JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 31, No. 5, September–October 1994

On-Orbit Nonlinear Structural Parameters Realization via Artificial Neural Network

Rafael Gluck*
TRW Space and Technology Group, Redondo Beach, California 90278
and
Han-Sen Dai† and Walter J. Karplus‡
University of California, Los Angeles, Los Angeles, California 90024

Structural parameters realization is formulated as a pattern recognition problem. Candidate mathematical models are designated as "patterns" with which computer simulations are conducted to generate simulated system responses. Patterns are organized into pattern classes in a topdown dichotomous manner based on the variation of the simulated system responses such that the coherence property of patterns within any pattern class is embedded. An adaptive neural network serves as a pattern classifier. The actual response of the real world system is classified as the pattern class of the most similar system response to determine the most probable mathematical descriptors of structural parameters. The proposed methodology was successfully applied to the realization of the disturbance damping torques at the alpha gimbals of the Phase I Space Station Freedom model. Our experimental data were obtained analytically by simulation with additive Gaussian noise. The results are encouraging, showing a high percentage of correct classification in a noisy environment.

Nomenclature

$a(t)$ = angular acceleration of system response measured at time t
k = number of classes
$\hat{M}_i$ = sample mean response vector of class i
$\hat{M}_0$ = mean response vector of all sample patterns
m = number of feature components in Y
N = total number of samples
N_i = number of samples in class i
n = number of exemplars
p = predetermined sampling number
p_1 = current number of samples in a region while performing topdown dichotomy algorithm
Q_c = exemplar of a correct training classification
Q_i = exemplar or prototype pattern
Q_s = exemplar of a training misclassification
$q(t)$ = squared sum of $a(t)$ over a time frame
$r(t)$ = logarithmic function of $q(t)$
S_b = between-class covariance matrix
S_w = within-class covariance matrix
$s(t)$ = moving average of $r(t)$
T = connection weight matrix of the second layer neural network
u_i = similarity computed by the first layer neural network
v_i = output of the second layer neural network
w_{ij} = connection weight of the first layer neural network
X = system response vector
X_j^i = jth sample response vector in class i
Y = feature vector
y_i = ith component of vector Y
Δt = sampling interval of system response
η = learning rate
$\phi(\cdot)$ = threshold logic function

Received Dec. 30, 1990; presented as Paper 92-2545 at the AIAA 33rd Structures, Structural Dynamics, and Materials Conference, Dallas, TX, April 13–15, 1992; revision received Jan. 22, 1993; accepted for publication Feb. 8, 1993.

*Currently Engineering Consultant, Torrance, CA 90505. Associate Fellow AIAA.
†Currently Senior Engineer, Viewlogic Systems Inc., 42808 Christy St., Suite 200, Fremont, CA 94538. Member AIAA.
‡Professor, Computer Science Department.

I. Introduction

ON-ORBIT parameter identification is indispensable for the development and verification of large, flexible multibody, controlled spacecraft because such spacecraft 1) are unstable in the 1-g environment and cannot be satisfactorily tested before flight, 2) often call for operational scenarios in space that involve large temporal variations of mass and inertia properties, and 3) can be subject to substantial growth via on-orbit construction.

For the past two years, TRW has concentrated its efforts in the realm of structural parameter identification on the dynamic programming filter (DPF), a member of a class of nonlinear filtering techniques based on modern control theory.[1] A major advantage of the DPF over many other structural estimation methodologies is its applicability to nonlinear mathematical models that characterize all real multibody dynamic systems.[2,3] With the DPF, candidate uncertain parameters may represent linear and/or nonlinear characteristics of multibody systems. For instance, joints with deadbands, discrete nonlinear springs, and dampers are examples of nonlinear parameters, whereas component resonant frequencies, distributed damping, and effective participation factors between sensor and actuators belong to a linear parameter set. In the general formulation of the DPF identification methodology, no distinction is made between linear and nonlinear parameters; both sets of parameters are dealt with in the same fashion.

One of the most difficult tasks in the identification of nonlinear structural parameters is to determine the most probable mathematical descriptors of the nonlinearities in question.[4] In reality, a nonlinear structural parameter may assume any of a large number of candidate descriptors from which one must choose the most likely candidate. This task, which is often referred to as system realization, constitutes a most intensive computational task.

In the sequel, we propose to apply artificial neural networks to the realization of nonlinear structural parameters, as a "front end" of the DPF methodology mentioned earlier. A brief summary of the proposed approach follows.

