AIAA JOURNAL
Vol. 32, No. 5, May 1994

Artificial Neural Networks for Predicting Nonlinear Dynamic Helicopter Loads

A. B. Cook,* C. R. Fuller,† W. F. O'Brien,‡ and R. H. Cabell§
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

The fatigue life of dynamic helicopter components is highly dependent on the history of loads experienced by the components during flight. However, practical methods of monitoring the loads on individual components during flight have not been developed. Current maintenance programs are characterized by frequent inspections and sometimes premature retirement of safety-critical components. This paper proposes using an artificial neural network (ANN) to predict the loads in critical components based on flight variable information that can be easily measured. The artificial neural network learns the relationship between flight variables and component loads through exposure to a database of flight variable records and corresponding load histories taken from an instrumented military helicopter undergoing standard maneuvers. Eight standard flight variables are used as inputs for predicting the time-varying mean and oscillatory components of the tailboom bending load and the pitch link load for seven flight maneuvers. The ANN predicts the mean and oscillatory components with accuracy ranging from 90.7% to 97.7% correct.

Nomenclature

d = desired ANN output
$\bar{E}$ = average percent-of-scale error
E = percent-of-scale error
H_n = output of hidden unit n
$\hat{H}_n$ = sum of weighted inputs to unit n
$W1_{mn}$ = weighted connection from input m to hidden unit n
$W2_n$ = weighted connection from hidden unit n to output unit
x_m = input from unit m
y = ANN output
$\hat{y}$ = sum of weighted inputs to output unit
η = learning rate

Introduction

THE fatigue life of a dynamically loaded helicopter component is dependent on the loads experienced by the component, which are in turn highly dependent on the flight history of the helicopter. A helicopter engaging in combat may experience excessive structural loads which severely reduce the operating life of safety-critical components.[1] This is particularly true for components located in the rotor system.[2] If the actual loads experienced by a component were known, the fatigue damage to that component could be estimated using standard methods.[3-5] Unfortunately, it is not currently feasible to mount instrumentation on each vehicle and on each component to measure flight loads. Hence, current helicopter maintenance programs are based largely on logged flight hours and an assumed flight history rather than the actual accrued fatigue damage. The resulting maintenance schedules are characterized by frequent inspections and possible premature replacement of some components. To reduce costs and improve reliability, a method for quantifying individual component fatigue damage according to the actual flight history is required.[2]

The method proposed in the current paper is to build a model of the relationship between flight variables and component loads. In this manner, the component loads that cannot be easily monitored are predicted using variables that can be easily monitored. The output of the model, either the predicted loads or fatigue damage accumulated during the flight, would be stored and used by maintenance personnel for more timely replacement of fatigue-sensitive components. Although systems of this type have been developed for fixed-wing aircraft, they have not been developed for helicopters due to the complexity of the relationship between flight variables and component loads in a helicopter.

An artificial neural network (ANN) is proposed as the framework for constructing a model of the relationship between flight variables and component loads.[6] ANNs have shown considerable promise for modeling complex nonlinear relationships.[7,8] A significant benefit of using an ANN-based model is its ability to learn relationships between variables with repeated exposure to those variables. Therefore, instead of using an analytical relationship derived from first principles to model a system, the neural network learns the relationship through an adaptive training process. The accuracy of the resulting model is then restricted only by the limitations of the neural network and the data base from which it learns the relationship.

The ANN framework for building a complex, nonlinear model is briefly explained in the first section of the paper. This is followed by a description of the database of flight records that were used to train and evaluate the ANN model. The results are given in the next section, followed by conclusions about the accuracy of the ANN model for predicting component loads.

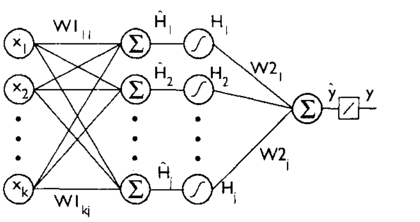

Fig. 1 Schematic of load-predicting ANN.

Presented as Paper 92-02-166 at the DGLR/AIAA 14th Aeroacoustics Conference, Aachen, Germany, May 11-15, 1992; received April 12, 1993; revision received Oct. 25, 1993; accepted for publication Oct. 25, 1993.

*Graduate Student, Mechanical Engineering Department; currently with Military Traffic Management Command, Transportation Engineering Agency, 720 Thimble Shoals Blvd., Newport News, VA 23606.

†Professor, Mechanical Engineering Department.

‡Department Head, Mechanical Engineering Department. Associate Fellow AIAA.

§Research Associate, Mechanical Engineering Department. Associate Fellow AIAA.

