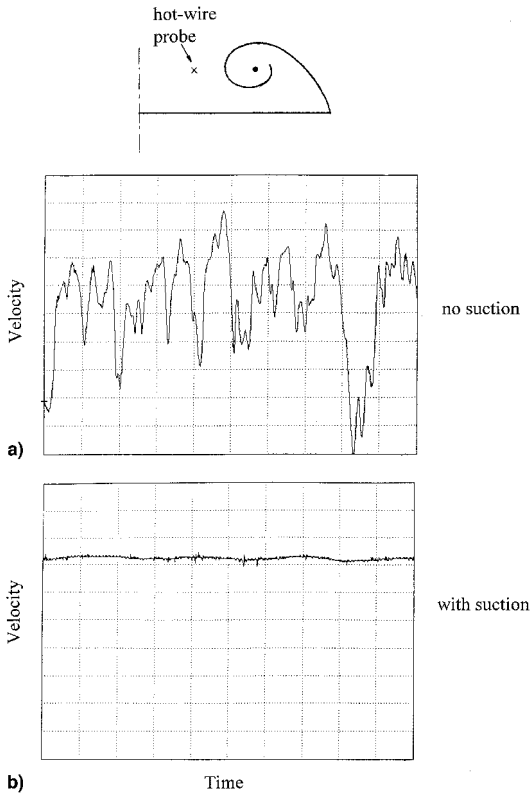

Fig. 3 Hot-wire signals at (x/c = 0.8, y/s = 0.3, and z/s = 0.22) a) without and b) with suction, y_s/s = 0.7, C_μ = 0.72, α = 25 deg, and $Re = 3.5 \times 10^4$.

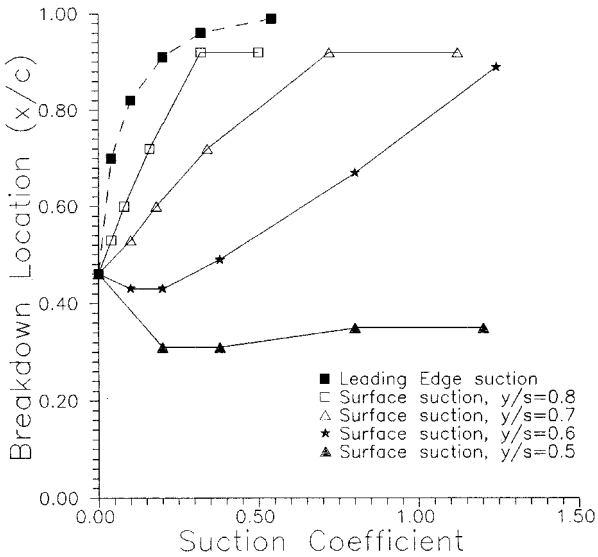

Fig. 4 Variation of breakdown location as a function of suction coefficient.

unsteady nature of flowfield when breakdown exists[7] and an almost steady flow in the absence of breakdown.

The effect of location of suction slit is shown in Fig. 4, together with the results for the leading-edge suction. Note that, in the absence of suction, the spanwise location of vortex axis is $y_v/s \cong 0.60$. For the smallest value of y_s/s tested (y_s/s = 0.5), vortex breakdown location moves upstream when suction is applied. Therefore, the effect of suction on vortex breakdown is negative. For y_s/s = 0.6, there is almost no change for small values of suction coefficient. However, there is an improvement at large values of suction coefficient. It is seen that, as the suction slit gets closer to the leading edge, suction becomes more effective in delaying vortex breakdown (y_s/s = 0.7 and 0.8). Even for small values of suction coefficient, considerable delay of breakdown is achieved. However, surface suction is less effective than leading-edge suction. For very small values of suction coefficient, large improvements can be obtained. On the other hand, both leading-edge suction and surface suction seemed to lose effectiveness when breakdown location reached the trailing-edge region. It is clear that the adverse pressure gradient near the trailing edge is very strong.

Conclusions

The effect of suction on the wing surface on vortex breakdown has been investigated. Suction has been found to be more effective in delaying vortex breakdown for suction slits closer to the leading edge. The exact mechanism of how the surface suction affects vortex breakdown is not clear. However, surface suction is less effective than the leading-edge suction.

Acknowledgment

This work was supported by the U.S. Air Force Office of Scientific Research Grant F49620-93-1-0516.

