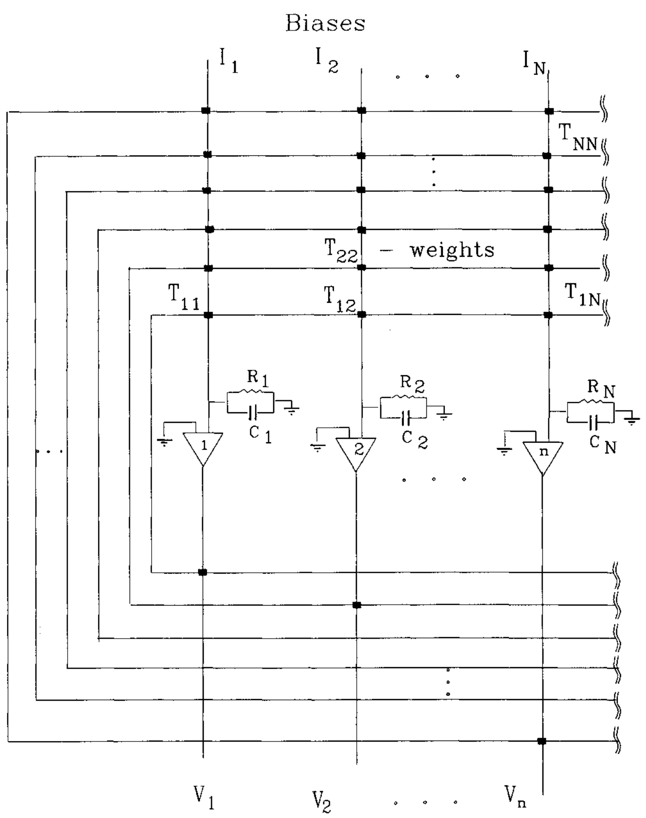

Fig. 1 Hopfield neural-network model.

Hopfield[5,6] showed that with either a synchronous or asynchronous update in the neuron states, the network retains the following function E,

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} I_iV_i + (1/\lambda)\sum_{i=1}^{N}(1/R_i)\int_0^{V_i} g^{-1}(V)\mathrm{d}V \tag{2}$$

E is also called the computational energy of the network. By using symmetric network connection strengths $T_{ij} = T_{ji}$ and $g(x)$ monotonically increasing sigmoid function, it was shown that the temporal evolution of V_i will change in such a way as to decrease the function E.

B. Neural-Network-Based Estimator Model

To represent the system parameters with neuron states, the concept of confining the state of neurons within unit hypercubes for an associative memory problem can be extended to large enough positive hypercubes G_i for each neuron. The G_i is defined as $G_i > 1$, $G_i \in \boldsymbol{R}^+$ for $1 \le i \le N$. By using the sigmoid activation function $g(x) = \tanh(x)$ for each neuron, the input-output relation of each neuron can be defined as

$$V_i = G_i \tanh(\lambda_i U_i) \tag{3}$$
$$U_i = -1/(2\lambda_i)\,\ell n \frac{(G_i - V_i)}{(G_i + V_i)} \qquad 1 \le i \le N$$

Here the output of each neuron V_i is constrained within the subset of positive hypercubes G_i. Therefore, for any parameter $\hat{p}$ in the system matrices, G_i must be chosen such that $|\hat{p}| < |G_i|$. Accordingly, with the input-output relation for each neuron defined by Eq. (3), the analog Hopfield network dynamics [Eq. (1)], when operated in high-impedance and unit capacitance conditions, can be described purely in terms of neuron output state variable V_i by substituting U_i in Eq. (3) into Eq. (1) as follows:

$$\frac{\mathrm{d}U_i}{\mathrm{d}t} = \left(\frac{\partial U_i}{\partial V_i}\right)\left(\frac{\mathrm{d}V_i}{\mathrm{d}t}\right) \tag{4}$$

where $\partial U_i/\partial V_i$ is given as

$$\left(\frac{\partial U_i}{\partial V_i}\right) = \frac{(G_i/\lambda_i)}{(G_i + V_i)\,(G_i - V_i)} \tag{5}$$

Hence, the dynamics equation of the resulting neural network can be formulated as

$$\dot{V} = K(TV + I) \tag{6}$$
$$K = \mathrm{diag}\,[k_i]$$
$$k_i = \left[\frac{\lambda_i (G_i + V_i)\,(G_i - V_i)}{(G_i)}\right] \qquad 1 \le i \le N$$

where N is the total number of neurons, V is a $N \times 1$ column vector containing the neuron outputs, and K is a diagonal matrix consisting of nonnegative diagonal elements. Comparing Eq. (6) to Eq. (1), the neural-network-based estimator architecture defined by Eq. (6) can be seen to have preserved the primary structure of the analog Hopfield model defined in Eq. (1).

