Journal of Sound and Vibration (1998) **209**(2), 329–342

NEURAL NETWORK MODELLING OF OSCILLATORY LOADS AND FATIGUE DAMAGE ESTIMATION OF HELICOPTER COMPONENTS

R. H. Cabell, C. R. Fuller and W. F. O'Brien

Vibration and Acoustics Laboratories, Mechanical Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, *U.S.A.*

(*Received* 4 *December* 1995, *and in final form* 21 *July* 1997)

A neural network for the prediction of oscillatory loads used for on-line health monitoring of flight critical components in an AH-64A helicopter is described. The neural network is used to demonstrate the potential for estimating loads in the rotor system from fixed-system information. Estimates of the range of the pitch link load are determined by the neural network from roll, pitch, and yaw rates, airspeed, and other fixed-system information measured by the flight control computer on the helicopter. The predicted load range is then used to estimate fatigue damage to the pitch link. Actual flight loads data from an AH-64A helicopter are used to demonstrate the process. The predicted load ranges agree well with measured values for both training and test data. A linear model is also used to predict the load ranges, and its accuracy is noticeably worse than that of the neural network, especially at higher load values that cause fatigue damage. This demonstrates the necessity of the non-linear modelling capabilities of the neural network for this problem.

1. INTRODUCTION

Efforts to apply on-line health monitoring to components in the rotor system of a helicopter have been hampered by cost and complexity constraints. Ideally, the loads in each flight-critical component would be monitored to ensure timely replacement of fatigue-damaged parts, however the cost of such an approach is prohibitive. A more reasonable approach that has been employed on fixed-wing aircraft is to monitor flight variables and estimate loads in the aircraft by using an appropriate model. Unfortunately, the complexity of the relationship between fixed-system variables and component loads on a helicopter makes the formulation of an accurate model problematic. As a result, current methods for predicting component fatigue damage are imprecise.

A common technique for fatigue damage assessment is to assume each helicopter follows a worse-case flight profile and then calculate the fatigue damage due to this profile. This approach is prone to error for aircraft that follow flight profiles significantly different from the assumed profile. These errors can be costly, due to either the premature replacement of parts or to the failure of a flight critical component.

Recent efforts to improve loads prediction include improvements to component loads models to allow the load to be synthesized from a limited set of fixed-system measurements [1–6]. One approach, described in reference [1], uses a transformation matrix to compute loads from control actuator forces and loads in the transmission support struts. The synthesized loads agreed well with actual loads, however instrumentation must be added to the helicopter to measure the loads in the transmission support struts.

0022–460X/98/020329 + 14 $25.00/0/sv971233

Another possibility for improving the model is to use a neural network, which can model non-linear complex relationships. The approach was described in references [2, 3, 5]. In references [2, 3], a feedforward neural network was used to predict the time-varying average load in the pitch link from measurements of fixed-system state variables. Measured data were used to train and test the neural network. The results obtained in these papers were promising, but the load predictions were not accurate enough to be used for fatigue damage estimation.

More recent work, carried out simultaneously and independently to this study, described the application of a neural network to the prediction of time-varying loads in three components of a helicopter [5]. A statistical analysis of the input parameters was used to quantify the importance of each on model accuracy and to help specify parameter combinations that might improve accuracy.

The approach taken here is similar to that described in reference [5]. The neural network used fixed system parameters as inputs to predict the range of load occurring in a component of the rotor system. For this work, the range of load occurring in the pitch link is studied. The output of the network is easily incorporated into a fatigue damage accumulation procedure on a cycle by cycle basis. Input parameters are limited to flight state variables monitored by the flight control system on the AH-64A. The accuracy of the neural network model is compared with that of a linear model, both in terms of load range prediction and the accuracy of the fatigue damage estimates computed from the output of each model.

The first section of the paper contains a brief description of the neural network, followed by a description of the helicopter flights in the database. The data preprocessing that was used to transform the flight data into a format suitable for the neural network is described next. In the Results section, comparisons are made between measured load ranges and neural network predictions of the load ranges. Fatigue damage is estimated from measured and predicted data to quantify the accuracy of the prediction method.

2. ANALYSIS

A feedforward neural network is used to model the relationship between fixed-system flight variables of a helicopter and the load in the pitch link. Inputs to the neural network consist of, for example, pitch, roll, and yaw rates, measured at the beginning of each loading cycle on the pitch link. The structure of the neural network and the training algorithm are discussed in the following paragraphs.