System realization is treated here as a pattern recognition problem.[5] Computer simulations are conducted over a set of candidate models. The simulated system responses are organized into clusters that correspond to pattern classes and are associated with their models. When the actual system response is presented, the real world system is classified within the model class that has the most similar system response.

The methodology proposed includes an off-line training phase and an on-line recognition phase. The training phase consists of four steps: pattern designation, feature extraction, classifier implementation, and performance evaluation. Pattern designation formulates distinct nonlinear mathematical models into pattern classes. The feature extraction scheme extracts invariant or similar properties of a pattern class as it enhances the difference between pattern classes. A neural network classifier is then trained to memorize the representative feature vectors of every pattern class. Finally, performance evaluation is conducted by estimating the misclassification probability and the noise effect on the classifications.

In the recognition phase, the response vector of the real world system is taken as an input to the pattern recognition system obtained in the training phase. This system is classified within the class of the most similar response vector. The identified model class contains the most probable mathematical descriptors that characterize this system. Since the pattern recognition system is massively parallel, system realization is accomplished instantaneously.

II. Methodology

A. Pattern Designation

Pattern designation formulates system realization as a pattern recognition problem. A piecewise linear approximation technique is proposed to approximate the unknown parameters that are functions of system states and to define a parameter space representing all possible candidate models. This parameter space is then partitioned into pattern classes via the topdown dichotomy algorithm.

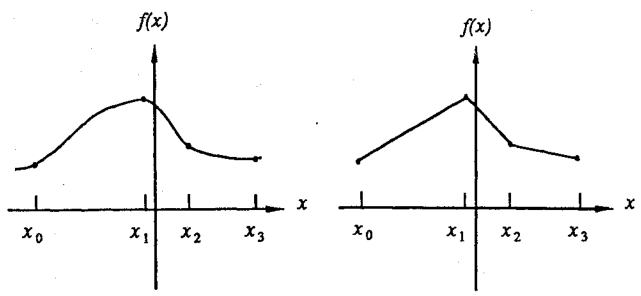
1. *Piecewise Linear Approximation*

The continuous domain of the unknown parameter functions is discretized into discrete elements. The value of the unknown function at each discrete point of the function domain becomes a variable in the function approximation. In the piecewise linear function scheme, the function within each discrete element is approximated by a straight line or a hyperplane. The surface or curve of a function $f(x)$ is then mapped to a point in the new parameter space. In other words, any point in the parameter space is a candidate model of the unknown system and is also designated to be a pattern for the pattern recognition system.

Suppose that $f(x)$ is an unknown function and the interval of interest in the domain of $f(x)$ is $[x_0, x_3]$. Figure 1a shows a possible curve of $f(x)$. Figure 1b shows an example of an approximation of $f(x)$ by piecewise linear functions. The new parameter space is a four-dimensional space constructed by $f(x_0)$, $f(x_1)$, $f(x_2)$, and $f(x_3)$.

2. *Partitioning Parameter Space*

Partitioning a parameter space deals with the separation of patterns into classes. A pattern class contains patterns that cause the model to generate similar system responses. The topdown dichotomy algorithm divides the parameter space into disjoint regions according to the variation of various simulated system responses that are generated by various patterns sampled from the parameter space. Thus, patterns of the same class have system responses varying within a certain threshold. In other words, there exists a certain degree of similarity among the patterns in one class.



a) b)

Fig. 1 a) Curve of the unknown function $f(x)$ and b) piecewise linear approximation of $f(x)$.

Algorithm 1: Topdown Dichotomy

Input. A mathematical model with unknown functions for some system components and a parameter space representing these unknown functions.

Output. A set of disjoint regions of the parameter space as well as a between-class covariance matrix and a within-class covariance matrix of the simulated system responses of the sample patterns.

1) Start with the whole parameter space as one region and an initial number of sample points in this region $p_1 = 0$, where p_1 denotes the current number of sample patterns in this region.

2) For a region in the parameter space, randomly select $p - p_1$ patterns with uniform distribution, where p is a predetermined number of total sample patterns in a region.

3) Conduct computer simulations of these $p - p_1$ patterns, and combine with the existing simulation results of p_1 patterns to get p sets of simulated system responses.

4) Calculate the variation of all of these p response vectors whose variance could be used as a measure of the variation.

5) Repeat steps 2–4 for all regions in the parameter space.