Table 1 Initial states of the Hopfield neutral network used for modal parameter identification of a flexible beam

i	1	2	3	4	5	6	7	8	9	10	11	12
G_i	200	200	200	200	200	200	200	200	200	200	200	200
λ_i	100	100	100	100	100	100	100	100	100	100	100	100
V_i	0	0	–196	0	0	1	0	0	–9	0	0	1
R_i	100	100	100	100	100	100	100	100	100	100	100	100
C_i	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

In the case of a high-gain limit condition where $\lambda_i \to \infty$, the corresponding network computational energy E and its time derivative $\dot{E}$ are given by

$$E = -(1/2\, V^T T V + I V)$$

$$\frac{dE}{dt} = \left(\frac{\partial E}{\partial V}\right)\dot{V} = -(TV+I)^T K (TV+I) \le 0 \tag{7}$$

where $(\ )^T$ indicates the transposition of a matrix. The time derivative of the network computational energy $\dot{E}$ in Eq. (7) is shown to always be nonpositive, which means that temporal evolutions of the state of neurons V_i stop at an equilibrium point where

$$\frac{dE}{dt} \le 0, \text{ and for } \frac{dE}{dt} = 0 \to \frac{dV}{dt} = 0 \tag{8}$$

The Hopfield net-based estimator can be formulated such that the parameters of the net (interconnection weights T_{ij}) are determined by the state vector of the dynamic system, and the state vector of the net represents the parameters of the dynamic system. As the state of the network evolves in time and converges to a state that minimizes the energy function, the values of these converged-state vectors are the values of the estimated parameters. Details of the Hopfield network-based parameter estimator design are not given here due to space limitations. The reader is referred to Ref. 8 for details.

III. Simulation Results

Modal parameter estimation of a flexible beam is considered as an example. The flexible beam is actuated by a torque motor at the base with a torque profile $u(k)$ at the kth sampling time

$$u(k) = \sin(0.013347k) - \cos(0.0441k) * \cos(0.02455k) \tag{9}$$

The true modal parameters of the simulated beam model are given as follows:

$$I_H = 0.44$$

$$[\omega_0, \omega_1]^T = [0.0, 11.81]^T \text{ rad/s}$$

$$\left[\frac{dY_0(0)}{dx}, \frac{dY_1(0)}{dx}\right]^T = [1.0, 2.97]^T$$

$$[\xi_0, \xi_1]^T = [0.0, 0.02]^T$$

The neural-network-based estimator architecture simulated is shown in Fig. 2, and the initial state of each neuron is listed in Table 1. In simulating a high-gain limiting case with a sampling period of 1 ms, a large learning rate is used for each neuron as $\lambda_i = 100$. The state of neurons evolve within the prescribed hypercubes where the size of hypercubes are chosen as $G_i = 200.0$, for $i = 1, 2, \ldots, 12$. Moreover, zero initial state of neurons are assigned to most of the neurons except for those associated with the system inertia, and natural frequency where arbitrary positive initial values are as-

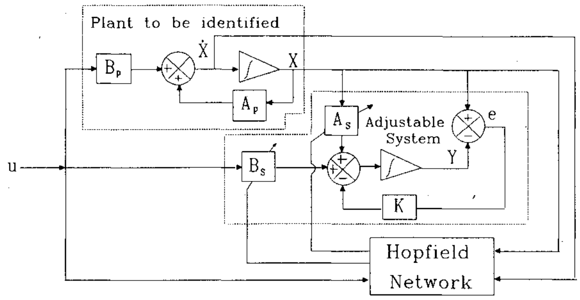

Fig. 2 Parameter identification using a Hopfield neural network.

