Inverse Problems, Design and Optimization Symposium
Rio de Janeiro, Brazil, 2004

THE INVERSE DETERMINATION OF AERODYNAMIC LOADING FROM STRUCTURAL RESPONSE DATA USING NEURAL NETWORKS

Cheril Carn
School of Aerospace, Mechanical & Manufacturing Engineering, RMIT University, DSTO/ RMIT Centre of Expertise in Aerodynamic Loading, Melbourne, VIC, Australia
ccarn@optusnet.com.au

Prof. Pavel M. Trivailo
School of Aerospace, Mechanical & Manufacturing Engineering, RMIT University, DSTO/ RMIT Centre of Expertise in Aerodynamic Loading, Melbourne, VIC, Australia
trivailo@rmit.edu.au

ABSTRACT

The prediction and monitoring of aircraft structural fatigue damage is vital for the safe operation of ageing aircraft. The ability to determine aerodynamic loading inversely using structural response data has the potential to significantly improve fatigue monitoring capabilities.

This paper examines how a Neural Network can be used to estimate and predict aerodynamic loading from structural response data. To simulate aerodynamic loading conditions F/A-18 Empennage fatigue test data which includes the application of both high frequency buffet and low frequency manoeuvre loading will be used.

The neural network was trained using response data from several strain gauges as input and known applied loads as output. It was then tested with new data and compared to the known applied loading corresponding to the new data.

The network was also tested in its ability to predict the aerodynamic loading across locations different to the locations of the training data.

INTRODUCTION

Aerodynamic loads can currently be determined directly using expensive and complex instrumentation. But this is primarily during developmental stages and is not effective for ongoing monitoring of fleet operations. New methods of measuring pressure distributions are being developed such as pressure sensitive paints (PSP). However, a system that enables aerodynamic loads to be reconstructed using existing structural instrumentation would be invaluable.

Methods have been developed, which aim at mathematically calculating the aerodynamic loading inversely from structural response data. Shkarayev et al. (2000) developed an inverse interpolation formulation based on a parametric approximation of the aerodynamic loading [1]. This approach becomes very complex under real flight conditions.

Cao et al (1998) were also able to demonstrate that simple aerodynamic loading can be found inversely by using Artificial Neural Networks [2].

A software system, ALORENES, was developed by E Sofyan and P. Trivailo (2001) which uses a hybrid Neural Network – Finite Element model to solve aerodynamic load inverse problems [3]. ALORENES was successful in determining linear, non-linear, non-steady and combined aerodynamic loads. This system used FE models to obtain structural response data used to train the neural networks.

Structural Health Monitoring (SHM) Systems monitor structural response data in order to determine the fatigue damage incurred during operations so that effective maintenance can be applied. Instrumentation such as strain gauges and accelerometers are already used to determine the levels of overall flight loads, such as wing root bending moment and torque [4]. The current systems do not measure pressure distributions or aerodynamic loads on the aircraft. A better and continuous understanding of these loads during real flight conditions is useful for more accurate structural health monitoring.

Research has also been conducted in the area of utilising ANN methods in SHM systems. Troudet & Merrill (1990) proposed and demonstrated the use of ANN methods in the real-time estimation of fatigue life of components of reusable rocket engines [5], Solge (1994) used ANN techniques for recognising structural defects in composite materials [6], and Lopes (1997) showed that ANN methods can be used to monitor fatigue damage in off shore structures [7]. Clearly ANN techniques are a valuable tool in any SHM system.

This paper demonstrates the ability of neural networks to estimate the complex aerodynamic loading as found in actual conditions inversely using structural response data.

DESCRIPTION of the TEST CONDITIONS

The nature of aerodynamic loads experienced by an aircraft surface during operation can vary greatly. Types of loading to be expected include low frequency manoeuvre loading and high frequency buffet loads. Loading may be static or dynamic, linear or non-linear.

The air pressures over a wing or control surface can change quickly over time and at different surface locations.

In order to have a useful Neural Network system which can inversely estimate aerodynamic loading from structural response data, the training and testing data should cover the whole range of the expected operational loading conditions.

For this research data was taken from the International Follow-On Structural Test Project (IFOSTP) F/A-18 fatigue test conducted by the Royal Australian Air Force and Canadian Forces.