2.1. NEURAL NETWORK ARCHITECTURE

A feedforward neural network consists of inputs, hidden units, and output units, as shown in Figure 1. The inputs are denoted by the solid circles in the figure, and correspond to values of the fixed-system variables for a particular loading cycle. The hidden units are shown to the right of the inputs in the figure. A single layer of hidden units is shown, although any number of layers of hidden units could be used. The number of hidden layers, and the number of hidden units per layer, is a design parameter that depends on the complexity of the modelling problem. A single output unit is shown to the right of the hidden layer in Figure 1. Units in each layer of the network are connected to units in adjacent layers by adjustable weights.

The operation of the neural network can be explained by analogy with a first order polynomial model, written as

$$y = w_0 + w_1 x_1 + \cdots + w_p x_p = \sum_{i=1}^{p} w_i x_i + w_0, \tag{1}$$

where y is a linear combination of the inputs, x_i, weighted by the w_i's. The w_i's are calculated to minimize a cost function, usually defined as the sum of the squared differences between the model output, y, and the desired output. Equation (1) describes a linear neural network with a single output unit and no hidden layer; the w_i's connect the inputs to the output unit.

The polynomial model can be improved by adding hidden units, whose outputs are non-linear functions of their inputs, between the input and output units. The non-linear functions in the hidden units provide the neural network with non-linear modelling capabilities. The output, y_j, of the jth hidden layer unit is

$$y_j = f\left(\sum_{i=1}^{p} x_i wh_{ji} + wh_{j0} \right), \tag{2}$$

where the x_i are the inputs and the wh's are adjustable weights. The weight wh_{j0} is a bias weight, because it does not modify any input values; it only introduces an offset or bias in the summation term. In addition, f is a non-linear sigmoid function which is written as

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

and maps x values in the range $(-\infty, \infty)$ to (0, 1). Because of the limited dynamic range of the sigmoid function, the input/output data are normalized to (0, 1) to reduce numerical problems with the network weights [7].

With the addition of the hidden layer, the output of the output unit, which is equal to the output of the network, becomes

$$z = \sum_{i=1}^{q} y_i wo_i + wo_0, \tag{4}$$

where y_i is given in equation (2) and q is the number of hidden layer units.

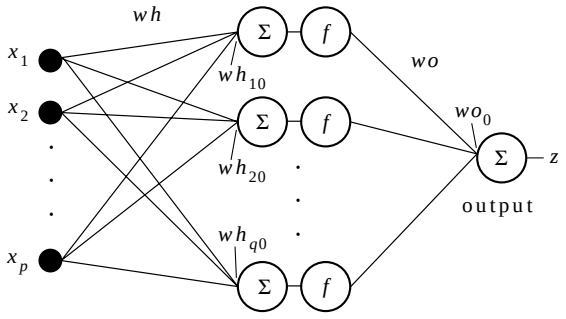

Figure 1. Structure of neural network.

The weights in the network, wh_{ij}, and wo_i, are calculated in order to minimize a cost function. The cost function is defined as the sum of the squared differences between the network output and the desired output. The cost, J, is thus

$$J = \frac{1}{n} \sum_{i=1}^{n} (z_i - d_i)^2, \tag{5}$$

where n is the number of training examples, z_i is the network output (from equation (4)), and d_i is the desired output for the ith training example.

Minimizing this cost function is difficult because of the hidden units. The output of each hidden unit is a non-linear function of its input, hence there is no closed form solution for the weights that minimize J. Instead, iterative techniques, called training algorithms in the terminology of neural networks, are used to determine the weights.

One of the most popular training algorithms is the back-propagation algorithm, which propagates the output error, $(z_i - d_i)$, back to each weight in the network (see references [8] and [9] for a description of the algorithm). A small correction based on the output error is calculated and applied to each weight in the network. This procedure continues through all training examples until the cost function is reduced to an acceptable level. The algorithm is simple and effective, but convergence can be very slow. The complexity of the cost function will also determine the rate of convergence.

The back-propagation algorithm is susceptible to convergence to a local minimum of the cost function. The chance of this occurrence is reduced if the network is trained several times with different random initial values of the weights each time. A comparison of the different solutions should highlight those that correspond to a local minimum.

Another important consideration is the number of training iterations that are needed to converge to a reasonable solution. Even though the error on the training examples decreases each time the network weights are adjusted, the network can begin to model characteristics of the training data that are not common to the true input/output relationship. The occurrence of this phenomenon, sometimes referred to as over-training, can be reduced with a uniform sampling of input/output data, but a cross validation technique should also be used. With cross validation, a second data set that is independent of the training set is used to monitor the accuracy of the neural network during training. If the error begins to increase on the second data set after initially decreasing, the network is beginning to over-train and training should be stopped.