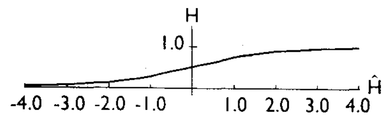

Fig. 2 Sigmoid function.

Analysis

The term artificial neural network encompasses a variety of computational methods, however, in this paper the term refers to a feedforward, backpropagation trained network. This is the most common type of neural network, probably because of its ability to form a minimum mean square error model of an input/output relationship.

The general configuration of the ANN is shown in Fig. 1. The operation of an ANN is analogous to that of a multiple regression model. That is, a set of independent variables, or inputs, are multiplied by a set of weighting coefficients to produce an estimate of the dependent variable, or output. In Fig. 1, the inputs are represented by the units on the left side of the figure, the weighting coefficients are denoted by the symbols $W1$ and $W2$, and the output is shown at the right side of the figure. Unlike a multiple regression model, the ANN carries out an additional series of computations between the inputs and the outputs. These computations take place in the hidden units, shown in Fig. 1, that contain nonlinear scaling functions. The hidden units allow the network to build a model of arbitrary complexity of the input/output relationship.[8] This model is developed during an adaptation process so as to minimize the mean squared error between the ANN output and the desired output. Computations in the ANN and the adaptation process are described in further detail hereafter.

Given a vector of input variables, where the nth such set of inputs is denoted

$$\mathbf{x}_n = [x_{n1}, x_{n2}, \ldots, x_{nk}] \tag{1}$$

the linear output of hidden unit i is computed as a weighted sum of the input values,

$$\hat{H}_i = \sum_{j=1}^{k} W1_{ji}x_{nj} + W1_{0i} \tag{2}$$

The weight $W1_{0i}$ is a bias offset, whose purpose is related to the nonlinear scaling function. The nonlinear scaling function, also called a sigmoid function, is applied to the linear summation $\hat{H}_i$, to produce a value between 0 and 1, as indicated by the following:

$$H_i = \frac{1}{1 + \exp(-\hat{H}_i)} \tag{3}$$

The sigmoid function is plotted in Fig. 2. Note that for small input values the output of the sigmoid is a nearly linear function of its input. However, for larger inputs, the output becomes more nonlinear. The application of the sigmoid function to the linear summation $\hat{H}_i$ produces an output that is a nonlinear function of the input variables. This operation allows the network to form the necessary nonlinear associations between the input and output variables. The bias variable $W1_{0i}$, in Eq. (2), affects the degree of nonlinearity by shifting the input away from the linear region of the sigmoid function.

The output of each hidden unit is then weighted and summed in the single output unit,

$$\hat{y} = \Sigma H_i W2_i \tag{4}$$

The linear summation $\hat{y}$ is not passed through a sigmoid function, but instead is multiplied by a constant factor, as indicated in Eq. (5) to produce the network output,

$$y = 0.25 \times \hat{y} \tag{5}$$

The factor 0.25, which corresponds to the slope of the sigmoid function for 0 input, is not critical and can easily be transposed onto the weights $W2$. The linear summation $\hat{y}$ is not passed through a sigmoid function because the output y must assume a continuous range of values. If the ANN were used for pattern recognition, then it would be useful to insert a sigmoid function at the output. That is, the desired network output for classification is binary; either a pattern belongs to a class or it doesn't. For the loads prediction problem, the output of the network assumes a continuous range of values, hence the sigmoid function is not needed.

The weights $W1$ and $W2$ in the ANN are determined by the backpropagation training procedure. Backpropagation training is an iterative procedure that implements a gradient descent of the error surface. At each presentation of input and output values, the training algorithm computes an incremental change to each weight to reduce the squared error between actual and desired output values. Although a complete derivation of the training algorithm is beyond the scope of this paper, it can be found in Refs. 8 and 9. The derivation involves a straightforward application of the chain rule, based on the assumption that the derivative of the error with respect to each weight is zero at the point of minimum error.

The weights in the ANN are initialized with values uniformly distributed in the range $[-0.3, +0.3]$ to differentiate the hidden units from one another.[8,9] Training data, consisting of input/output values, are then presented to the network. The output of the network, y, is computed according to Eqs. (2–5), and compared with the desired output to obtain an error,

$$\text{Error} = (d - y) \tag{6}$$

If the error is less than a specified tolerance no changes are made to the weights. If the error is greater than the tolerance, then the weights are adapted by propagating the output error back to each weight. That is, a weight connecting a hidden unit to the output unit is updated according to

$$W2_i^{t+1} = W2_i^t + \eta(0.25)(d - y)H_i \tag{7}$$

where the superscript $t + 1$ indicates the new value of the weight $W2$. The error is then propagated back to the previous layer of weighting coefficients,

$$W1_{ij}^{t+1} = W1_{ij}^t + \eta(0.25)(d - y)W2_j[H_n(1 - H_n)]x_i \tag{8}$$

Weights are updated after each input/output pair is presented to the network. One complete pass through the entire set of input/output values is referred to as one training iteration.

