The interval set denoted by the brackets is an interval set of matrices where B_l is a matrix of lower bounds on the elements of $B(x)$ and B_u is a matrix of upper bounds on the elements of $B(x)$.

Using the basic properties of interval sets[3] we see that

$$\frac{\partial v}{\partial x}B(x)x \in \left[\frac{\partial v}{\partial x}B_l x,\ \frac{\partial v}{\partial x}B_u x\right] \tag{14}$$

$$\left[\frac{\partial v}{\partial x}B_l x,\ \frac{\partial v}{\partial x}B_u x\right] \triangleq [w_l(x),\ w_u(x)] \tag{15}$$

with

$$w_l(x) < 0 \tag{16}$$

$$w_u(x) < 0 \tag{17}$$

so that

$$[w_l(x),\ w_u(x)] < 0 \tag{18}$$

and the equivalent linear system with the structured uncertainties given by the interval matrices $[B_l, B_u]$ will always be stable.

At this point we have proven that if the original equations are asymptotically stable, then the resulting structured uncertainty equations will be asymptotically stable. However, can the structured uncertainty linear equations indicate stability if the original equations are not stable? We can use the Lyapunov parallel theorems[2] to show how the structured uncertainty linear equations will not be stable if the original nonlinear equations are not stable and thereby answer the preceding question in the negative. This is done by repeating the above proof with again a positive-definite Lyapunov function $v(x)$ but with $w(x)$ being positive definite,

$$w(x) > 0 \tag{19}$$

for the original unstable equations instead of negative definite.

Limitation of Proof

The proof made use of the assumption of existence of a positive-definite autonomous function $v(x)$. It is by no means guaranteed that such an autonomous function exists, although existence of the necessary positive-definite but nonautonomous function can be shown to exist [i.e., existence of the required $v(x, t)$ with v dependent on time as well as on x can be proven[2]].

Example

An example illustrating the usefulness of the technique is as follows.

Gibson[4] presents the example of a set of nonlinear equations

$$\dot{x}_1 = x_2 \tag{20}$$

$$\dot{x}_2 = -x_1^3 - x_2 \tag{21}$$

whose asymptotic stability can be proven with the aid of a fairly complicated variable gradient method for generating Lyapunov functions. The nonobvious Lyapunov function arrived at is

$$v = \tfrac{1}{2}x_1^4 + \tfrac{1}{2}x_1^2 + x_1x_2 + x_2^2 \tag{22}$$

with the resulting negative-definite function $w(x)$ being

$$w = -x_1^4 - x_2^2 < 0, \qquad x_1, x_2 \neq 0 \tag{23}$$

Using our technique, we make the time scale change

$$\frac{d\tau}{dt} = 1 + x_1^2 \tag{24}$$

so that

$$\frac{dx_1}{d\tau} = \alpha x_2 \tag{25}$$

$$\frac{dx_2}{d\tau} = -(1-\alpha)x_1 - \alpha x_2 \tag{26}$$

where

$$\alpha = 1/\left(1 + x_1^2\right) \tag{27}$$

$$0 < \alpha < 1 \tag{28}$$

The characteristic equation for (25) and (26) is

$$s^2 + \alpha s + (1-\alpha)s = 0 \tag{29}$$

which shows asymptotic stability[1] for the conditions of Eq. (28).

Conclusions

A proposition relating nonlinear stability analysis to stability theory for linear differential equations having structured uncertainties was proved and illustrated by an example. The stability analysis of nonlinear differential equations is greatly simplified by applying the proposition because the necessity of guessing a Lyapunov function that works for each particular case can be eliminated, as illustrated by the example.

The proof that was presented strengthens confidence in the use of the resulting technique for solving applicable problems, albeit a limitation of the proof is the assumption of existence of an autonomous Lyapunov function for the nonlinear differential equations.

References

[1]Misra, P., "On Stabilization of Systems with Uncertain Parameters: An Interval Arithmetic Approach," 1989 American Control Conference, Vol. 1, June 1989, pp. 657–658.

[2]Hahn, W., *Stability of Motion*, Springer-Verlag, 1967.

[3]Neumaier, A., *Interval Methods for Systems of Equations*, Cambridge Univ. Press, 1990.