This fatigue test involved the simultaneous application of both manoeuvre and buffet loads using airbag actuators and shakers. The applied loads were representative of the actual loads experienced by an F/A-18 during flight tests.

From the large number of gauges installed on the test article a selection of 19 strain gauges along with 9 load actuators and 2 shakers were selected. These gauges are located on the port side vertical stabiliser and port side aft fuselage stub frames of the test article, and the selected actuators and shakers apply loads to the vertical stabiliser.

For these gauges and load channels, a data set of 200,000 data points per gauge/load channel was selected which correstponds to 5 minutes and 33 seconds of test time. This was then divided into a set of 160,000 data points which was used to train the neural network, and a data set of 40,000 data points which was selected to test the neural network.

The relationship between the strain gauges and the applied loads is complex. The gauges are placed within the aircraft structure primarily on the surfaces of spars and ribs at different depths and orientations.

The photograph in Figure 1 shows several gauges installed on an aft fuselage stub frame. The stub frame is attached to the vertical tail structure so the responses of gauges located on the stub frame will be affected by the loading on the vertical tail.

The 19 strain gauges locations are shown in Figure 2. Some locations have two or more gauges in close proximity.

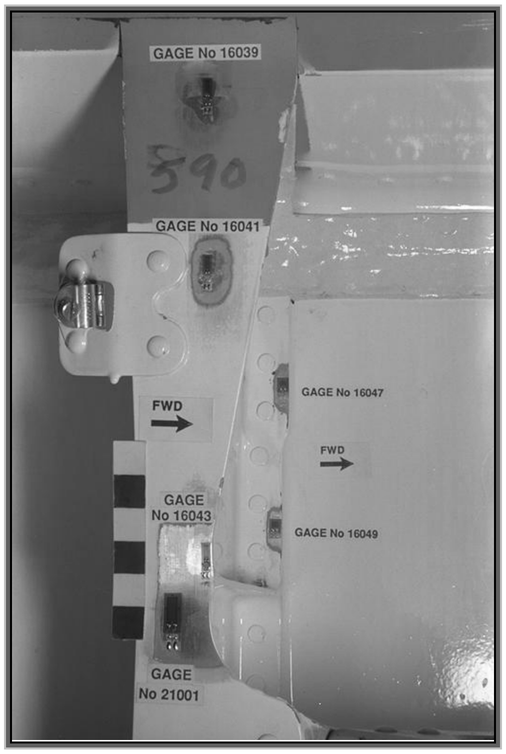

Figure 1. Strain gauges installed on the aft fuselage in the vicinity of the vertical tail.

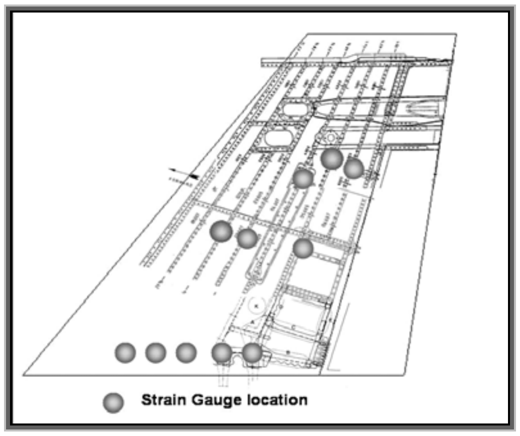

Figure 2. Locations of strain gauges used to train the neural network.

Figure 3 below shows the training input data taken from the 19 strain gauges. The data shows the gauge response to both low frequency manoeuvre loading applied by the actuators and the high frequency buffet loading applied by the shakers.

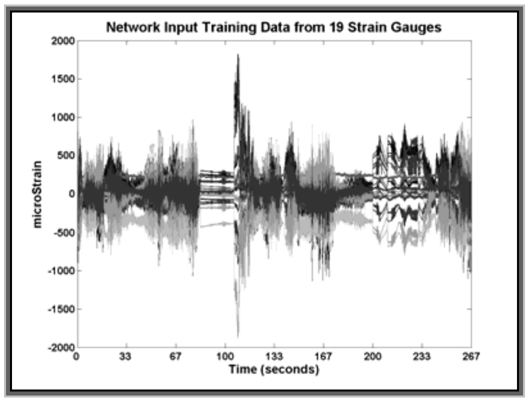

Figure 3. Strain gauge data used as input data to train the network.