Because of the iterative nature of the backpropagation algorithm, the output error does not usually converge to an acceptable value until several hundred iterations through the entire set of training data are completed. This convergence time is dependent on the complexity of the problem, the number of input and hidden units, and the initial weight values. After the mean square error between the network

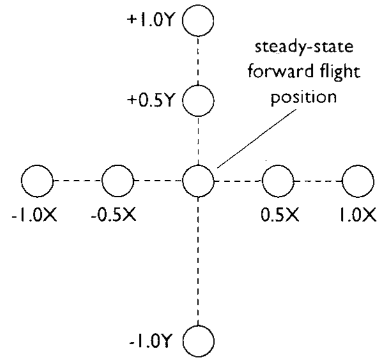

Fig. 3 Maneuvers used for ANN training and testing.

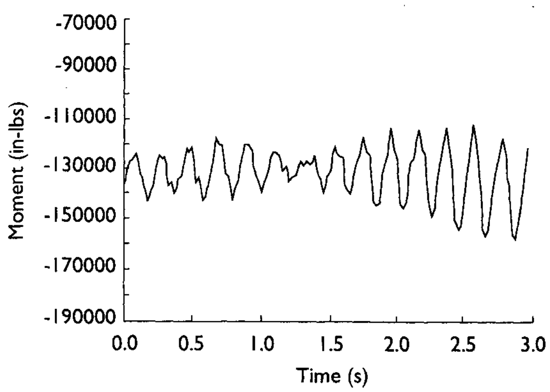

Fig. 4 Vertical tailboom bending load, $-0.5X$ maneuver.

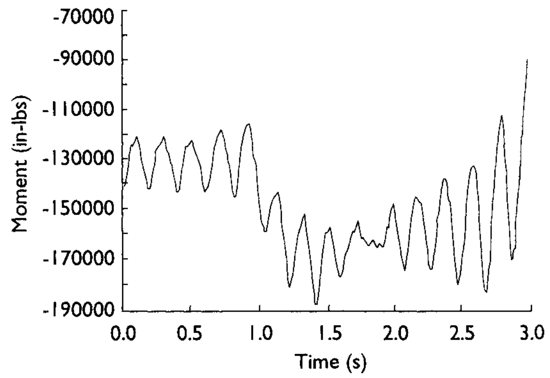

Fig. 5 Vertical tailboom bending load, $+1.0Y$ maneuver.

output and the desired output has reached an acceptable level, the weights are saved and are used for predicting loads from data sets not included with the original training data.

Although the neural network minimizes the sum of the squared differences between actual and predicted loads, the average percent of scale error will be used to compare the accuracy of prediction results. The output range of the network is [0, 1], so the average percent of scale error is computed as

$$\bar{E} = (100/N) \sum_{i=1}^{N} |d(i) - y(i)| \tag{9}$$

for N training samples.

Network parameters such as the training rate, number of hidden units, training time, and training tolerance were determined during initial training and testing. Each of these parameters has some effect on the performance of the neural network, however there is typically a large range of values of each that will produce good results. The leaning rate η in Eq. (7) was set to 0.4, and the network was configured with four units in the hidden layer. The training tolerance, discussed with Eq. (7), was set to 2%. The training time varied slightly depending on the initial weight values, but 500 iterations through the training set were usually sufficient to reduce the average percent of scale error [Eq. (9)] to 10%.

Helicopter Data

A database for developing the ANN for loads prediction was assembled from structural loads data measured on an AH-64 helicopter. These data were collected in a separate structural analysis of 17 safety-critical components and 21 fatigue-sensitive locations on the helicopter.[1] The vehicle was instrumented extensively to monitor flight variables and loads in fatigue-sensitive locations. Although the complexity and weight of the data acquisition system precludes its installation on the entire fleet of helicopters, the information obtained from this vehicle provides a unique database for developing and evaluating a loads prediction system.