It has been proven that a feedforward neural network with non-linear hidden units, such as that described here, can model functions of arbitrary complexity [10]. Hence, failure of a neural network to model an input/output relationship indicates that either the network is too small, the training time is too short, the training data are insufficient, or there is no deterministic relationship between the inputs and the outputs. Unfortunately, this result does not prove the existence of a training algorithm that will find the optimum network weights in a finite amount of time [10]. However, acceptable accuracy can be achieved in most applications using the back-propagation training algorithm.

A sufficiently sized network is almost guaranteed if a large number of hidden units are used, however there is a drawback to this approach. Each additional hidden unit and hidden layer increases the number of weight updates to calculate at each iteration, which increases the training time. In addition, the possibility of over-training the network increases as the degrees of freedom in the network (i.e., adjustable weights) increases. A practical procedure to determine the number of hidden units is to train and evaluate differently sized networks. The smallest network with acceptable performance should be used.

TABLE 1

Flights in database

Flight	No. of load cycles
Training data	
517-20	43
517-18	108
522-5	213
516-10	51
521-5	51
518-8	51
518-10	51
Test data	
513-7	235
517-11	80
522-25	51
521-21	51
518-7	51
517-7	51

3. EXPERIMENTAL DATA

Flight loads data from a fully instrumented AH-64A helicopter were used to train and test the neural network. In each flight, the pilot of the AH-64A was directed to follow a certain flight path until contact was made with a second aircraft. Once contact was made, the pilots positioned themselves to attack the other ship. As a result, the data consist of long periods of steady flight followed by maneuvering, and ending with steady flight [1, 11]. Only the maneuvering portions of each flight were used to train and test the neural network.

Flight-loads and fixed-system data were measured on 13 flights, listed in Table 1. Eight of the 13 flights involved theoretical fatigue damage to the pitch link (i.e., damage quantifiable by a cycle-ratio summation algorithm), as will be discussed later in the paper. No effort was made to insure that the selected flights adequately sampled the entire flight envelope of the aircraft, although the nature of the flights themselves insures a reasonably varied data sampling.

The flight loads data were preprocessed in order to extract relevant input and output values and to normalize the measurements. An example of the raw data from the helicopter is shown in Figure 2, which is the time history of the pitch link load from flight 517-20. This plot contains about 5600 data points. The loads data were reduced by computing the range of the pitch link load over each loading cycle. The length of each loading cycle was determined by applying a bandpass filter centered at 4.75 Hz with a 1.5-Hz bandwidth to the raw loads data. The output of this filtering operation was a sine wave corresponding to the rotation rate of the main rotor shaft. The sine wave was divided into whole periods, and then the negative to positive zero-crossing of each cycle was used to designate the beginning of a loading cycle. The range of the original pitch link load was then computed for each cycle.

The flights were arbitrarily split into training and test sets of approximately equal size. Data from seven maneuvers were used to train the network, whereas data from six maneuvers were used to test the network as shown in Table 1. Flight identification numbers are listed in the first column of these tables, and the number of pitch link loading cycles per flight are given in the second column.

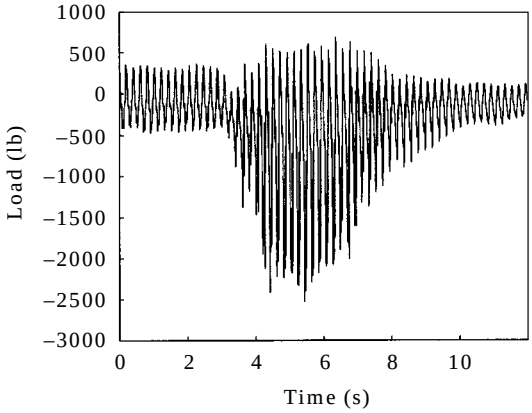

Figure 2. Pitch link load, flight 517-20.

Eleven fixed-system measurements were used as inputs to the neural network. The 11 variables, their maximum and minimum values, and the units for each, are listed in Table 2. These 11 were chosen not only to describe the state of the helicopter, but also because they are all monitored by the flight control computer on the AH-64A. A flight loads prediction system using these variables could be implemented by tapping into existing data bases on the helicopter.

The output of the network was the range of variation of the pitch link load over a single loading cycle. Load range and mean value variation are both used to calculate fatigue damage, but the load range typically changes faster than the mean value. Prediction of the actual load value at each sample time had been attempted in a preliminary study, but the approach was abandoned due to the inaccuracy of the predicted values.