References

[1]Werle, H., "Sur l'Eclatement des Tourbillons d'Apex d'une Aile Delta aux Fibles Vitesses," *La Recherche Aeronautique*, Vol. 74, Jan. 1960, pp. 23–30.

[2]Parmenter, K., and Rockwell, D., "Transient Response of Leading-Edge Vortices to Localized Suction," *AIAA Journal*, Vol. 28, No. 6, 1990, pp. 1131–1133.

[3]Wood, N. J., Roberts, L., and Celik, Z., "Control of Asymmetric Vortical Flows over Delta Wings at High Angles of Attack," *Journal of Aircraft*, Vol. 27, No. 5, 1990, pp. 429–435.

[4]Gu, W., Robinson, O., and Rockwell, D., "Control of Vortices on a Delta Wing by Leading-Edge Injection," *AIAA Journal*, Vol. 31, No. 7, 1993, pp. 1177–1186.

[5]McCormick, S., and Gursul, I., "Effect of Shear Layer Control on Leading Edge Vortices," *Journal of Aircraft*, Vol. 33, No. 6, 1996, pp. 1087–1093.

[6]McCormick, S., "Effect of Leading-Edge Suction on Vortices over Delta Wings," M.S. Thesis, Univ. of Cincinnati, Cincinnati, OH, 1995.

[7]Gursul, I., "Unsteady Flow Phenomena over Delta Wings at High Angle of Attack," *AIAA Journal*, Vol. 32, No. 2, 1994, pp. 225–231.

Neural Network Parameter Extraction with Application to Flutter Signals

B. H. K. Lee*
National Research Council,
Ottawa, Ontario K1A 0R6, Canada
and
Y. S. Wong†
University of Alberta,
Edmonton, Alberta T6G 2G1, Canada

Introduction

DETERMINATION of the flutter boundary of an aircraft requires accurate measurements of frequencies and damp-

Received Oct. 6, 1996; revision received July 31, 1997; accepted for publication Sept. 29, 1997. Copyright © 1997 by B. H. K. Lee and Y. S. Wong. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Head, Experimental Aerodynamics and Aeroelasticity Group, Institute for Aerospace Research; also Adjunct Professor, Department of Mathematical Sciences, University of Alberta, Edmonton, Alberta T6G 2G1, Canada. Associate Fellow AIAA.

†Professor, Department of Mathematical Sciences.

ing values of the critical vibration modes as a function of the flight velocity. Numerous methods[1,2] have been proposed for real-time flutter identification, with varying degrees of success. With the rapid advances in wavelet theory[3,4] for signal processing and the use of artificial neural networks[5,6] to model complex characteristics of nonlinear systems, more advanced methods to analyze flutter signals can be devised based on these modern developments. In this Note, a technique is outlined to determine frequencies and damping from a time series formed by a linear superposition of a number of exponentially decaying sine waves and corrupted by noise, given by

$$y(t) = \sum_{i=1}^{N} A_i e^{-\alpha_i t} \sin(\omega_i t + \phi_i) + n(t) \tag{1}$$

where N denotes the number of modes, A_i, α_i, ω_i, and ϕ_i represent the amplitude, damping ratio, frequency, and phase angle of the ith mode, respectively, and $n(t)$ is the noise level in the signal. Equation (1) represents a multimode vibratory time series that is often used as a model equation to simulate flutter signals generated by an impulsive input. It also represents the decaying portion of the response signals from sine dwell or sine sweep excitations of the aircraft. This equation has also been used to generate data for evaluation of various flutter analysis techniques.[2,7] Unlike Refs. 2 and 7, where the noise level was introduced as the ratio of the peak value of a random noise to the amplitude A_1 in Eq. (1), $n(t)$ is defined here as

$$n(t) = c \times y(t) \times \text{Gaussian random number} \tag{2}$$

where c is a constant.