[4]Gibson, J. E., *Nonlinear Automatic Control*, McGraw-Hill, New York, 1963.

Multiple Optimal Solutions for Structural Control Using Genetic Algorithms with Niching

K. KrishnaKumar,* R. Swaminathan,† and L. Montgomery†
University of Alabama, Tuscaloosa, Alabama 35487

Introduction

THE interaction between the placement of actuators and sensors and the optimal control system synthesis poses a unique challenge to the problem of optimal structural control. Studies conducted in the past have generally examined the structural control problem to consist of two separate problems: 1) placement of actuators/sensors to minimize some criteria and 2) syntheses of feedback control systems to suppress structural vibration, station keeping, attitude control, etc. For example, Refs. 1–3 use a linear quadratic regulator (LQR) technique to synthesize a feedback controller and then use the resulting control system for the computation of a performance index (PI for actuator/sensor placement) based on control effort. The resulting strategy, to a large extent, depends on the characteristics of the LQR. In Ref. 4, pole-positioning techniques are used with an optimal energy formulation. Here also, first the closed-loop poles are chosen and then the actuators are placed to minimize the

Received Feb. 12, 1993; revision received Nov. 5, 1993; accepted for publication Nov. 5, 1993.

*Assistant Professor of Aerospace Engineering, Department of Aerospace Engineering. Senior Member AIAA.

†Graduate Student of Aerospace Engineering, Department of Aerospace Engineering.

control effort. In Ref. 5, feedback gains were fixed a priori before the actuators/sensors are optimally placed.

In this Note, two unique ideas in optimal structural control are presented and are implemented using a genetic algorithm optimization technique with niching. First, we treat a structural control problem as a simultaneous actuators/sensors placement and feedback controller gain optimization problem. We use a genetic algorithm as a parameter optimization technique to optimize for the locations of actuators and sensors and the feedback gains. Second, we examine the feasibility of finding more than one "distinctly different" near-optimal solutions. Here, the multiple near-optimal solutions define the solutions near local optima. The motivation behind this is simple: The designer is given more than one optimal choice for the placement of actuators and sensors. To carry out this idea, we induce niche formation in a genetic algorithm using a sharing mechanism. The end result is the convergence of the genetic algorithm population to multiple locally near-optimal solutions. We use a simple clamped-free beam to demonstrate the above concepts.

Niching in Genetic Algorithms

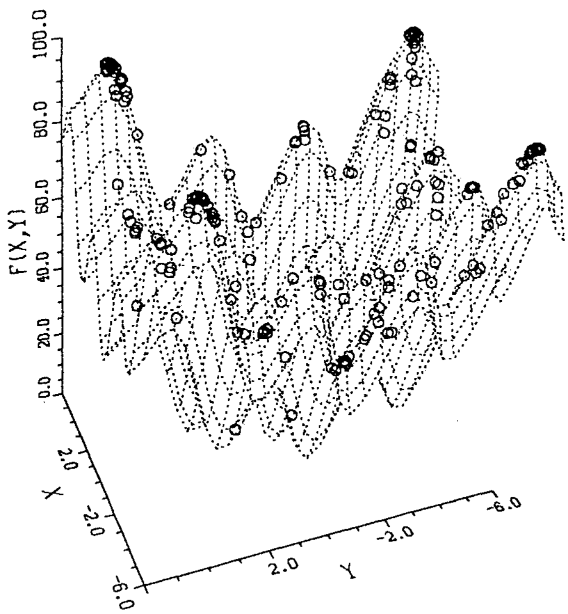
Before niching in genetic algorithms is defined, we present the mechanics of a simple genetic algorithm. In the simplest sense, the implementation of a genetic algorithm (GA) can be viewed as finding an optimal set of parameters based upon the concept of "survival of the fittest." The GA uses a process that resembles natural genetics. It requires no derivative information, and its search process is inherently global in nature. A brief overview of the operation of a genetic algorithm follows. Detailed presentations are given in Refs. 6 and 7.

In a simple GA,[7] the parameters chosen for optimization are coded as a binary string. In actuator/sensor placement and control gain optimization, strings represent actuator/sensor locations and the gains. For example, if actuator locations for a clamped-free beam were to be discretely determined using a five-bit binary string, the string [00000] would represent a position at the base of the beam ($x/L = 0$) and the string [11111] would give a position at the tip of the beam ($x/L = 1$). Since there are $2^5 - 1$, or 31, total choices for actuator locations, there would be 29 other possible locations. Initially, a number of strings are generated randomly to create what is termed a "population."