Pitch link load range values for flight 517-20 are plotted in Figure 3. Several data points at the beginning of the flight were discarded in bandpass filtering the original load, hence there are fewer rotor revolutions in Figure 3 than in Figure 2. Separating the loads into loading cycles could be done on a helicopter by monitoring a position signal on the main rotor shaft.

The input variables changed more slowly than the pitch link load, thus only the value of each input at the beginning of a loading cycle was used. To include dynamic

TABLE 2

Input variables

Description	Min	Max	Units
Pitch rate	−50	50	deg/s
Roll rate	−100	100	deg/s
Yaw rate	−100	100	deg/s
Pitch actuator position	−2	102	%
Roll actuator position	−3	102	%
Yaw actuator position	0	100	%
Collective actuator position	−4	130	%
Airspeed	0	200	knots
Pitch attitude	−180	180	deg
Roll attitude	−180	180	deg
Yaw attitude	0	360	deg

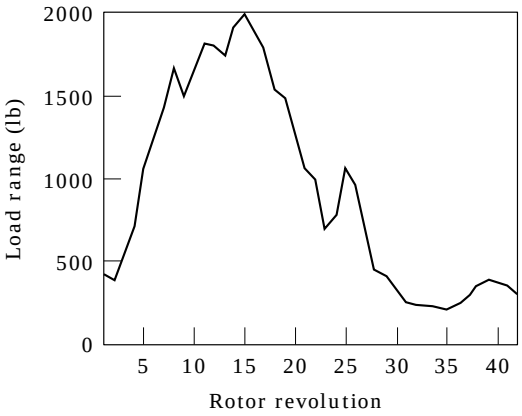

Figure 3. Pitch link load range, flight 517-20.

information, changes from the current value of an input variable to its value at one and two previous loading cycles were computed. This brought the total number of inputs to 33 (11 input variables plus 2 change values per variable). Dynamic information could also have been included by using the values of the flight variables at previous loading cycles as inputs to the network. However, by including the changes from one cycle to the next as inputs, the network did not have to learn that differences between certain inputs were significant.

Examples of the input variables for flight 517-20 are shown in Figures 4–7. Figure 4 is a plot of the pitch, roll, and yaw rates, in degrees per second. Figure 5 is a plot of the collective, pitch, roll, and yaw actuator positions, as a percentage of full range. Pitch, roll, and yaw attitudes are plotted in Figure 6, in degrees. The airspeed, in knots, is shown in Figure 7.

All input and output data were normalized to (0, 1) using the equation

$$z = \frac{x - min}{max - min}\,. \tag{6}$$

The values z and x are the normalized and original data values, respectively. The minimum

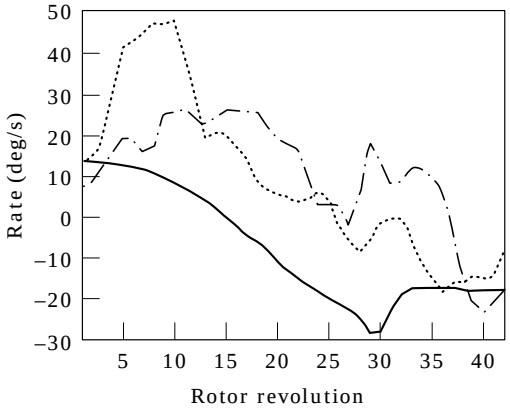

Figure 4. Fixed system rate inputs; ——, pitch; ---, roll: – - –, yaw.

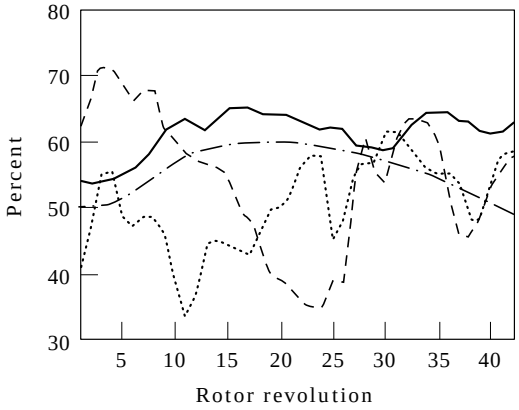

Figure 5. Fixed system actuator position inputs: ——, pitch; · · · ·, roll; – – –, collective; – - –, yaw.

and maximum values, listed in Table 2, were provided to the authors with the flight loads data.

After preprocessing, all data from a single maneuver was assembled into a matrix. Each row of the matrix corresponded to a loading cycle; each column contained values of the fixed-system variables and pitch link load range. For example, for flight 517-20, the matrix contained $43 - 2 = 41$ rows (the first two load cycles in a flight were not used because the change from the current input to the input at two previous cycles did not exist), and 34 columns.