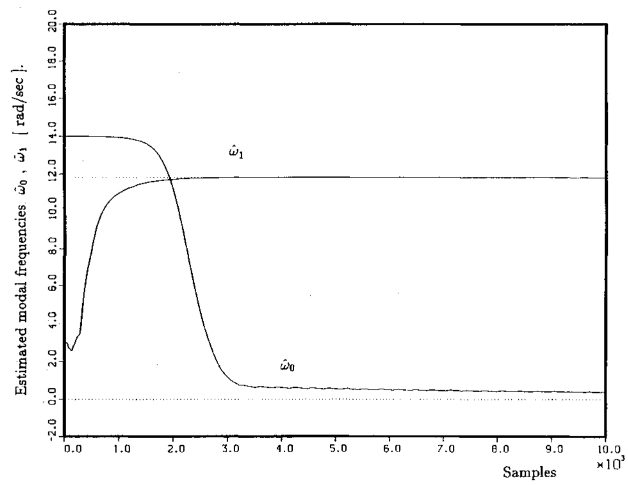

Fig. 3 Estimated modal frequencies $\hat{\omega}_0$, $\hat{\omega}_1$ as functions of the number of iterations.

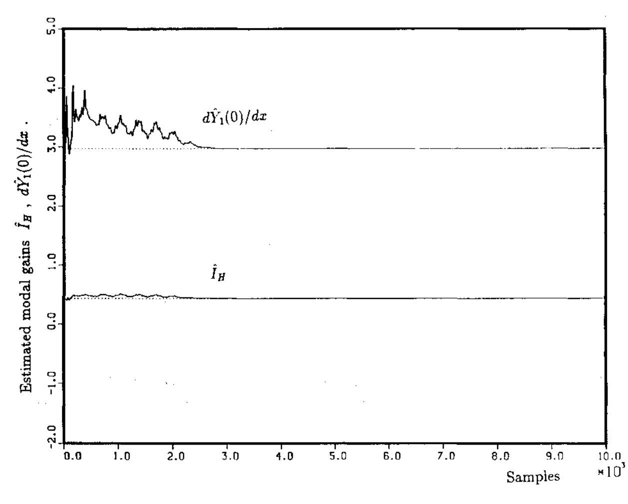

Fig. 4 Estimated modal gains $\hat{I}_H$, $d\hat{Y}_1(0)/dx$ as functions of the number of iterations.

signed in complying with a physical beam model. In the simulations presented here, we assumed that all states and their first time derivatives were available through an appropriate measurement system. This is the same assumption made in Ref. 7.

The simulation result depicted in Fig. 3 shows the estimated natural frequency of the system. It is worth noting that the initial frequencies were intentionally given an unreasonable magnitude order $[\hat{\omega}_0, \hat{\omega}_1]^T = [14.0, 3.0]^T$ rad/s such that the flexural frequency is much lower than that of the rigid mode. For this particular example, the neural estimator demonstrates the capability to learn the frequency spectrum of the system accurately within 3000 iterations. Moreover, the learning time could be reduced if more accurate initial guess values were provided. The learning rate may vary significantly from one application to another depending on the complexity of the problem. The simulation result in Fig. 4 shows the estimation of system inertia $\hat{I}_H$ and flexural modal gain $d\hat{Y}_1(0)/dx$ where unitary initial values are assigned to both parameters without loss of generality. It is observed that both parameters converge to true values much faster than the flexural frequency estimation. The estimated modal damping $\hat{\xi}_0, \hat{\xi}_1$ also converged to the correct values within about the same number of iterations.

IV. Conclusions

In this paper, a neural-network-based estimator is used for the modal parameter estimation of a flexible beam. The neural estimator is designed based on a modified form of the Hopfield network architecture. Simulation results demonstrate the asymptotic convergence in the state of neurons to the modal parameters of a case study without prior knowledge, assuming that the system states and its time derivatives are available. The proposed estimator has the following advantages over the traditional parameter estimators such as recursive least squares: the network model, and therefore the estimator itself, is a nonlinear dynamic system. It has the potential to handle the parameter estimation problem of nonlinear systems. With the analog circuit implementation of Hopfield nets, it can be implemented in real time and be applied to highly nonlinear systems.

Acknowledgment

This material is based on work supported by the National Science Foundation under Grant MSS-9024266, and the U.S. Air Force Office of Scientific Research under Grant AFSOR-90-J-0490.

References

[1]Selinsky, J., and Guez, A., "A Trainable Neuromorphic Controller," *Journal Robotic System*, Vol. 5, No. 4, Aug. 1989, pp. 363–388.