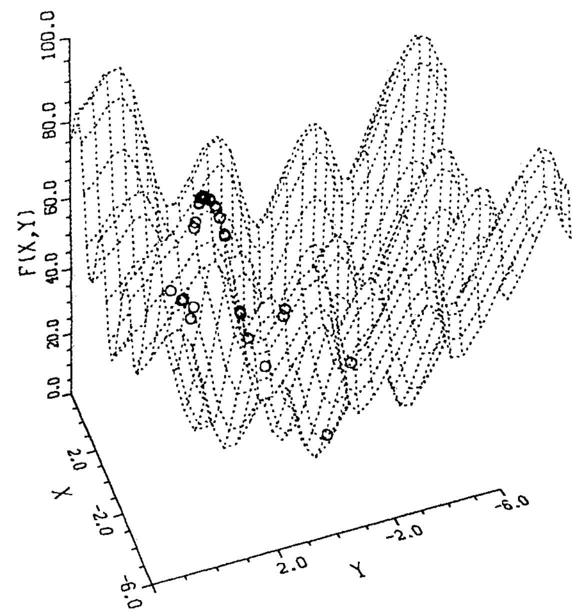
Optimal strings are found through population reproduction via "selection," "crossover," and "mutation." Selection is a process where an old string is carried through into a new population depending on its PI (i.e., fitness) value. Due to this move, strings with above-average fitness values get larger numbers of copies in the next generation. This strategy, in which good strings get more copies in the next generation, emphasizes the survival-of-the-fittest concept of genetic algorithms. There are many equivalent selection techniques,[6,7] with the most popular being the tournament selection technique.

A simple crossover follows selection in three steps. First, the newly selected strings are paired together at random. Second, an integer position n along every pair of strings is selected uniformly at random. Finally, based on a probability of crossover, the paired strings undergo crossover at the integer position n along the string. This results in new pairs of strings that are created by swapping all the characters between characters 1 and n inclusive. Mutation is simply an occasional random alteration of a string position (based on probability of mutation). In a binary code, this involves changing a 1 to a 0 and vice versa. The mutation operator helps in avoiding the possibility of mistaking a local minimum for a global minimum. When mutation is used sparingly (about one mutation per thousand bit transfers) with selection and crossover, it improves the global nature of the genetic algorithm search.

For many optimization problems there may be multiple, equal or unequal, optimal solutions. A simple GA cannot maintain stable populations at different optima of such functions.[8] In case of optimal solutions with equal fitness, sampling errors in GA operators cause the population to converge to a single solution. Whereas in the case of unequal optimal solutions, the population invariably converges to the global optimum. Figure 1 presents a two-parameter search space example with four equal global optima and many local optima. A



a) With niching



b) Without niching

Fig. 1 Distribution of population members with and without niching.

simple GA with no niching converges to a single optimum. However, a modification of the GA process with niching helps in maintaining subpopulations near global and local optima.

The availability of alternate solutions is of great practical utility. In the structural control problem, such solutions would give the designer the much desired flexibility regarding the actuator/sensor positioning. This is especially true when certain practical considerations forbid the positioning of actuators and sensors at GA-suggested optimal locations. To achieve this objective, it is essential to introduce a controlled competition among different solutions near every locally optimal region. This would maintain stable subpopulations at such optimal regions. This could be accomplished by incorporating the concepts of niche and species into the GA search process.[7,8]

A niche is viewed as an organism's environment, and species is a collection of organisms with similar features. The subdivision of environment based on an organism's role reduces interspecies competition for environmental resources, and this reduction in competition helps stable subpopulations to form around different niches in the environment. In GA terms, the organism is analogous to an individual member (string) of the population, and the environment is analogous to the fitness function.