6) If there exists a region with variation exceeding a specified threshold, this region is split into two. Otherwise, calculate the covariance matrices and stop.

7) Calculate p_1, the current number of sample points, in these two new regions. Go to step 2 and handle each region independently and recursively.

In this algorithm, the parameter p is chosen such that all p points in a region are able to characterize the variation property of this region.

B. Feature Extraction

The objectives of feature extraction are to increase the clustering within any one class, and to enhance the separation between classes, and to reduce the dimensionality of the data. Therefore, the within-class and between-class covariance matrices are used as the measurements of the clustering within any one class and the separation between classes, respectively. Here the feature extraction scheme is a linear transformation that minimizes the variation of features within classes as it maximizes the variation of features between classes.[6,7] The transformation matrix is composed of the eigenvectors corresponding to the largest eigenvalues of the matrix $S_w^{-1}S_b$ where S_w and S_b are the estimated within-class and between-class covariance matrices of the system responses, respectively.

In summary, the procedure of constructing the feature extractor is as follows:

1) Estimate the sample scatter (covariance) matrices:

$$S_w = \frac{1}{\sum_{i=1}^{k} N_i - k} \sum_{i=1}^{k} \sum_{j=1}^{N_i} \left(X_j^i - \hat{M}_i\right)\left(X_j^i - \hat{M}_i\right)^T \tag{1}$$

$$S_b = \frac{1}{k-1} \sum_{i=1}^{k} \frac{N_i}{N} (\hat{M}_i - \hat{M}_0)(\hat{M}_i - \hat{M}_0)^T \tag{2}$$

where $N = \sum_{i=1}^{k} N_i$, and

$$\hat{M}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} X_j^i \tag{3}$$

and

$$\hat{M}_0 = \frac{1}{\sum_{i=1}^{k} N_i} \sum_{i=1}^{k} \sum_{j=1}^{N_i} X_j^i \tag{4}$$

are the sample mean response vector of class i and the sample mean response vector of all sample patterns, respectively.

2) Calculate the eigenvectors and eigenvalues of the matrix $S_w^{-1}S_b$, normalize the eigenvectors, and sort the eigenvalues in a descending order.

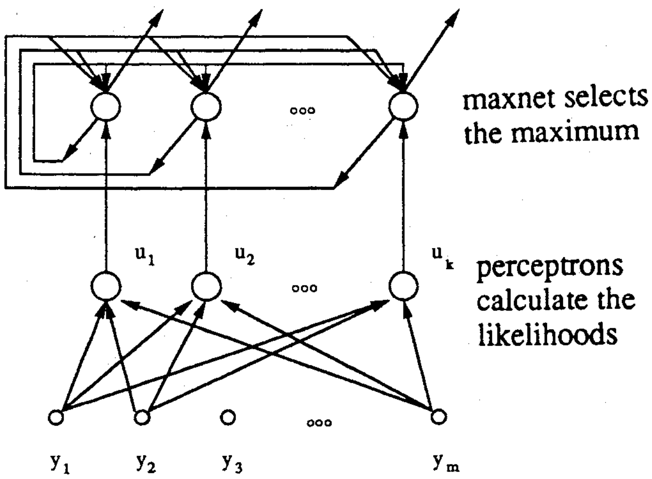


Fig. 2 Neural network classifier.

3) Select the first m eigenvectors to construct the transformation matrix A.

4) The feature vector Y of an observation X is obtained by applying the transformation

$$Y = AX \tag{5}$$

C. Classification—Neural Network Classifier Based on the Hamming Net

In this study, we employ an adaptive neural network as the pattern classifier. The classifier for the feature extraction scheme can be designed in the sense of Euclidean distance to find the nearest exemplar representing a pattern class. That is,

$$\min_i \left[(Y - Q_i)^T (Y - Q_i)\right] \tag{6}$$

where Y is obtained from the output of the feature extractor. The input feature vector Y is classified as the class of the nearest exemplar. This expression is equivalent to

$$\max_i \left[\left(Y - \frac{Q_i}{2}\right)^T Q_i\right] \tag{7}$$

It turns out that this classifier can be implemented using an artificial neural network with a similar architecture to the Hamming net.[8]

Figure 2 shows a neural network implementation of the classifier. The perceptron of the first layer computes the following function:

$$u_i = \sum_{j=1}^{m} w_{ij} y_j - (Q_i)^T Q_i/2 \tag{8}$$

where u_i is the similarity or likelihood of Y to Q_i. In this equation, the connection weight w_{ij} is the jth component of Q_i and $Q_i = [w_{i1}, w_{i2}, \ldots, w_{im}]^T$.