Analysis

Wavelet Package for Signal Processing

The coefficients in wavelet decomposition can be computed using a fast wavelet transform. Let ϕ be a real-valued, compact supported function with orthonormal shifts that satisfy the conditions of multiresolution and, in particular, the refinement equation

$$\phi(x) = \sum_{j \in Z} a(j)\phi(2x - j) \tag{3}$$

for some sequence $a(j) \in l_2(Z)$. Once ϕ and $a(j)$ are determined, an orthogonal wavelet ψ can be constructed. Using a discrete wavelet transform, a wavelet sum can be formed from a function f as

$$S_n = \sum_{j \in Z} f_j \phi_{j,n} \tag{4}$$

In many applications, the nonzero values f_j are taken to be $f(j2^{-n})$, where the point $j2^{-n}$ corresponds to the support of $\phi_{j,n}$. Because $S_n \in S^n$, it can be written as

$$S_n = P_o S_n + \sum_{k=0}^{n} \sum_{j \in Z} c(j, k)\psi(2^k x - j) \tag{5}$$

A wavelet transform is an algorithm to compute the coefficients $c(j, k)$ from the f_j, and an inverse wavelet transform recomputes S_n from the coefficients $c(j, k)$. There are various wavelet software packages available, such as the MATLAB Wavelet Toolbox.[8] In this Note, the adapted wavelet package software is used, which includes a library of orthonormal bases, such as Daubechies' wavelets,[3] Coifman's and Wickerhauser's wavelets,[4] and localized sine and cosine transforms.

Using the example studied in Ref. 7, Eq. (1) is written as follows after using the parameters given in that reference:

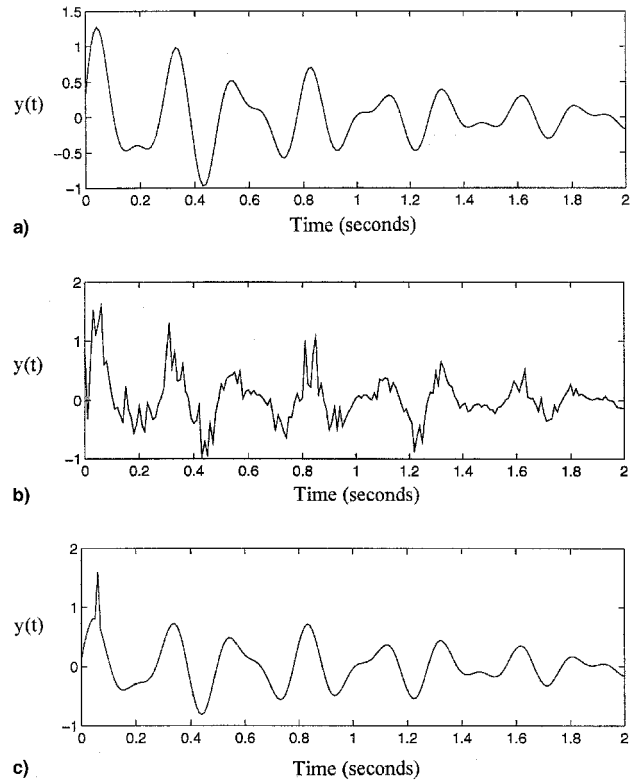

Fig. 1 a) Time series of two exponentially decaying sine waves, b) time series contaminated with noise, c = 0.5, and c) denoised time series using wavelet package.

$$y(t) = e^{-0.9818t} \sin(24.54t + 0.1) + 0.5e^{-0.7854t} \sin(39.27t + 0.5) + n(t) \tag{6}$$

This equation will be used to generate numerical data for the test case in this Note. Figure 1a shows the time series in the absence of noise, and Fig. 1b shows the contaminated signal for c = 0.5 in Eq. (2). The wavelet package can be used to denoise the signal by introducing a threshold into the wavelet coefficients, so that reconstruction is performed using only those coefficients above the threshold value. The reconstructed waveform is shown in Fig. 1c, which gives a very good presentation of the original signal.

The wavelet package can be used to separate the two-mode signal in Eq. (6), which results in two exponentially decaying sine waves. The decomposed one-mode signal will be affected by the choice of the wavelet basis. For the simulated signal given in Eq. (6), the localized cosine transform was used because it gives better results when compared with those using other basis functions. There are numerous techniques to solve for the frequency and damping of the two modes. Here, a neural network is described that is designed for a single mode, as a multimode signal can always be decomposed into single-mode components using wavelets.