The data consist of time history values of 24 flight variables and 17 component loads measured during seven standard flight maneuvers. Each maneuver consists of a step change in the directional control stick position from a steady-state forward flight position. The step changes in the data set are shown in Fig. 3. Each maneuver is labeled in Cartesian coordinates, where $+Y$ indicates the front of the aircraft and $+X$ indicates the right side of the aircraft. The magnitudes indicate inches of travel of the stick. For example, $+1.0X$ indicates that the stick was moved one-inch to the right during the maneuver. A one-inch displacement corresponds to about 15% of the stick position range. In subsequent sections of the paper, the one-inch displacements will be referred to as large stick displacements, while the half-inch displacements will be referred to as small stick displacements. Although the data cover a very limited portion of the entire flight spectrum of the helicopter, the information is still useful for this initial study of loads prediction on helicopters.

The ANN was trained to predict either the time-varying mean or time-varying oscillatory components of the tailboom bending load and the pitch link load. These loads were reduced to mean and oscillatory components because none of the monitored flight variables varied in a sinusoidal fashion that was directly correlated with the motion of the main rotor blades. Without such an input, there is no way for the network to produce a sinusoidal varying output. The time-varying mean value was computed from the raw data by averaging the data over a sliding time window. The result was a smoothed version of the load that preserved large changes in the level of the load. The oscillatory component of the load was calculated by subtracting the time-varying mean from the raw data, then computing the standard deviation of the load over a similar sliding time window. All mean and oscillatory data were scaled to the range [0, 1] based on their minimum and maximum values over the entire database of maneuvers. This normalization is done to minimize numerical difficulties when training the neural network due to the limited range of the sigmoid function.[8,9]

Only a small segment of data from each maneuver was used for this study. Each maneuver recording contained substantial amounts of data from steady-state flight that did not contain significant changes in either the flight variables or the component loads. Consequently, only two seconds of data centered about the change in stick position were used.

Example load histories of the tailboom bending load from small and large stick displacement maneuvers are shown in Figs. 4 and 5, respectively. The tailboom bending load is produced by vertical accelerations and changes in the pitch rate of the aircraft, and transient aerodynamic loads on the tailboom due to the orientation of the aircraft in forward flight. The vertical tailboom bending load ranged from $-190{,}000$ lbf to $-70{,}000$ lbf for the seven maneuvers consid-

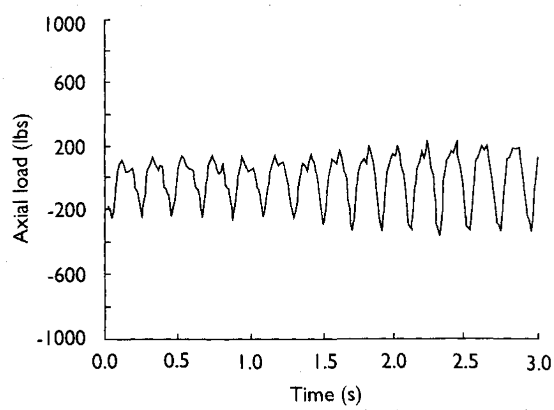

Fig. 6 Pitch link load, $-0.5X$ maneuver.

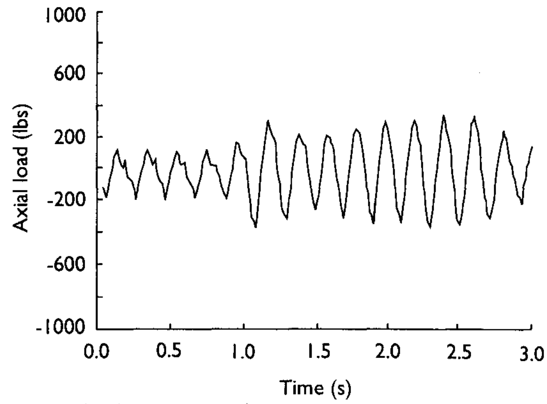

Fig. 7 Pitch link load, $+1.0Y$ maneuver.

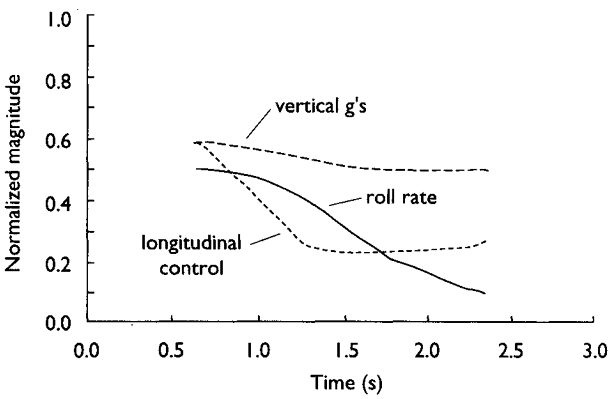

Fig. 8 Input data time histories, large stick displacement maneuver.

ered. A frequency analysis of the load reveals the principal cyclic components to be at 4.75 Hz and 19 Hz. The lower frequency, which is the more prominent of the two, corresponds to the rotation frequency of the main rotor. The component at 19 Hz is a result of the blade passage frequency of the main rotor, which is exactly four times 4.75 Hz.