The second layer is a maxnet that performs the winner-take-all operation and picks up the maximum. Mathematically, the maxnet iteratively operates the following function to find the maximum

$$v_i(t+1) = \phi\left[\sum_{i=1}^{n} T_{ij} v_j(t)\right] \tag{9}$$

where the connection weight

$$T_{ij} = \begin{cases} 1 & i = j \\ -w \quad \left(w < \dfrac{1}{n-1}\right) & \text{otherwise} \end{cases} \tag{10}$$

The threshold logic function $\phi(\cdot)$ is defined as

$$\phi(u) = \begin{cases} u \text{ if } u \geq 0 \\ 0 \text{ if } u < 0 \end{cases} \tag{11}$$

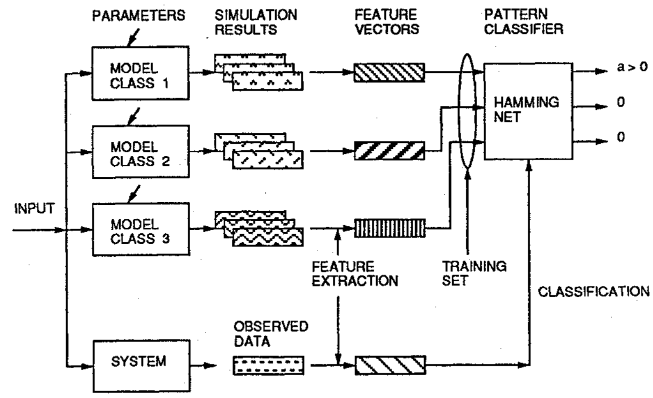


Fig. 3 Feature extraction and pattern classification.

The input is fed from the output of the first layer before time zero and then removed. After convergence, the node with a positive output value indicates the exemplar with which the feature vector Y is classified. The input pattern is therefore classified as the class represented by this exemplar.

An adaptive technique is applied to find the appropriate exemplars. Since the decision boundary may be nonlinear, every pattern class could have multiple exemplars to govern its region. The learning rule that is a modification of Kohonen's LVQ is as follows[9]:

1) For correct classification of the feature vector Y with the exemplar Q_c, learning only takes place in the early several training epochs and

$$Q_c(t+1) \leftarrow Q_c(t) + \eta[Y - Q_c(t)] \tag{12}$$

2) If Y is misclassified with Q_s and the nearest exemplar of correct class to Y is Q_c,

a) punish Q_s

$$Q_s(t+1) \leftarrow Q_s(t) - \eta[Y - Q_s(t)] \tag{13}$$

and b) reward Q_c

$$Q_c(t+1) \leftarrow Q_c(t) + \eta[Y - Q_c(t)] \tag{14}$$

In the learning rule, t is the learning time in a discrete form. And $0 \leq \eta < 1$ is the learning rate that decreases monotonically with time. The convergence of this algorithm is guaranteed when η goes to zero. The initial exemplars are set to be the mean feature vectors of their classes with small additive noise. Step 1 of the learning rule evenly distributes the exemplars inside the regions of their representing classes. After spreading the exemplars into their corresponding regions, the learning process only takes place on misclassification to refine the decision boundary.

D. Performance Evaluation

The objectives of performance evaluation can be addressed in two aspects: 1) evaluation of the problem-solving procedure and 2) reliability of the classification result. If the performance is not acceptable, the problem-solving procedure, from function approximation, to feature extraction, to classifier implementation, requires re-examination and modification to improve the performance.

The performance evaluation is based on the misclassification probability that is estimated from the training set, a testing set generated randomly, different parameter function approximations as a testing set, and the aforementioned three sets including different noise levels. The percentage of correct classification also provides a measure of confidence in the realization of the mathematical descriptors of the unknown parameters.

E. Summary

We formulate system realization as a pattern recognition problem. A piecewise linear approximation technique is employed to approximate the unknown parameter functions. We have also developed a topdown dichotomy algorithm to partition the parameter space and to define pattern classes. The feature extraction scheme is based on the Fisher's discriminant method. An adaptive neural network is introduced to perform the classification task. The training of feature extractor and pattern classifier as well as the recognition process are shown in Fig. 3. The observation of the unknown system is fed to the pattern recognition system that classifies the unknown system within a class of candidate models instantaneously. This procedure can be conducted iteratively to refine the model class.