[2]Narendra, K. S., and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4–26.

[3]Rumelhart, D. E., and McClelland, J., "Learning Internal Representation by Error Back Propagation," *Parallel Distributed Processing*, Massachusetts Inst. of Technology Press, Cambridge, MA, 1986, pp. 319–362.

[4]Cetinkunt, S., and Chiu, H. T., "A Study of Learning Controllers for Tip Position Control of a Flexible Arm Using Artificial Neural Networks," *ASME Winter Annual Meeting* (Atlanta, GA), DSC-Vol. 31, American Society of Mechanical Engineers, New York, 1991, pp. 15–19.

[5]Hopfield, J. J., and Tank, D. W., "Neural Computation of Decision in Optimization Problems," *Biological Cybernatics*, Springer-Verlag, Vol. 52, 1985, pp. 141–152.

[6]Tank, D. W., and Hopfield, J. J., "Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit," *IEEE Transactions on Circuits Systems*, Vol. CAS-33, No. 5, 1986, pp. 533–541.

[7]Shoureshi, R., Chu, R., and Tenorio, M., "Neural Networks for System Identification," *Proceedings of American Control Conference*, Vol. 1, June 1989, pp. 916–921.

[8]Chiu, H. T., "Recurrent Neural Network Based Dynamic System Identification and Control," Ph.D. Thesis, Dept. of Mechanical Engineering, Univ. of Illinois at Chicago, Chicago, IL, 1993, pp. 49–91.

Evaluation of Inertial Integrals for Multibody Dynamics

Achille Messac* and Joel Storch†
Charles Stark Draper Laboratory, Cambridge, Massachusetts 02139

Introduction

WHEN a continuum representation is employed in the development of multibody dynamics equations (e.g., Ref. 1), a class of inertial integrals must be evaluated to enable numerical simulation of the dynamical system. For all but the simplest cases, direct and explicit evaluation of these integrals is impractical; some form of approximation is generally employed. We herein present an approach for the evaluation of the needed integrals in terms of readily available finite element derived quantities. Because of our particular choice of modes, the integral evaluation requires no further approximations than those inherent to the finite element analysis and the model reduction. The inertial representation of the individual bodies may involve either lumped or consistent mass matrices.

The approach developed involves evaluating the kinetic energy of a vibrating body via two distinct methods, followed by a transformation to a common set of independent generalized velocities. Expressions for the required inertial integrals follow from equating the matrices of the respective quadratic forms.

Displacement Field Representation

System Kinematics

Consider a generic flexible body whose kinematics are defined with the aid of two frames: an inertial frame $\boldsymbol{F}_n$ and an embedded-body frame $\boldsymbol{F}_b$. In the present context, all flexible and rigid-body translations/rotations are assumed to be small. This assumption is an expedient to the integral evaluation. As will be seen, the end result is applicable to both large and small displacements of a flexible body. The absolute displacement (not position) of a generic point of the flexible body is expressed in the form

$$d(r, t) = [I^3 \tilde{r}^T]\, U_o(t) + u(r, t), \quad U_o(t) = \begin{Bmatrix} u_o(t) \\ \theta_o(t) \end{Bmatrix} \tag{1}$$

where

$$\tilde{r} \equiv \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}; \quad r \equiv \begin{Bmatrix} r_x \\ r_y \\ r_z \end{Bmatrix} \tag{2}$$

Here and in the sequel, for any integer p, I^p denotes the p-dimentional identity matrix.

The vector u_o denotes the translation of the origin of $\boldsymbol{F}_b$ relative to that of $\boldsymbol{F}_n$, r denotes the location of a generic point before deformation relative to the origin of $\boldsymbol{F}_b$, and $u(r, t)$ represents the deformation field. Since the body frame is embedded, the boundary conditions are $u(0, t) = 0$ and curl $u(0, t) = 0$. The triplet θ_O denotes the successive small rotations of the three axes of $\boldsymbol{F}_b$ relative to those of $\boldsymbol{F}_n$. All quantities are expressed in the body frame.

Received June 20, 1992; revision received June 14, 1993; accepted for publication June 23, 1993.

*Senior Member, Technical Staff, 555 Technology Square. Senior Member AIAA.

†Principal Member, Technical Staff, 555 Technology Square. Senior Member AIAA.