Sharing functions are one way of inducing nichelike behavior in genetic search.[8] The concept of sharing is carried out by degrading an individual's fitness proportional to the number of other members in its neighborhood. The amount of sharing contributed by each individual into its neighbor depends on the proximity between the two, and the closer the individuals are, the more degradation there is. Mathematically, a metric d_{ij} can be introduced over the ith and jth individuals of the population to measure the closeness in solution space. A sharing function $s(d_{ij})$ may be defined as[8]

$$s(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_{\text{share}}) & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and the shared fitness of the ith individual is given as

$$\text{Shared fitness} = \frac{\text{True fitness}}{\sum_j s(d_{ij})} \tag{2}$$

In the equation presented above, σ_{share} is the limiting distance between the individuals to be shared. If the parameter space ranges from $x_{\min}$ to $x_{\max}$ and the peaks are uniformly spaced, σ_{share} may be calculated as the average distance required to identify each niche distinctly in the solution space. For a single-parameter function, d_{ij} may be computed as the absolute difference $|x_i - x_j|$. It is shown in Ref. 8 that for a problem with p parameters of unequal boundaries and q assumed peaks, the equivalent normalized distance measure d_{ij} can be used such that

$$d_{ij} = \sqrt{\sum_{k=1}^{p} \left(\frac{x_{k,i} - x_{k,j}}{x_{k,\max} - x_{k,\min}} \right)^2} \tag{3}$$

where

$x_{k,i}$ = kth parameter of individual i
$x_{k,j}$ = kth parameter of individual j
$x_{k,\max}$ = maximum allowable value for kth parameter
$x_{k,\min}$ = minimum allowable value for kth parameter

Also, an estimate of σ_{share} for this normalized metric is given as[8] $\sigma_{\text{share}} = 0.5q^{(-1/p)}$.

Structural Control Problem

To show the feasibility of obtaining multiple optimal structural control solutions using a genetic algorithm with niching, a simple clamped-free beam is chosen. The beam properties used for the study are shown in Table 1.

For modeling the structure, the first five natural modes of the structure are used. The state-space equations, including the closed-loop feedback control, are given as[9]

$$\dot{q}(t) = \begin{bmatrix} 0 & I \\ -M^{-1}K_S & 0 \end{bmatrix} q(t) + \begin{bmatrix} 0 \\ M^{-1}\Phi_A \end{bmatrix} u(t) \tag{4}$$

where

$$q(t) = \begin{bmatrix} W(t) \\ \dot{W}(t) \end{bmatrix} \quad \text{of order } 2n \times 1$$

$$u(t) = [K_A C_W \quad C_A C_W] q(t)$$

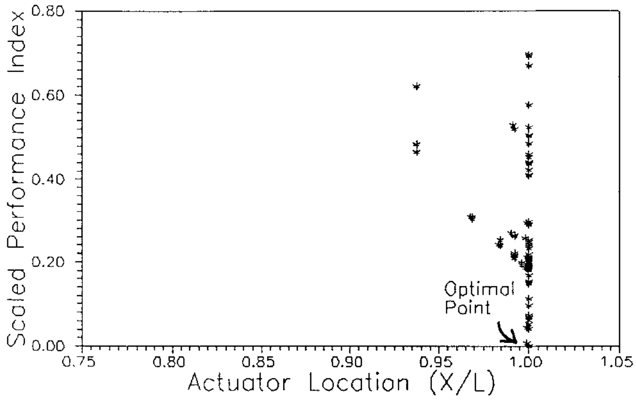
I = identity matrix of order $n \times n$
0 = null matrices of appropriate order
$W(t)$ = modal coordinate vector (order $n \times 1$)
n = number of modes
M = mass matrix
K_s = stiffness matrix
K_A, C_A = controller gain matrices
C_W = output modal matrix (depends on sensor locations along beam)
Φ_A = modal matrix (depends on actuator locations along beam)

The control vector $u(t)$ represents the feedback input. The parameters to be optimized will comprise not only actuator and sensor locations, but also the controller gain matrices K_A and C_A.