III. Case Study: Space Station Model

In this section, the nonlinear structural parameters realization methodology is applied to the Phase I Space Station Freedom (SSF) model.

A. System

Figure 4 presents the SSF model consisting of a central body and starboard and port bodies, all modeled as flexible bodies. Solar panels are attached to the extraneous bodies that are connected to the central body by one degree-of-freedom alpha gimbals. The simulated maneuver depicted a transient rotation of the solar arrays to achieve perpendicularity with respect to the sun line while maintaining the central body in a predetermined attitude control mode. The characteristics of the alpha gimbals are thus very important in designing the alpha gimbal control system.

B. Objective

The objective of this effort is to realize the mathematical characteristics of the disturbance damping torques at the alpha gimbals as functions of the gimbals' angular velocities.

C. Experiment Design

Our on-orbit experimental data were obtained analytically by simulation. In the selected maneuver, the extraneous bodies are rotated 15 deg relative to the central body, whereupon the gimbal control system is suddenly shut off. The free vibration that ensues is monitored by angular accelerometers mounted to the solar panels and sampled every 0.01s for a period of 50s. Figure 5 shows a typical system response in the time domain for the aforementioned Space Station maneuver.

D. Experiment Data

The application of our methodology to the realization of the disturbance damping torque at the alpha gimbals focused on the time domain response because of its sensitivity to variations of this parameter. In contrast, we found that the damping torque does not affect the frequency of the system's response significantly, and therefore frequency domain analysis techniques are not particularly useful in promoting the classification of this problem.

Our methodology utilizes the logarithm of the power density of the acceleration decay signal averaged over time. The predominant flexible mode in the system's response has a frequency of 1.47 Hz (see Fig. 5). The basic element of the processed data is the squared sum of the system's response over a time frame window of 1.36 s that constitutes a two-cycle period of the predominant mode. The decay signal is given then as

$$q(t) = \sum_{i=0}^{135} a(t + i\Delta t) * a(t + i\Delta t) \qquad (15)$$

where $\Delta t = 0.01$ s and $a(t)$ is the angular acceleration of the system response measured at time t.

A moving average technique is applied to smooth the decay signal,

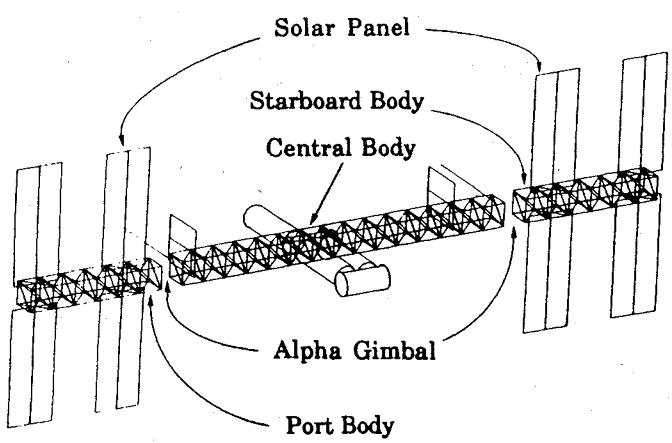


Fig. 4 Space Station Freedom model.

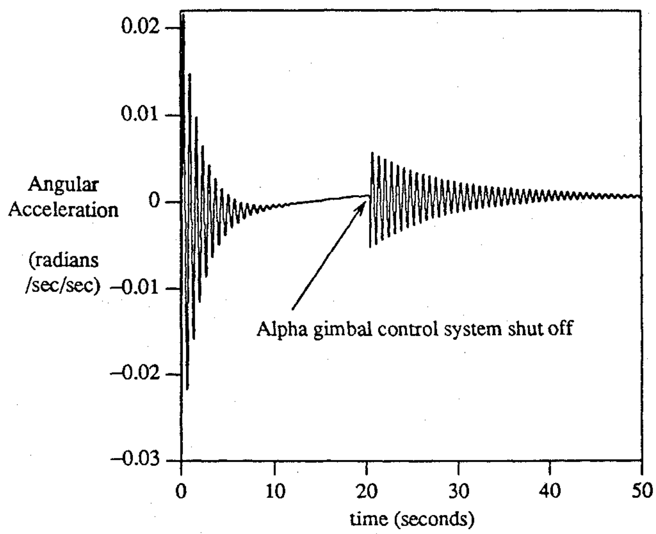


Fig. 5 System acceleration response in the time domain.