Artificial Neural Network

A neural network has the capability to model highly complex nonlinear systems with only limited input information. Simply stated, it consists of a set of calculation units (neurons) that are connected to other units by weights. Each unit computes a weighted sum of the inputs and translates it to outputs making use of a transfer function. There are numerous degrees of sophistication in the design of a network, but for this application a two-layer model with feedforward-only connections (Fig. 2) is used. The output vectors are given by

$$\boldsymbol{a}^1 = f^1(W^1\boldsymbol{p} + \boldsymbol{b}^1), \qquad \boldsymbol{a}^2 = f^2(W^2\boldsymbol{a}^1 + \boldsymbol{b}^2) \tag{7}$$

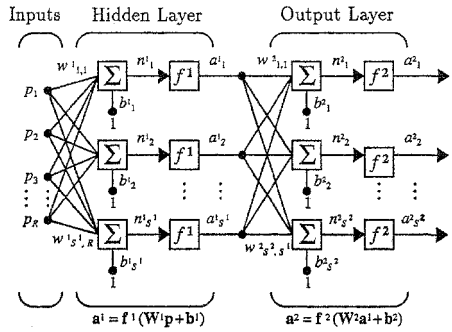

Fig. 2 Two-layer neural network with feedforward connection.

Here f^1 and f^2 are the transfer functions and $\boldsymbol{p}$ is the input vector. Once the weighted matrices W^1 and W^2 and the bias vectors $\boldsymbol{b}^1$ and $\boldsymbol{b}^2$ are determined, the output $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$ can be computed from Eq. (7). A supervised learning rule is used in the present network, and W and $\boldsymbol{b}$ are determined from a backpropagation algorithm. In the training process, a set is obtained

$$(p_1, t_1), (p_2, t_2), \ldots, (p_k, t_k)$$

where p_i is the input to the network and t_i is the corresponding current output or target. The learning rule is used to adjust the weights and biases of the network to make the network outputs close to the targets. This is achieved by minimizing a performance index $F(x)$, defined as

$$F(x) = \boldsymbol{e}(k)^T \boldsymbol{e}(k) \tag{8}$$

where $e(k) = t(k) - p(k)$ in the hidden layer and $e(k) = t(k) - a(k)$ in the output layer. Using the steepest descent method, the weights and biases can be derived as

$$W^2(k+1) = W^2(k) - \alpha \boldsymbol{S}^m \boldsymbol{a}^T, \quad \boldsymbol{b}^2(k+1) = \boldsymbol{b}^2(k) - \alpha \boldsymbol{S}^m \tag{9}$$
$$W^1(k+1) = W^1(k) - \alpha \boldsymbol{S}^m \boldsymbol{p}^T, \quad \boldsymbol{b}^1(k+1) = \boldsymbol{b}^1(k) - \alpha \boldsymbol{S}^m$$

where α is the learning rate and $m = 1, 2$ denotes the hidden and output layers, respectively. $\boldsymbol{S}^m$ is a sensitivity vector given by

$$\boldsymbol{S}^m = \frac{\partial F}{\partial n_i^m} = \begin{bmatrix} \dfrac{\partial F}{\partial n_1^m} \\ \dfrac{\partial F}{\partial n_2^m} \\ \vdots \\ \vdots \\ \dfrac{\partial F}{\partial n_s^m} \end{bmatrix} \tag{10}$$

Initially, the elements of W^1, W^2, $\boldsymbol{b}^1$, and $\boldsymbol{b}^2$ are generated from a random-number generator. To accelerate the convergence in the training process, other optimization algorithms, such as the conjugate gradient technique or Newton's method, can be used in place of the steepest descent algorithm.

The neural network code is a custom-written program in C^{++} language for a desktop personal computer. The input signal is of the form $y(t_j) = e^{-\alpha t_j} \sin(2\pi f \times t_j)$, where $j = 1, 2, \ldots, 30$. The inputs contain 30 data points, and thus the network requires 30 nodes. It has 10 neurons in the hidden layer and 2 neurons in the output layer, as only values of the frequency and damping are required. Various ranges of the parameters α and f have been used in training the network. For illustration purposes, $0.5 \leq \alpha \leq 0.85$, and $1 \leq f \leq 8$ are chosen in this example. In the training process, 200 signals