4. RESULTS

A feedforward neural network with 33 inputs, 10 hidden layer units, and 1 linear output unit was used to predict load ranges. The network was trained several times; each time a different set of randomly initialized weights was used. The network usually converged to one of two solutions, which had nearly identical performance on the data in the test set.

The correlation between predicted and measured load range values was quantified with two parameters, which are explained with reference to Figure 8. The figure is a plot of

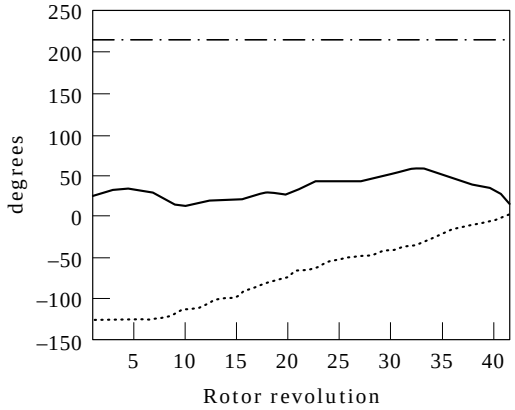

Figure 6. Fixed system attitude inputs: ——, pitch; - - -, roll; – - –, yaw.

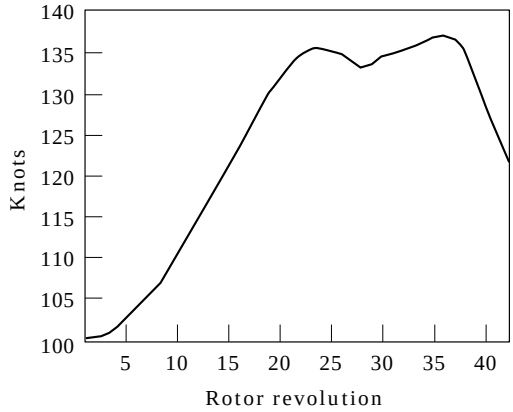

Figure 7. Fixed system input: airspeed.

measured versus predicted load ranges for flight 517-20. Each point in the plot is the predicted value (y-axis) corresponding to a measured value (x-axis). A dotted line has been fitted through the data points in a least mean squares sense. The solid line corresponds to the ideal case where the predicted values equal the measured values. The first parameter used to quantify prediction accuracy was the slope of the fitted line. A slope of 1 is an indication of accurate predictions, on average. The second parameter was the scatter of points about the fitted line. The scatter is quantified by the coefficient of multiple determination [12], which is

$$r^2 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}, \tag{7}$$

where $\hat{y}_i$ and y_i are the predicted and measured values, respectively, and $\bar{y}$ is the mean of

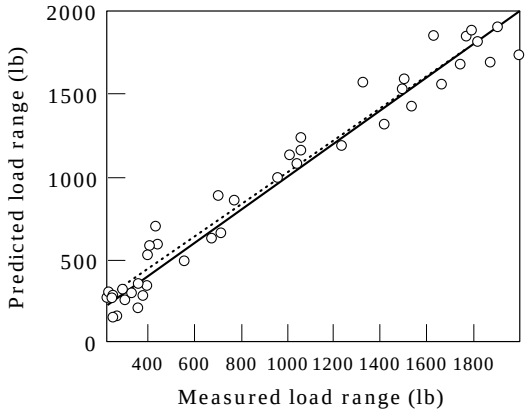

Figure 8. Measured versus predicted load ranges, flight 517-20: ○, load range; – – – –, actual best-fit line; ——, ideal best-fit line.

TABLE 3

Prediction accuracy

Flight	Neural network Slope	Neural network r^2	Linear model Slope	Linear model r^2
Training data				
517-20	0·98	0·96	0·66	0·90
517-18	0·95	0·93	0·77	0·61
522-5	0·94	0·94	0·58	0·41
516-10	0·93	0·86	0·39	0·54
521-5	0·96	0·93	0·91	0·91
518-8	0·94	0·95	0·63	0·92
518-10	0·90	0·96	0·68	0·84
Test data				
513-7	0·97	0·72	0·93	0·59
517-11	0·69	0·60	0·68	0·67
522-25	1·10	0·90	0·80	0·74
521-21	1·00	0·87	1·18	0.88
518-7	1·55	0·83	1·24	0·87
517-7	0·83	0·90	0·45	0·45

the measured values. An r^2 value of 1·0 corresponds to the case where all of the points lie directly on the dotted line; decreasing values correspond to increasing scatter.