In order for the neural network to be able to process the manoeuvre loading the buffet loading was first filtered using a MATLAB algorithm to separate the high frequency strain response data from the low frequency data. The actuator load data was not processed at all so that the loads data that the neural network output was tested against is still the actual data applied to the vertical tail. Only the strain response data was filtered. Figure 4 shows a graph of the original data from one gauge along with the data after filtering.

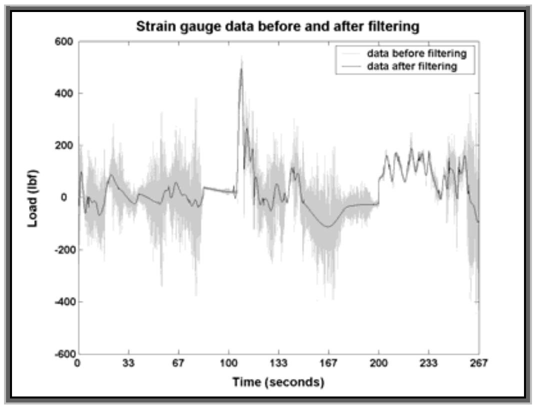

Figure 4. Strain data before and after filtering.

The filtered data was used to train the network when determining its ability to model the manoeuvre loading. The original unfiltered data was used to train the network when determining its ability to model the buffet loading,

The diagram in Figure 5 shows a simplified model of the relationship between aerodynamic loads and the structural response data on the aircraft vertical tail.

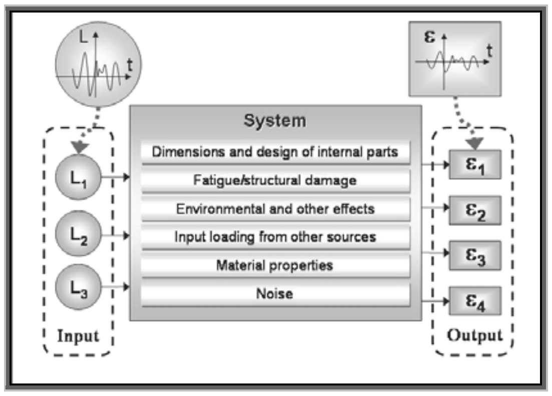
L
t
ε
t
System
Dimensions and design of internal parts
Fatigue/structural damage
Environmental and other effects
Input loading from other sources
Material properties
Noise
L_1
L_2
L_3
ε_1
ε_2
ε_3
ε_4
Input
Output

Figure 5. Model of relationship between aerodynamic loading and strain response.

In this paper the 'real' system of the tail structure, properties and other variables is replaced with the neural network which is used to simulate the inverse relationship as shown in Figure 6.

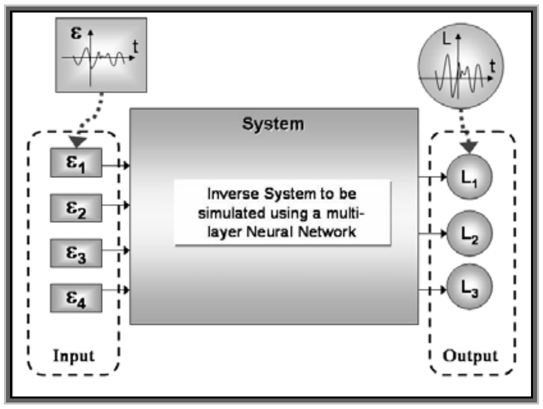

Figure 6. Model of inverse relationship between strain response data and aerodynamic loading.

This model assumes that there is a unique solution of the load values for the given strain values. This is achieved by using a set of strain values taken from gauges at different locations in the structure. When the number of gauges used as input increases the accuracy of the system can be improved.

This test shows that a neural network can be used to simulate the system without having to know the properties of the system.

DEVELOPING THE NEURAL NETWORK

The software used throughout this research is MATLAB version 6.5 by *The Mathworks, Inc.* with the MATLAB Neural Network Toolbox.

Initially, several common network types were considered for modelling the system, as listed in table 1. These network types are supported by the MATLAB software and descriptions may be found in Haykin (1994) [8].

Table 1. Common network types.