The load histories in the pitch link for small and large stick displacements are shown in Figs. 6 and 7, respectively. The pitch link is located in the rotor system and controls the pitch of a main rotor blade. The range of loads for the seven maneuvers was approximately -400 to 400 lbf. In both Figs. 6 and 7, an oscillatory component at 4.75 Hz is quite prominent in the load history. The mean value of the pitch link load in the figures varies less than the mean value of the tailboom bending load in Figs. 4 and 5. This is most likely a result of the type of maneuvers included in the data base. For example, a forward stick input requires the pitch of each main rotor blade to change cyclically, as the blade rotates. This change is implemented through the pitch link, and results in the strong cyclic loading seen in Figs. 6 and 7. In contrast, a change in the mean value of the pitch link is produced by a change in the collective pitch of the blades. The collected data consist of only directional stick changes, not collective stick changes, hence the mean value of the pitch link load does not change dramatically in any of the maneuvers.

Eight input variables for the neural network were selected from the 24 measured flight variables. These inputs were chosen on the basis of an intuitive understanding of the relationship between ship motions, and loads in the component of interest. These input parameters were the following: pitch rate, roll rate, yaw rate, vertical acceleration, lateral acceleration, longitudinal acceleration, longitudinal control position, and lateral control position. A more complex study could be done to establish the adequacy of a set of inputs for a given output variable, but this was beyond the scope of the present work.

Three input variables are plotted in Fig. 8 for one of the large stick displacement maneuvers. The five other input variables did not change noticeably during this maneuver, and hence are not shown in Fig. 8 to avoid cluttering the plot. Values of the eight input variables at each time step were used to predict either the mean or oscillatory component of the load at the corresponding time. Note that the input variables were also scaled to the range [0, 1]. The data span a time segment slightly shorter than two seconds due to the length of the sliding window used in computing the mean and oscillatory values.

Results

The ANN was trained with input/output data from six maneuvers, then tested on the seventh maneuver. This procedure was repeated for each of the seven maneuvers in the data set to ensure that the ANN was generating an accurate model of the relationship between the flight variables and the load of interest as opposed to memorizing patterns in the training data. The network was initialized with randomly distributed weights for each set of training data and for each load parameter. Thus, each plot in this section represents a unique training and test set of data, and a unique set of weights in the ANN.

The flight variables and loads data from the chosen training maneuvers were assembled into a matrix of values for training the ANN. Each row of the matrix contained values of the eight flight variables at a time t, plus, either the mean or oscillatory value of the load at the same time t. The rows of the matrix were then placed in random order and presented to the ANN for training. Randomizing the input data speeds up learning in the ANN,[9] but would be unnecessary if the network took into account the time-varying nature of the load to improve prediction accuracy. Test data were presented to the network in the correct time order so the network prediction results could be plotted as a function of time.

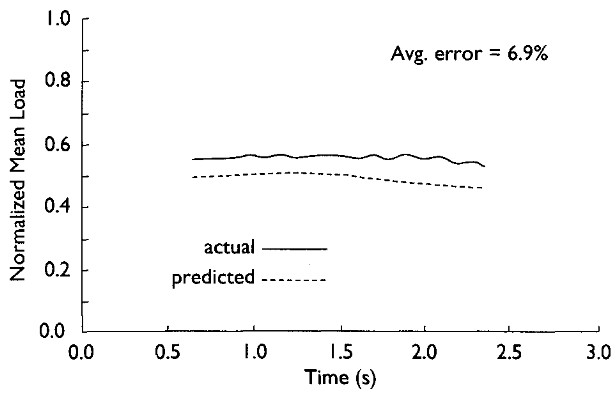

Fig. 9 Mean tailboom load prediction, $-0.5X$ maneuver.

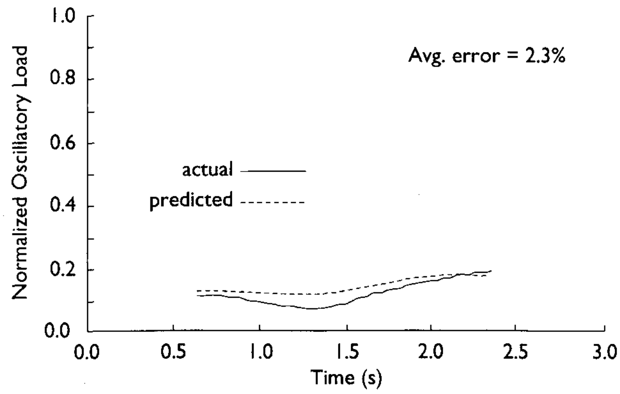

Fig. 10 Oscillatory tailboom load prediction, $-0.5X$ maneuver.