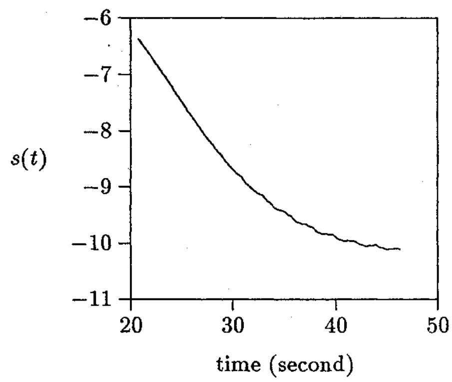


Fig. 6 Plot of $s(t)$.

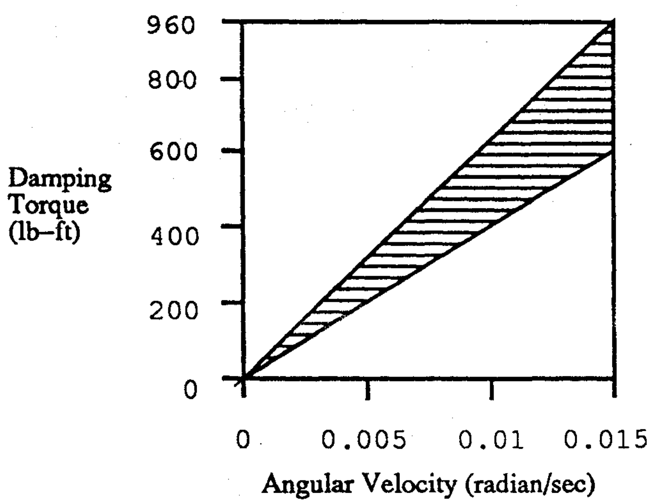


Fig. 7 Possible region of the unknown damping torque function.

Table 1 Classification results with noise of fixed standard deviations

Standard deviation	0	0.00005	0.0001	0.00015	0.0002
SNR[a]	∞	30	15	10	7.5
Percent of correct classification	100	100	92.19	79.69	62.75

[a]SNR $\triangleq$ rms of signal/ rms of noise.

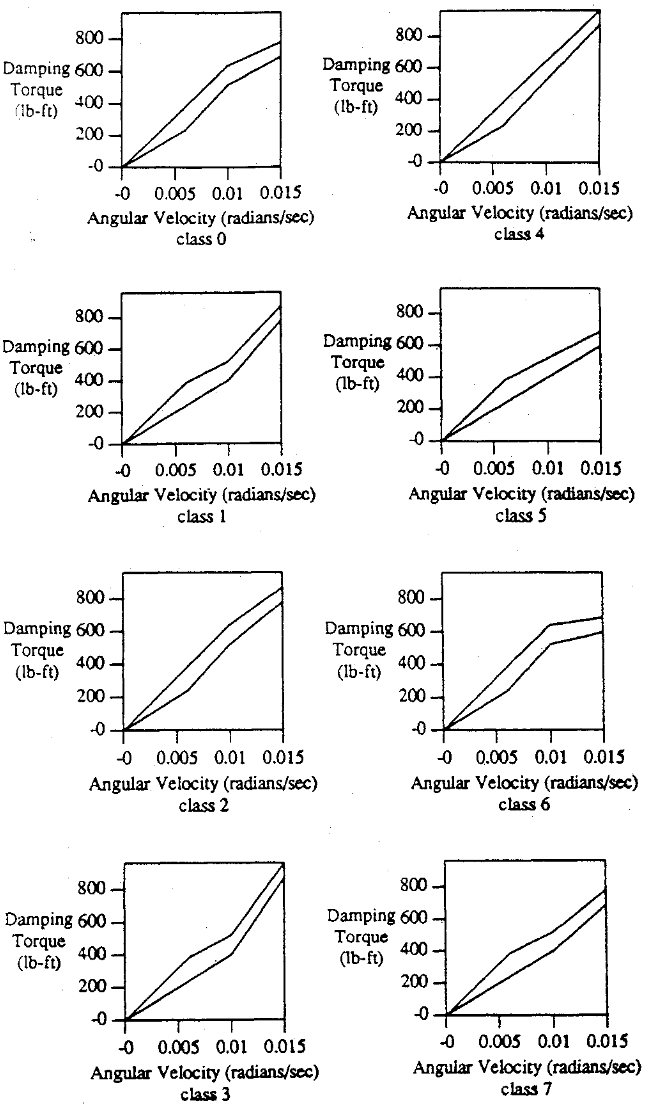


Fig. 8 Pattern classes.