4.1. PREDICTION ACCURACY

A plot of measured load ranges versus network predictions is shown in Figure 8 for one flight from the training set. In this figure, the dotted line (best-fitted line) falls almost directly on top of the solid line, which indicates that, on average, the neural network accurately predicted the load range. The r^2 value for the data in Figure 8 is 0·96. The neural network predicted the load ranges more accurately for data from this flight than for any other flight.

The dispersion of points about the dotted line in Figure 8 is a result of inaccuracies in the neural network model. These inaccuracies are due to either the lack of a deterministic relationship between the inputs and the output, or an insufficiently large network. Although training time can also affect the accuracy, it was found that increasing the training time tended to decrease prediction accuracy on test data.

Slope and r^2 values for all flights in the training set are listed in the top half of Table 3. The slopes ranged from 0·90 to 0·98, which are all close to the ideal value of 1·0. This indicates that the neural network accurately predicted the load range values, in the mean for both high and low values of the load range. The r^2 values ranged from 0·86 to 0·96, which are close to the ideal value of 1·0. The r^2 values indicate that the dispersion of points about the fitted lines was small for data in the training set.

Measured and predicted load ranges for two flights in the test set (i.e., the data not used to train the neural network) are shown in Figures 9 and 10, corresponding respectively to the most accurate and least accurate cases in the test data. The measured and predicted values agree closely in Figure 9; the slope of the best-fitted line is 1·10, and the r^2 value is 0·90. The measured and predicted data show less agreement in Figure 10. The best-fitted line has a slope of 0·69, and the r^2 value is 0·60. The low slope is an indication that the neural network model under-predicted large load range values, which are important for

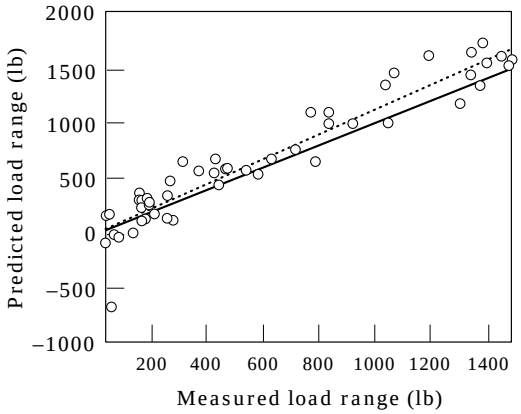

Figure 9. Measured versus predicted load ranges, flight 522-25: ○, load range; – – –, actual best-fit line; ——, ideal best-fit line.

estimating fatigue damage. The low r^2 value is probably a result of the dispersion of points near small load range values.

Slope and r^2 values for flights in the test set are listed in the bottom half of Table 4. The slope values, which ranged from 0·69 to 1·55, and r^2 values, which ranged from 0·60 to 0·90, represent a wider variation in prediction accuracy than that seen for the training data. The average slope, over all data from the test set, was 0·97, and the average r^2 value was 0·72. This is not as high as might be needed for precise fatigue damage prediction, but it does indicate that the neural network found a significant relationship between the fixed system measurements and the range of the pitch link load.

Results are also given for a linear model which is described in equation (1). The accuracy of the linear model is compared with the neural network to judge the degree of non-linearity in the problem. The training data were used to calculate the w_i's in equation (1). Slope and r^2 values for the linear model are given in the fourth and fifth columns of Table 3.

In general, the linear model was less accurate than the neural network model. Curiously, this contrast is more apparent for the training data. The slopes of the best-fitting lines for the linear model predictions are consistently lower than the corresponding slopes for the

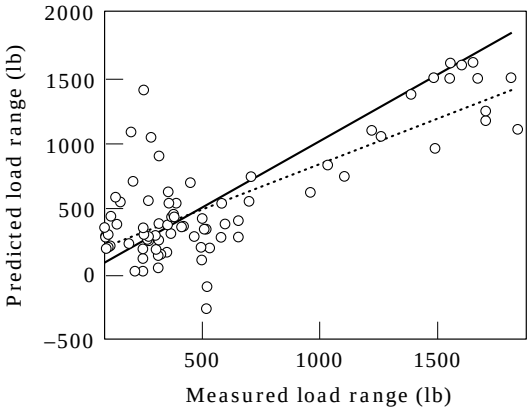

Figure 10. Measured versus predicted load ranges, flight 517-11: ○, load range; – – –, actual best-fit line; ——, ideal best-fit line.

neural network. One interpretation of this is that the linear model consistently under-predicted higher load range values in the training set. The r^2 values are also lower for the linear model, which indicates that for a particular best-fitting line, the scatter of points about the line was greater than for the neural network, and hence the linear model was less accurate.