Of the seven maneuvers, the $-0.5X$ and $+1.0Y$ maneuvers are shown here for demonstration of the load predicting capabilities of the ANN. The time histories of the two loads of interest corresponding to the $-0.5X$ maneuver are shown in Figs. 4 and 6. Time histories for the loads from the $+1.0Y$ maneuvers are shown in Figs. 5 and 7. Note, from Fig. 3, that the expected range and direction of flight variables and loads for the $-0.5X$ maneuver should be encompassed by the data corresponding to the $-1.0X$ maneuver. That is, when the ANN predicts the loads for the $-0.5X$ maneuver, it will essentially be interpolating from the training data. In contrast, the ANN will be extrapolating from the training data when predicting loads for the $+1.0Y$ maneuver, because the training set of data extend only to $+0.5Y$. These two cases were chosen to investigate differences between the interpolating and extrapolating capabilities of the ANN.

Figure 9 shows the ANN prediction of the time-varying mean value of the tailboom bending load for the small stick displacement maneuver, $-0.5X$. The units of the load are plotted in normalized coordinates. The predicted value follows the measured value closely, but is consistently low. The average prediction error [see Eq. (9)] is 6.9%. The small sinusoidal variation in the actual mean in Fig. 9 is not present in the predicted mean. This small variation is a remnant of large cyclical variations in the load that were not removed when the data were preprocessed. Because the training data were presented in random order, any time-dependent variation, such as that in the small ripple, is perceived by the ANN to be noise during training.

ANN prediction of the time-varying oscillatory component of the tailboom bending load for the $-0.5X$ maneuver is shown in Fig. 10. The prediction follows the load more closely than the prediction of the time-varying mean value presented in Fig. 9. The average error $\bar{E}$ of the prediction in Fig. 10 is only 2.3%.

Predictions of the time-varying mean and oscillatory tailboom bending are shown in Figs. 11 and 12, respectively, for the large stick displacement maneuver, $+1.0Y$. In Fig. 11, the

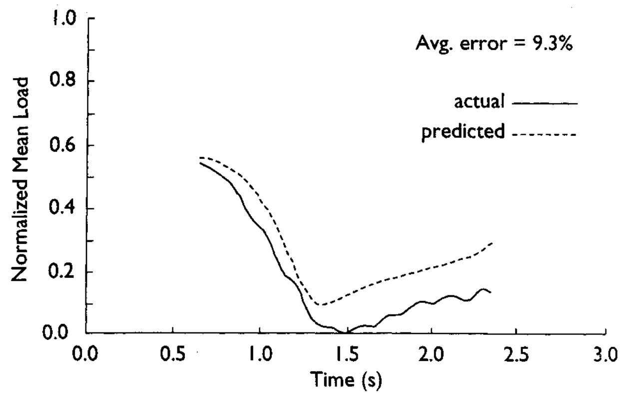

Fig. 11 Mean tailboom load prediction, $+1.0Y$ maneuver.

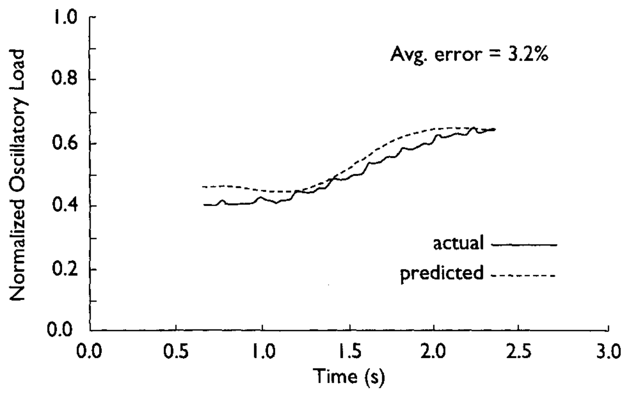

Fig. 14 Oscillatory pitch link load prediction, $-0.5X$ maneuver.

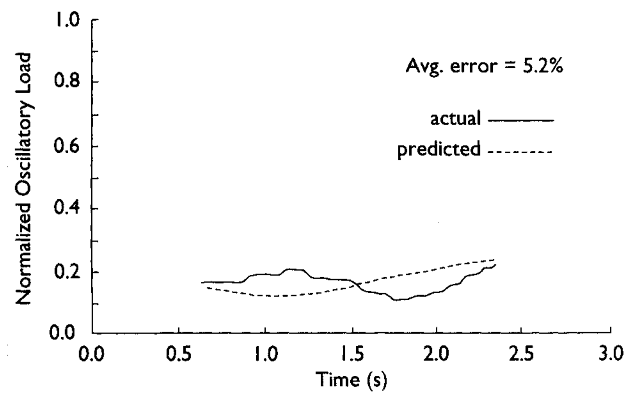

Fig. 12 Oscillatory tailboom load prediction, $+1.0Y$ maneuver.