Name	Acronym
Generalized Regression Neural Network	GRNN
Radial Basis Network	RBNN
Probabilistic Neural Network	PNN
Cascade-Forward Neural Network*	CFNN
Elman Neural Network*	ENN
Feed-Forward Neural Network*	FFNN
Feed-Forward Time-Delay Neural Network*	FFTDNN
Hopfield Recurrent Network	HRN
Linear Neural Network	LNN
Learning Vector Quantization Neural Network	LVQNN

*Back-Propagation Network

Tests were conducted using a sample of input and output training and test data. The input data consisted of 10,000 data points for 19 gauges and one output load, with a test set of 2,000 data points. This resulted in an input training matrix with a size of 19 x 10,000 elements. Parameters and network structure was adjusted for network type to optimise performance. Most network types were found to be unsuitable for the task. A brief summary of results is shown in Table 2. It lists only networks that could form any kind of convergence during training.

Table 2. Summary of results for several networks.

Network Type	Training Time (min)	Error %	No of Epochs
CFNN	6:02	33	303
CFNN	3:19	8	313
ENN	7:38	7	399
FFNN	7:05	27	496
FFTDNN (Linear)	9:45	57	6
FFTDNN (Sigmoid)	11:32	145	6

Following further testing and comparisons between the Cascade-Forward and Elman back-propagation networks, a specially adapted Elman back-propagation network was selected to model the system and for testing with a larger training and testing data sets.

An Elman network with three linear layers was used. The first input layer contained 19 neurons (corresponding to the 19 input variables), a hidden layer of 8 neurons and a single neuron output layer. Each neuron contains a linear transfer function. Figure 7 shows the layout of the Elman back-propagation network used.

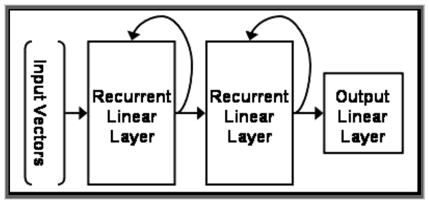

Figure 7. Layout of the selected Elman neural network.

The first layer in the selected Elman back-propagation network is a recurrent layer of linear neurons that accepts input from the network input vectors. The output from the first later is connected to the input to the second layer and is also fed back into itself with a delay factor.

The second layer also contains linear neurons that likewise has input which is a combination of the output of the first layer as well as its own output with a time delay factor. The output from the second layer is also fed to the output layer which contains a single linear neuron.

The output layer is not recurrent and its output produces the network output vector.

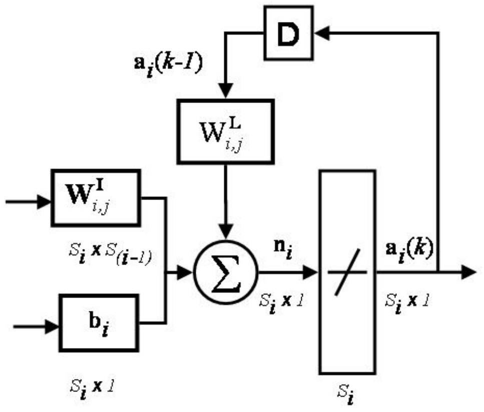

Figure 8. The layout of the first and second layers in the selected Elman network.

The architecture of the recurrent layers is identical and is depicted in Figure 8,

where $\mathbf{a}_i(k)$= the output vector of the ith layer and is dependent on time: k;

$\mathbf{W}^{\mathbf{I}}_{\mathbf{i,j}}$ = the vector of input weights for the ith layer and the jth input;

$\mathbf{W}^{\mathbf{L}}_{\mathbf{i,j}}$ = the vector of recurrent weights for the ith layer and the jth input;

$\mathbf{b}_i$ = the bias for the ith layer;

s_i = indicates the number of elements for the given layer;

D = the delay factor for the recurrent inputs.

The structure of the output layer is shown in Figure 9. The output layer is not a recurrent layer.

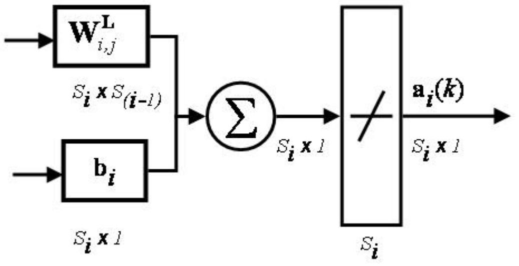

Figure 9. Diagram of the output layer in the selected Elman network.