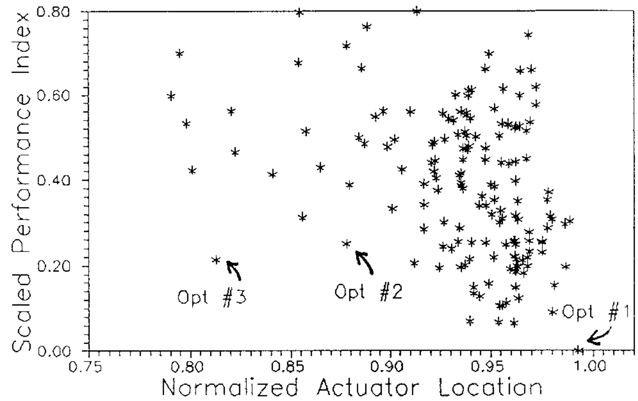
Table 1 Test article properties

t, in.	w, in.	A, in.^{2}	I, in.^{4}	L, in.	Density, lbm/in.^{3}	E, lbf/in.^{2}
0.250	0.82	0.205	0.001068	144	0.0002591	1×10^7

Note: t = beam thickness; w = beam width; lbm = pounds mass; lbf = pounds force; L = beam length; A = cross-sectional area; E = modulus of elasticity; I = moment of inertia.



a) With niching



b) Without niching

Fig. 2 Actuator locations with niching: a) case 1, one actuator, one sensor and b) Case 2: three actuators, three sensors.

Genetic Algorithm Search Process

The GA search process incorporated is outlined as follows:

1) The GA provides actuator and sensor locations and feedback gains. This changes the system equations,[9] and requires that the modeling be updated with each subsequent control configuration. Parameters to be optimized could include actuator types and numbers using varying arrangements. This study will use a specific number of linear force actuators and sensors.

2) Once a control system is defined by the GA, the structure is subjected to a predefined random disturbance. The performance index is subsequently determined through simulation. This investigation treats the time-averaged control energy required to minimize the structural response as the performance index to be minimized. It should be noted that the performance index could be changed without changing the general scheme. The shared fitness, as opposed to the true fitness, is used to select individuals for next generation.

Two cases were tried: 1) one actuator and one sensor case and 2) three actuators and three sensors. For case 1 the number of parameters is 4 (1 actuator, 1 sensor location, and 2 gains), and for case 2 the number of parameters is 24 (3 actuators, 3 sensor locations, and 18 gains). The number of desired optimal solutions was designated to be 2 for case 1 and 10 for case 2. For case 2, the actuators and sensors were restricted so that each pair is located between 0 and

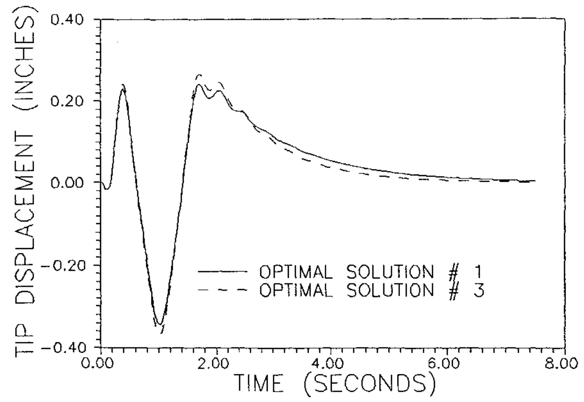


Fig. 3 Closed-loop responses for two "distinctly different" optimal solutions for case 2.

0.33L, 0.33L and 0.66L, and 0.66L and 1.0L, respectively. The coding used for this study is shown below:

10 bits each	10 bits each
1011100010...	1000010111...
Position gains	Velocity gains
10 bits each	10 bits each
0001111011...	1101111011...
Actuator locations	Sensor locations

Results

The technique outlined above was used to solve the two optimization cases presented earlier. Reference 9 includes detailed presentation of the numerical results obtained for the two cases. Figure 2 presents plots of the final population solution for cases 1 and 2. In Fig. 2, the x axis represents an actuator location norm defined as $x/L = x/L$ for case 1 and $x/L = (1/L)(x_{A1}/0.33 + x_{A2}/0.66 + x_{A3}/0.99)$ for case 2. The y axis represents a normalized performance measure, with zero indicating the best solution found by the GA. It is evident from these figures that for the one-actuator, one-sensor case the local optimum is the global optimum. However, for the three-actuator, three-sensor case, three distinct optimal points were found. These optimal points are marked in Fig. 2b. Figure 3 presents closed-loop responses for optimal solutions 1 and 3, as noted in Fig. 2b. Although the actuator locations differ significantly, the closed-loop responses are similar.