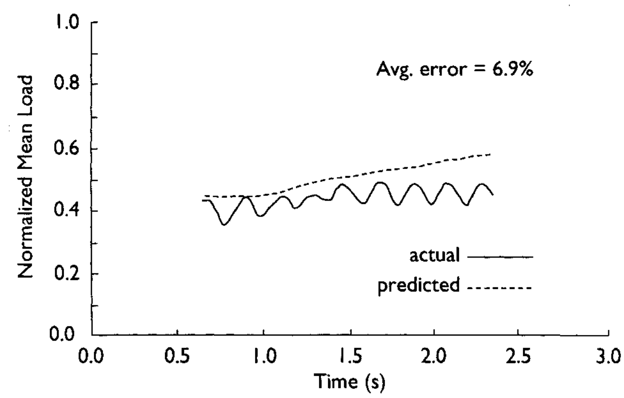

Fig. 15 Mean pitch link load prediction, $+1.0Y$ maneuver.

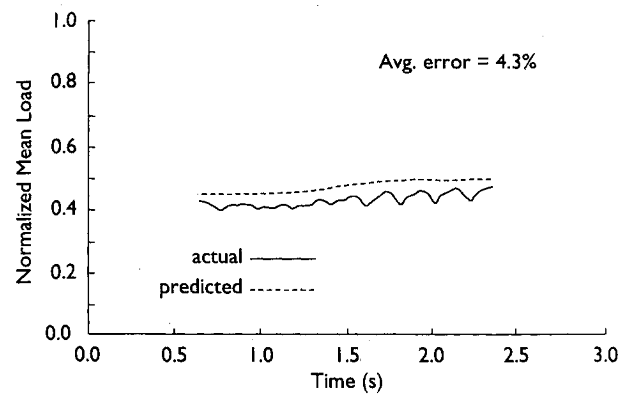

Fig. 13 Mean pitch link load prediction, $-0.5X$ maneuver.

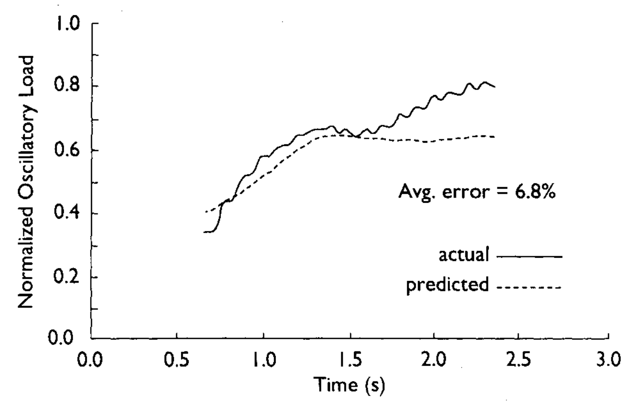

Fig. 16 Oscillatory pitch link load prediction, $+1.0Y$ maneuver.

ANN prediction of the time-varying mean bending load follows the downward trend in the actual data, although the ANN prediction bottoms out early and begins rising before the actual load begins to rise. The average error $\bar{E}$, in this case is 9.3%. In Fig. 12, the ANN prediction does not follow the slow variation of the oscillatory load, and the average percent of scale error is 5.2%.

The average errors for the tailboom bending load are greater for the large stick displacement maneuver than for the small stick displacement maneuver. This is most likely a result of the ANN having to extrapolate beyond the training data when predicting the loads for the large stick displacement maneuver.

Figure 13 shows the ANN prediction of the time-varying mean pitch link load for the $-0.5X$ stick displacement maneuver. The predicted mean follows the actual value closely, with only 4.3% average error. Once again, the sinusoidal variation in the actual time-varying mean is not seen in the predicted time-varying mean.

The actual and predicted values of the oscillatory component of the pitch link load for the $-0.5X$ maneuver are plotted in Fig. 14. The actual oscillatory component of the load rises during the maneuver, and the predicted value follows this rise quite closely. The average percent of scale error is only 3.2%.