The performance of the linear model was not as bad on the test data as on the training data, but again it was not as good as the neural network. In all cases but one (flight 521-21), the slope of the best-fitting line for the neural network model was greater than the corresponding slope for the linear model. On data from three of the six test flights, the r^2 values of the best-fitting lines for the linear model was greater than the corresponding values for the neural network. However, because the slopes for the linear model tended to be lower than for the neural network, the predictions were still inaccurate at higher load ranges. This decreased accuracy of the linear model at high load ranges is evident in the fatigue prediction results to be discussed in the next section, where accurate prediction of the high load ranges is important for accurate estimation of fatigue damage.

4.2. FATIGUE DAMAGE PREDICTION

Fatigue damage estimates for the pitch link were computed from measured loads data, neural network predictions, and linear model predictions. Fatigue damage was quantified using the Palmgren–Miner cycle-ratio summation theory, or Miner's rule [13, Sections 7–18]. Using this rule, the life of a part subjected to different loads for varying numbers of load cycles is assumed to be given by

$$\frac{n_1}{N_1} + \frac{n_2}{N_2} + \cdots + \frac{n_i}{N_i} = 1{\cdot}0, \tag{8}$$

where n_i is the number of cycles that a stress, σ, is applied to the material, and N_i is the life corresponding to that stress level. The N_i value for given σ was computed using a piecewise approximation to the S–N curve for AISI 4340 steel alloy. The right hand side of equation (8) indicates that failure is assumed to occur when the left hand summation is equal to 1·0. The fatigue damage estimates given here are values of the summation on the left hand side of equation (8).

None of the helicopter flights in the database contained load ranges that resulted in a significant amount of fatigue damage to the pitch link. As a result, the accumulated fatigue damage during a flight (i.e., the left hand side of equation (8)) is a tiny fraction of the fatigue life of the pitch link, which makes the damage estimate very sensitive to the few high loading cycles that occur in a given flight. Inaccurate prediction of these few critical loading cycles will greatly affect the accuracy of the cycle ratio summation computed in equation (8). For this reason, the summations given here should be taken as only a rough indication of the prediction capabilities of the models, since the results are so sensitive to only a few data values. A much larger database of load ranges would be needed for a more accurate comparison of the different models. Nonetheless, the fatigue damage estimates are helpful in interpreting the slope values listed in Table 3.

Damage estimates are listed in Table 4. The low cycle-ratio summation values in the table are a reflection of the low load range values in the database of flights. Using measured load range values, the summation was zero for five out of 13 flights in the database. The highest cycle-ratio summation computed from the measured data was 0·000022.

In general, the neural network predictions of the pitch link load were more accurate for fatigue prediction than the predictions from the linear model. When compared to cycle-ratio summations computed from the actual data, the neural network predictions resulted in low cycle-ratio summations of fatigue damage on four flights, but high

TABLE 4

Comparison of fatigue damage estimates

Flight	Cycle ratio summations		
	Measured	Neural net	Linear
Training data			
517-20	2·2 E − 5	1·9 E − 5	1·1 E − 6
517-18	1·1 E − 7	1·2 E − 7	0·0
522-5	0·0	1·3 E − 6	0·0
516-10	8·9 E − 6	4·1 E − 7	0·0
521-5	0·0	0·0	0·0
518-8	1·7 E − 5	2·1 E − 5	3·3 E − 7
518-10	6·8 E − 6	1·0 E − 6	0·0
Test data			
513-7	7·5 E − 6	5·5 E − 5	2·8 E − 6
517-11	6·1 E − 6	0·0	0·0
522-25	0·0	4·0 E − 7	0·0
521-21	0·0	0·0	8·4 E − 7
518-7	0·0	3·3 E − 5	0·0
517-7	2·2 E − 5	5·6 E − 7	0·0

cycle-ratio summations on six flights. On data from one flight, number 521-21, the linear model predicted a greater amount of fatigue damage than that predicted by the neural network. This was the one flight for which the slope of the best-fitting line for the linear model predictions was higher than the slope for the neural network predictions (see Table 3).

From Table 3, column 4, it is apparent that for a majority of the flights, the output of the linear model indicated than no fatigue damage occurred. This can be correlated with the low slope values seen in Table 3, column 4, since a low slope indicates an under-prediction of high load ranges, where fatigue damage occurs.

On the test data, the neural network over-predicted fatigue damage for flights 513-7, 522-25 and 518-7. One reason for the over-prediction is reflected in the slopes of the best-fitting lines, which were 1·10 and 1·55 for flights 522-25 and 518-7, respectively.