Conclusions

This Note has proposed two unique ideas in the realm of optimization for structural control: 1) simultaneous actuator/sensor placement and feedback controller gain optimization and 2) finding more than one "distinctly different" near-optimal solution. To implement these ideas, a genetic algorithm optimization technique was used. For the multiple optimal solutions, we induced niche formation in a genetic algorithm using a sharing mechanism. The end result is the convergence of the genetic algorithm population to multiple locally near-optimal solutions. The concepts were demonstrated using a simple clamped-free beam.

References

[1]Lindberg, R. E., and Longman, R. W., "On the Number and Placement of Actuators for Independent Modal Space Control," *Journal of Guidance and Control*, Vol. 7, No. 2, 1984, pp. 215–221.

[2]Rao, S. S., Pan, T., and Venkayya, V. B., "Optimal Placement of Actuators in Actively Controlled Structures Using Genetic Algorithm," *AIAA Journal*, Vol. 29, No. 6, 1991, pp. 942, 943.

[3]Montgomery, L., KrishnaKumar, K., and Weeks, G., "Structural Control Using Connectionist Learning Principles," AIAA Paper 92-4467.

[4]Chang, I. J., and Soong, T. T., "Optimal Controller Placement in Modal Control of Complex Systems," *Journal of Mathematical Analysis and Applications*, Vol. 75, 1980, pp. 340–358.

[5]Choe, K., and Baruah, A., "Actuator Placement in Structural Control," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 1, 1992, pp. 40–48.

[6]KrishnaKumar, K., and Goldberg, D. E., "Genetic Algorithms in Control System Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, 1992, pp. 735–740.

[7]Goldberg, D. E., *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[8]Deb, K., "Genetic Algorithms in Multimodal Function Optimization," M.S. Thesis, College of Engineering, Univ. of Alabama, Tuscaloosa, AL, 1989.

[9]KrishnaKumar, K., Swaminathan, R., and Montgomery, L., "Multiple Optimal Solutions for a Structural Control Problem Using a Genetic Algorithm with Niching," AIAA Paper 93-3873.

Measures of Modal Controllability and Observability in Vibration Control of Flexible Structures

Zhong-sheng Liu,* Da-jun Wang,* Hai-chang Hu,* and Min Yu*
Peking University, Beijing 100871, People's Republic of China

Introduction

A COMPLEX issue in the control of large flexible space structures is that there may exist repeated or closely spaced modes clumping together in the lower range of the natural frequency spectrum. In practice, only a few modes can be selected for control because of the limited capacity of the hardware. One of the criteria for mode selection is modal controllability and observability. Thus it is desirable to have a method to evaluate the modal controllability and observability quantitatively. Reference 1 investigated the modal controllability and observability for distinct modes. References 2–4 proposed criteria for the controllability and observability of repeated modes, which are, however, not quantitative. References 5 and 6 addressed the model reduction problem based on the singular values of the controllability and observability grammians. The new idea of the present paper is the use of the singular-value decomposition of the input matrix B_0 in defining controllability. This approach is also presented for observability on the basis of duality. It is suitable for both distinct and repeated modes. Using this technique, modal controllability and observability can be quantitatively measured by the associated singular values. In the case of repeated frequency modes, the present method can generate two new groups of orthogonal principal vectors that span the eigensubspace associated with the repeated modes. These principal vectors are ordered in the sequence of their controllability and observability measures.

Background

Consider a linear, time-invariant, second-order control system

$$M\ddot{x} + Kx = Bu(t) \tag{1a}$$

$$y = Cx(t) \tag{1b}$$

where $M = M^T \in R^{n\times n}$, $K = K^T \in R^{n\times n}$, $B \in R^{n\times p}$, $C \in R^{l\times n}$, $x(t) \in R^n$, $u(t) \in R^p$, and $y \in R^l$. Matrices B and C are called the actuator distribution matrix and sensor distribution matrix, respectively, indicating the locations of control forces and sensors, respectively. Mass matrix M and stiffness matrix K are assumed to be positive definite.

Transforming Eq. (1) to the modal coordinates through the coordinate transformation

$$x(t) = \Phi\eta(t) \tag{2}$$

Received May 19, 1993; revision received Jan. 14, 1994; accepted for publication Jan. 20, 1994.

*Department of Mechanics.