Figures 15 and 16 illustrate the prediction of the mean and oscillatory components of the pitch link load for the large stick displacement maneuver, $+1.0Y$. The predicted mean value follows the actual mean with an average error of 6.5%. The sinusoidal variation is very apparent in the actual load, but was averaged out as noise during training, as was previously mentioned. The predicted and actual values of the oscillatory component in Fig. 16 are close to one another in the beginning of the maneuver, but diverge towards the end of the maneuver. The average error is 6.8%. The divergence of the predicted and actual oscillatory components in Fig. 16 illustrates the difficulty of predicting a load when it must be extrapolated from the training data.

Conclusions

An artificial neural network (ANN) has been used to predict time-varying loads on two fatigue-sensitive components in a helicopter using eight commonly monitored flight variables. The ANN model can be used to accurately monitor flight loads without requiring installation of bulky instrumentation on fatigue-sensitive components in the helicopter. It is postulated that the relationship between flight variables and component loads is nonlinear, hence the nonlinear modeling capabilities of the ANN are well-suited for this problem. The ANN learns the flight variable/component load relationship through an adaptive procedure involving exposure to measured flight data. A backpropagation training algorithm is used to adapt the parameters in the ANN to accurately predict component loads.

The predictive capability of the network was analyzed using data from seven maneuvers, each of which consisted of a change in the directional control stick position of either 0.5 in. or 1.0 in. from a steady-state, forward flight position. The ANN was trained with data from six maneuvers, then used to predict the load from the seventh remaining maneuver. This procedure quantified the ability of the ANN to accurately model the relationship between the input variables and the output load.

Prediction of the tailboom bending load and pitch link load were investigated. The ANN was used to predict the time-varying mean and time-varying oscillatory component of each load. Results from two representative maneuvers were shown; a 0.5-in. stick displacement maneuver and a 1.0-in. stick displacement maneuver. The average percent of scale error for the mean tailboom bending load prediction was 6.9% for the 0.5-in maneuver, and 9.3% for the 1.0-in. maneuver. Average error for the oscillatory component of the same loads was 2.3% and 5.2%, respectively. The average percent of scale error for prediction of the time-varying pitch link load was 4.3% and 6.5% for the 0.5- and 1.0-in maneuvers, respectively. Errors for the oscillatory components of the same loads were 3.2% and 6.8%, respectively.

Prediction errors increased on the 1.0-in. stick displacement maneuvers because the ANN was extrapolating from information in the training data set. Prediction is more accurate when the network can interpolate from the training data, as seen in the increased prediction accuracy for the loads from the 0.5-in. displacement maneuver.

Although these results cover a small portion of the helicopter flight spectrum, it is believed that similar agreement could be achieved for a larger envelope of maneuvers if the training data were similarly expanded.

Acknowledgments

The authors wish to acknowledge the financial support of this work, and the invaluable assistance of Randy Buckner, both from the Air Vehicle Structures Division of the US Army Aviation Applied Technology Directorate, Ft. Eustis, Virginia.

References

[1]Harrington, J., and Roesch, R. D., "Air-to-Air Combat Test IV (AACT IV) Volume II, AH-64 Structural Analysis," U.S. Army Aviation Systems Command TR-88-D-18B, Nov. 1989.

[2]Johnson, R. B., Martin, G. L., and Moran, M. S., "A Feasibility Study for Monitoring Systems of Fatigue Damage to Helicopter Components," U.S. Army Mobility Research and Development Lab. TR-74-92, Jan. 1975.

[3]Ryan, J. P., Berens, A. P., Coy, R. G., and Roth, G. J., "Helicopter Fatigue Load and Life Determination Methods," U.S. Army Mobility Research and Development Lab. TR-75-27, Aug. 1975.

[4]Gunsallus, C. T., Hardersen, C. P., and Stennett, P. G., "Investigation of Fatigue Methodology," U.S. Army Aviation Systems Command TR-87-D-17, May 1988.

[5]Arden, R. W., "Hypothetical Fatigue Life Problem," *Proceedings of the Specialists Meeting on Helicopter Fatigue Methodology Sponsored by the Midwest Region of the American Helicopter Society* (St. Louis, MO), March 1980, pp. 18-1–18-8.

[6]Cook, A. B., "Application of Neural Networks to Indirect Monitoring of Helicopter Loads from Flight Variables," M.S. Thesis, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, Oct. 1991.

[7]Obermeir, K. K., and Barron, J. J., "Time to get Fired Up," *BYTE Magazine,* Vol. 14, No. 8, 1989, pp. 217–224.

[8]Lippmann, R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine,* Vol. 4, No. 2, April 1987, pp. 4–21.

[9]Nelson, M. M., and Illingworth, W. T., *A Practical Guide to Neural Nets,* Addison Wesley, Reading, MA, 1991.