Each of the recurrent linear layers (i = 1, 2) performs according to equation (1) shown below:

$$\mathbf{a}^i(k)=f((\mathbf{W}^{\mathbf{I}}_{\mathbf{i,j}}\,\mathbf{p}_i+\mathbf{W}^{\mathbf{L}}_{\mathbf{i,j}}\,\mathbf{a}_i(k\text{-}1)+\mathbf{b}_i)) \qquad \textbf{(1)}$$

where f is the linear transfer function, and $\mathbf{p}_i$ is the input at layer i.

The output layer (i=3) performs according to the equation (2) shown below, where f is the same linear transfer function

$$\mathbf{a}^i(k) = f((\mathbf{W}^{\mathbf{L}}_{\mathbf{i,j}}\,\mathbf{p}_i+\mathbf{b}_i)) \qquad \textbf{(2)}$$

For the first layer of the network (i = 1), the input is the input vector of strain values for j number of gauges for one time interval, as shown in equation (3):

$$\mathbf{p}_i = \varepsilon \qquad \textbf{(3)}$$

where ε is the input strain matrix

For the second and third layers of the network (I = 2, 3), the input is taken from the output of the previous layers as shown in equation (4):

$$\mathbf{p}_i = \mathbf{a}_{(i-1)}(k) \qquad \textbf{(4)}$$

Equation (5) represents the calculations of the Load at position j, $\mathbf{L}_j$, as a function of the input strain matrix ε.

$$\mathbf{L}_j = f((\mathbf{W}^{\mathbf{I}}_{\mathbf{3,j}}\Big[f((\mathbf{W}^{\mathbf{I}}_{\mathbf{2,j}}\Big[f((\mathbf{W}^{\mathbf{I}}_{\mathbf{1,j}}\,\varepsilon + \mathbf{W}^{\mathbf{L}}_{\mathbf{1,j}}\,\mathbf{a}_1(k\text{-}1)+\mathbf{b}_1))\Big] + \mathbf{W}^{\mathbf{L}}_{\mathbf{2,j}}\,\mathbf{a}_2(k\text{-}1)+\mathbf{b}_2))\Big]+\mathbf{b}_3)) \qquad \textbf{(5)}$$

The Elman back-propagation network was tested with various parameters and structures. It was found to perform better with linear layers. In trials conducted using one or two layers of neurons with the 'tansig' transfer function were not able to approximate the output data even when the number of neurons in each layer was varied. A network with only one recurrent layer and two layers in total was trialled but it produced less accurate results than the three layer network. When extra neurons were added to the first layer the training time was significantly increased. The inclusion of the second recurrent layer enabled greater accuracy with a lower increase in training time.

The network was trained using the MATLAB 'traingdx' function with is a gradient descent with momentum and adaptive learning rate back-propagation algorithm.

Other training methods were tried such as the Levenberg-Marquardt back-propagation method but they could not operate successfully due to the computer memory required to process the large input matrices. Even when the Levenberg-Marquardt back-propagation method was tried with one input vector and a training data set of only 1,000 data points the network could not converge.

TEST RESULTS

Network Performance in Estimating Buffet Loads

After 355 epochs the Elman network converged to a mean square error (MSE) of 4096.

The MSE for a network fitted perfectly to the training data would be zero, however considering the complex nature of the system involved and the large input training matrix it would be undesirable for the network to fit the data exactly. A non-zero MSE value indicates that the network is approximating the system, which is better if new data different to the training data is to be successfully estimated. To see how well the network has modelled the training data a sample of the same training data was fed back into the network and the output was compared to the actual output (load) as shown in Figure 10.

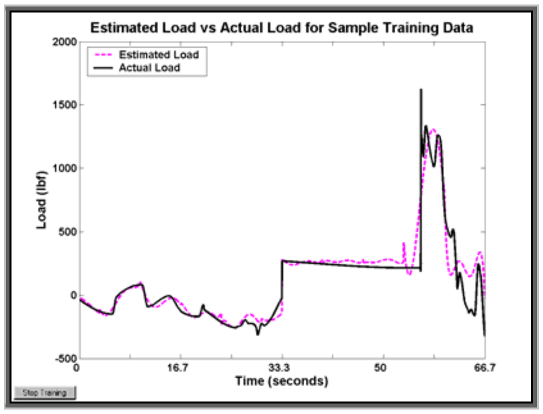

Figure 10. Estimated load compared with actual load for sample training data.