5. CONCLUSIONS

A neural network has been used for dynamic helicopter loads prediction using real data, where the inputs consist of readily obtainable fixed-system measurements. The neural network was used to predict in-flight values of the pitch link load on an AH-64A helicopter. Fixed system measurements such as roll, pitch, and yaw rates were used as inputs to the neural network. These inputs were selected because they are all monitored by the flight control computer of the helicopter, and hence require no additional instrumentation for their measurement. The neural network was trained and tested with data from several different flights of a helicopter flown in mock combat maneuvers. The prediction accuracy of the neural network was compared with that of a linear model. Estimates of fatigue damage were computed from the measured data and from the load predictions of the neural network and linear models.

Data reduction was accomplished by computing the range of the pitch link load over a single rotation of the main rotor. The load range is easily incorporated into a cycle-ratio summation analysis to determine fatigue damage to the pitch link. The neural network was

used to predict this load range given fixed system information at the start of the loading cycle and two previous loading cycles.

The neural network accurately predicted the pitch link load range on data from the seven flights used for training, with a tendency to under-predict the load range slightly, on average. The under-prediction had little effect on fatigue damage estimates, however. On three of the seven flights fatigue damage estimated from neural network predictions was low compared to measured data, and on three flights the estimated damage was high.

The prediction accuracy was more varied on data from the six flights not used to train the network. The neural network under-predicted the load ranges, on average, for three flights and over-predicted on two flights. Fatigue damage estimates from the neural network predictions were similarly high on three flights and low on two flights.

The linear model consistently under-predicted the loads when trained and tested on the same data as the neural network. The under-prediction was worst for high load range values where fatigue damage occurs. Correspondingly, the fatigue damage estimates on data from the linear model were low on all flights except one. The inadequacy of the linear model in predicting the load ranges validates the neural network solution.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the financial support of the Aviation Applied Technology Directorate, U.S. Army, Ft. Eustis, Virginia, for this work.

REFERENCES

1. C. T. Gunsallus and E. Robeson 1991 *AHS Structures Specialist Meeting*, *Williamsburg*, *VA*. AH-64A rotating load usage monitoring from fixed system information.
2. A. B. Cook, C. R. Fuller, W. F. O'Brien and R. H. Cabell 1992 *Proceedings of* 15*th Aeroacoustics Conference*, *Aachen*, *Germany*, AIAA Paper no. DGLR/AIAA-92-02-166.
3. A. B. Cook, C. R. Fuller, W. F. O'Brien and R. H. Cabell 1994 *AIAA Journal* **32**, 1072–1077. Artificial neural networks for predicting nonlinear dynamic helicopter loads.
4. D. J. Haas 1991 *AIAA/ASME/ASCE/AHS/ASC* 32*nd Structures*, *Structural Dynamics*, *and Materials Conference*, 710–718. Determination of helicopter flight loads from fixed system measurements.
5. D. J. Haas, J. Milano and L. Flitter 1995 *Journal of the American Helicopter Society* **40**, 72–82. Prediction of helicopter component loads using neural networks.
6. S. S. Tang and L. J. O'Brien 1991 *AIAA/ASME/ASCE/AHS/ASC* 32*nd Structures*, *Structural Dynamics*, *and Materials Conference*, 1357–1370. A novel method for fatigue life monitoring of non-airframe components
7. A. Lapedes and R. Farber 1988 in *Evolution*, *Learning*, *and Cognition* (Y. C. Lee, editor) 331–346. World Scientific. How neural nets work.
8. D. E. Rumelhart, G. E. Hinton and R. J. Willians 1986 in *Parallel Distributed Processing*: *Explorations in the Microstructure of Cognition*, *Volume* 1: *Foundations* (D. E. Rumelhart and J. L. McClelland, editors) 318–364. MIT Press. Learning internal representations by error propagation.
9. T. J. Sejnowski and C. R. Rosenberg 1987 *Complex Systems* **1**, 145–168. Parallel networks that learn to pronounce English text.
10. K. Hornik, M. Stinchcombe and H. White 1989 *Neural Networks* **2**, 359–366. Multilayer feedforward networks are universal approximators.
11. J. Harrington and R. D. Roesch 1989 *Air-to-Air Combat Test IV* (*AACTIV*) *Volume II*, *AH*-64 *Structural Analysis*, Technical Report TR-88-D-18B, US Army AVSCOM.
12. R. E. Walpole and R. H. Myers 1985 *Probability and Statistics for Engineers and Scientists*. New York: McGraw-Hill; third edition.
13. J. E. Shigley and L. D. Mitchell 1983 *Mechanical Engineering Design*. New York: McGraw-Hill.