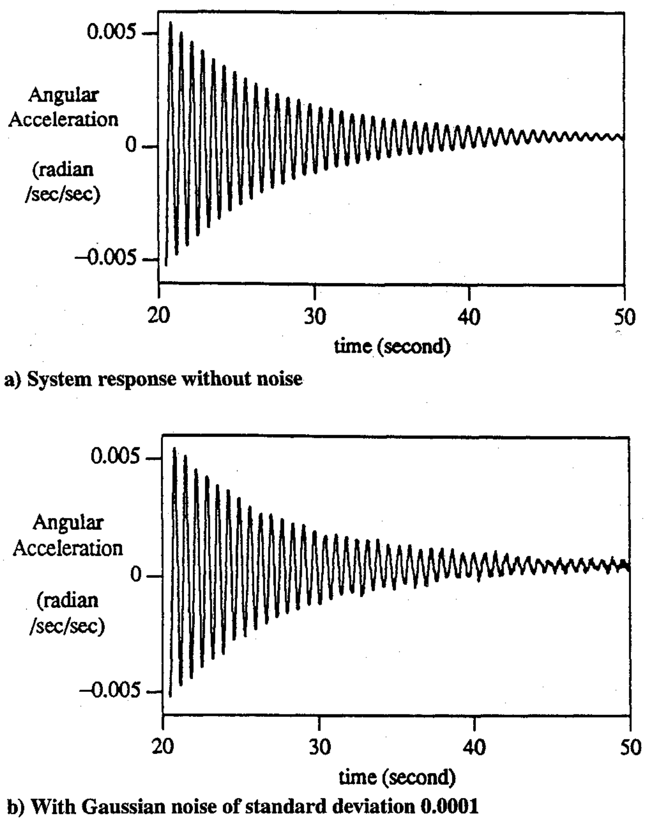


Fig. 9 System response with and without noise.

yielding

$$s(t) = \frac{1}{L+R+1}\sum_{i=-L}^{R} r(t + i\Delta t) \qquad (16)$$

where $r(t) = \log[q(t)]$ and $L = R = 25$ are empirical constants. The signal used in this application of our methodology was sampled from $s(t)$ at 256 points. A plot of $s(t)$, representing the vibration decay shown in Fig. 5 (following the gimbal control system shutoff), is shown in Fig. 6.

E. Classification Results

The disturbance damping torque of interest was assumed to be bounded inside the shaded region shown in Fig. 7. Figure 8 illustrates the classes of functional regions obtained via the topdown dichotomy algorithm.

The classification results were generated in a noisy environment, where various Gaussian noise levels were added to the system response. Two types of noise were examined. One was a Gaussian noise with a fixed standard deviation. The other was a Gaussian noise with a standard deviation proportional to the amplitude of the signal.

1. *Gaussian Noise with a Fixed Standard Deviation*

Table 1 presents percentages of correct classification obtained in the presence of Gaussian noise characterized by fixed standard deviations. The overall signal-to-noise ratio (SNR) in the table was calculated from the standard deviation value that is set equal to the root-mean-square (rms) amplitude of the noise. It should be noticed that over 90% of correct classification was obtained in the presence of noise with an overall SNR of 15. Figure 9 presents the temporal responses of the SSF model with and without noise.

2. *Gaussian Noise Proportional to the Signal's Amplitude*

Table 2 contains classification results obtained in the presence of Gaussian noise where the rms amplitude of the noise is proportional to the signal's amplitude. As the signal decays (see Fig. 5), the noise rms amplitude is reduced until a minimum threshold, defined as minimum noise level (MNL), is reached; thereafter, the noise is maintained at the MNL, regardless of the signal's amplitude. As shown in Table 2, our realization methodology yields high percentages of correct classification in the presence of noise levels as high as 20% of the signal's amplitude. Figure 10 shows the temporal system response with noise levels of 10 and 20% of the signal's amplitude and MNL standard deviation = 0.0001.