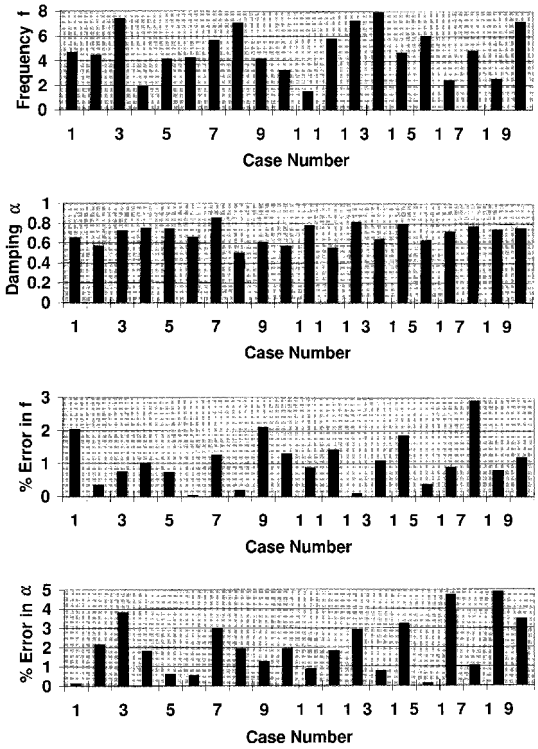

Fig. 3 Frequency and damping parameters and errors using a two-layer neural network.

with various frequencies and damping coefficients are used, and the values of α and f are selected from a random-number program. The test data set consists of 20 signals that are similar to those generated for the training set. The performance of using the network is shown in Fig. 3, where the predicted frequency and damping and their errors are shown. The results demonstrate the neural network can be trained to extract features from the signal with acceptable accuracy. Figure 3 shows that the errors for most of the cases studied are within 3% in predicted damping and 1.5% in frequency values.

It should be noted that the neural network developed here is based on supervised training. In the training process, not only input data sets are provided, but the correct outputs are also presented to the network. Consequently, the accuracy of the network will suffer if the ranges of the parameters in the test data fall outside the values used for the training sets. This limitation may be improved by expanding the training set to increase the ranges of parameters' values or by switching from supervised to unsupervised training, in which the network has the ability to learn on its own. Other improvements can be achieved by continuing refinement of the network, such as using more efficient optimization algorithms to accelerate the convergence rate in the training process, and investigating the effects of addition of neurons and hidden layers.

In summary, the neural network developed here has been successfully tested to extract frequency and damping parameters from a simulated exponentially decaying sine wave signal. A particularly attractive feature of the present approach is the ability to perform real-time parallel processing of multimode signals. For a given multimode signal, a wavelet package is used to decompose the signal into a sequence of single-mode signals. A program can be developed so that the present neural network is implemented in each processor of a parallel computer. Consequently, the frequency and damping parameters can be extracted in a parallel fashion. Further development is required to achieve this capability. The results suggest that the neural network can be used to analyze flutter signals. Flight test data,[2] which predominantly consist of two modes of the form given in Eq. (1), are being tested, and the results will be reported in a later paper.

References

[1]Walker, R., and Gupta, N., "Real Time Flutter Analysis," NASA CR-170412, 1984.

[2]Lee, B. H. K., and Laichai, F., "Development of Post-Flight and Real Time Flutter Analysis Methodologies," *Proceedings Forum International Aéroélasticité et Dynamique de Structures* (Strasbourg), Association Aéronautique et Astronautique de France, 1993, pp. 703–719.

[3]Daubechies, I., "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Transactions on Information Theory*, Vol. 36, No. 5, 1990, pp. 961–1005.

[4]Coifman, R., and Wickerhauser, M. V., "Wavelets and Adapted Waveform Analysis," *Proceedings of Symposia in Applied Mathematics*, Vol. 47, American Mathematical Society, Providence, RI, 1993, pp. 119–154.

[5]Chen, S., Billings, S., and Grant, P., "Nonlinear System Identification Using Neural Networks," *International Journal of Control*, Vol. 51, No. 6, 1991, pp. 1191–1214.

[6]Parlos, A., Atiya, A., and Sunkel, J., "Parameter Estimation in Space Systems Using Recurrent Neural Network," AIAA Paper 91-2716, Aug. 1991.

[7]Lee, B. H. K., and Jones, D. J., "Comparison of Two Methods for the Reduction of Free Decaying Data in Aircraft Flutter Tests," *Canadian Aeronautics and Space Journal*, Vol. 26, No. 4, 1980, pp. 322–335.

[8]Strang, G., and Nguyen, T., *Wavelets and Filter Banks*, Wellesley–Cambridge Press, Wellesley, MA, 1996.