Table 2 Classification results with noise proportional to signal's amplitude

Noise level, % of signal	5	10	15	20
Percent of correct classification with MNL = 0.00005	100	95.31	89.06	85.94
Percent of correct classification with MNL = 0.0001	90.62	90.62	87.50	79.69

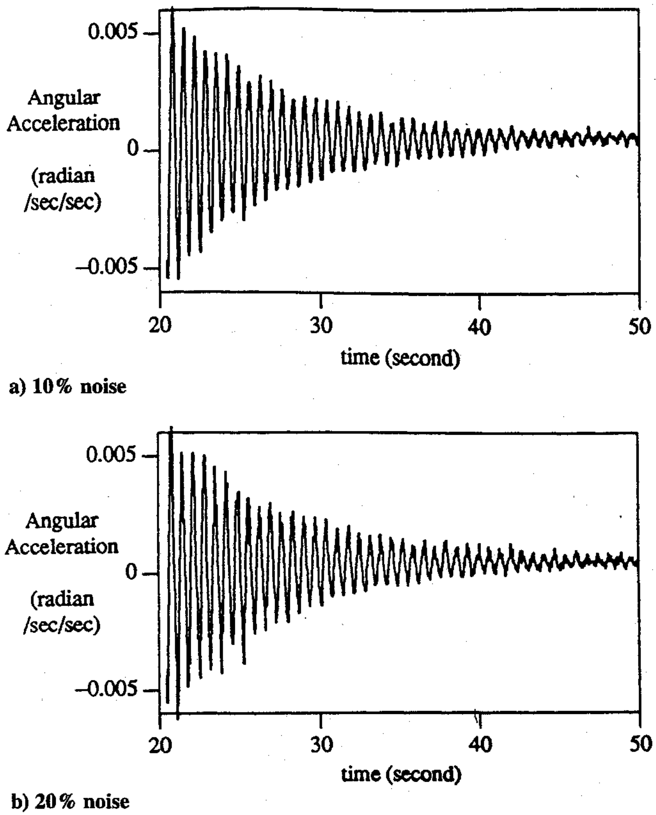


Fig. 10 System response with 10 and 20% noise at MNL = 0.0001.

IV. Conclusions and Recommendations

This paper describes a methodology that utilizes an artificial neural network to determine the most probable mathematical descriptors of nonlinear structural parameters. These heretofore unattainable descriptors, denoted as patterns, were used to encompass all portions of a function space that are plausible locations of the model parameters to be realized. A pattern recognition algorithm was then developed and used to determine from noisy observations which of the patterns best described the modeled parameters.

The methodology was successfully applied to the realization of the disturbance damping torques at the alpha gimbals of a realistic model of the SSF. The results are encouraging, showing a high percentage of correct classification in a noisy environment. This initial structural parameter realization effort should now be enhanced to produce a robust and useful operating tool. This undertaking may consist of the following activities:

1) Automate pattern designation and construction of feature extractors to increase the number of models remembered and recognized by the artificial neural network.

2) Increase the variety of simulated on-orbit experiments to improve the quality of the observable data for nonlinear structural parameter characterization.

3) Generalize the demonstrated concept of pattern recognition to include a wide variety of geometric nonlinearities such as slippage and backlash in joints and gear trains, material nonlinearities, and friction.

References

[1]Simonian, S. S., "On-Orbit System Parameter Identification," *Workshop on Modal Determination for Large Space System*, JPL D-5574, Vol. 1, March 1988.

[2]Juang, J. N., and Pappa, R. S., "Effects of Noise on Model Parameters Identified by Eignesystem Realization Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 3, 1986.

[3]Mook, D. J., and Lew, J. S., "A Robust Algorithm for System Realization/Identification," *Journal of Astronautical Science*, Vol. 38, No. 2, 1990.

[4]Glass, B. J., and Hanagud, S., "Identification of Time-Varying Structural Dynamic Systems: An Artificial Intelligence Approach," *AIAA Journal*, Vol. 30, No. 5, 1992.

[5]Karplus, W. J., "System Identification and Simulation—A Pattern Recognition Approach," *AFIPS—Conference Proceedings*, Vol. 41, 1972, pp. 385–392.

[6]Fisher, R. A., "The Use of Multiple Measurements in Taxonomic Problems," *Annual Eugenics*, Vol. 7, 1936, pp. 179–188.

[7]Duda, R. O., and Hart, P. E., *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[8]Lippmann, R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, April 1987, pp. 4–22.

[9]Kohonen, T., *Self-Organization and Associative Memory*, Springer-Verlag, New York, 1984.