As Figure 10 shows the Elman network does not match the manoeuvre loading exactly. When the Elman network is tested with new data from strain response data not already in the input data set the network can still provide a reasonable approximation of the actual manoeuvre load, as shown in Figure 11.

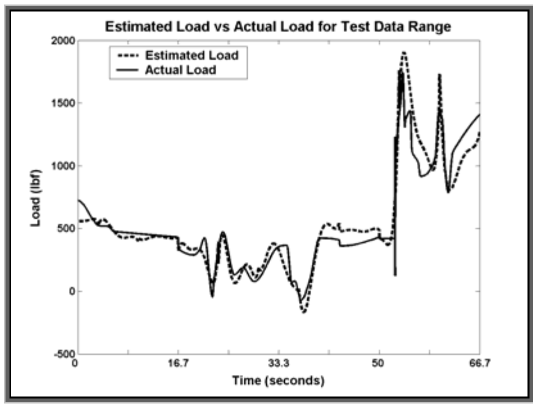

Figure 11. A comparison of estimated manoeuvre load to actual manoeuvre load.

As Figure 11 shows the network can estimate the manoeuvre loading at the location where the training loads data was recorded. The estimated load provides an indication of the magnitude and frequency of the actual load.

Network Performance in Estimating Buffet Loads

It is important that the network not only be able to determine the manoeuvre loads, but also the magnitude and frequency of the buffet loads. In this case the network was retrained using a subset of the same data set as was used for the manoeuvre load estimation, but without filtering. The matrix for the training input contained 20,000 data points for 19 gauges. The network parameters remained the same and after 980 epochs the network was able to model the training data as shown in Figure 12. The results using new test data are shown in Figure 13.

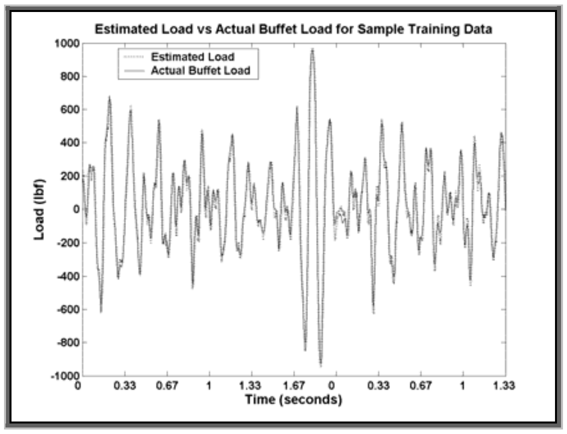

Figure 12. A comparison of estimated buffet load to actual buffet load for the training data.

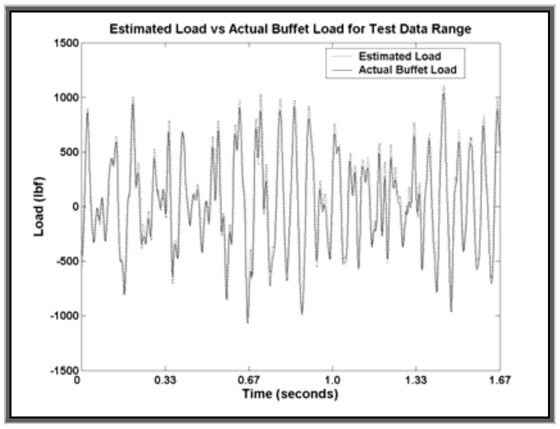

Figure 13. A comparison of estimated buffet load to actual buffet load for the test data.

The error in the test data range is slightly higher than that of the training data, but the results still show that the network is capable of estimating the high frequency dynamic buffet loading to a good level of accuracy.

Calculating Load at a New Location

The aerodynamic loading across an aircraft surface differs from the test conditions so far in that actual loading is distributed across the surface and may differ significantly at different locations. This exercise tests the networks ability to estimate the aerodynamic loading at a location different from the training data to determine whether the neural network can estimate the load at any point across the surface.

By learning how the position of the actuators influences the actuator load's influence on the strain response data, the network is able to estimate the load at a 'new' location outside of the training data.

The 'new' location is between the locations of the other actuators so that it is within the location range the network has been trained for.

Figure 14 shows the locations of the 9 load actuators on the vertical tail, along with the locations of three strain gauges.

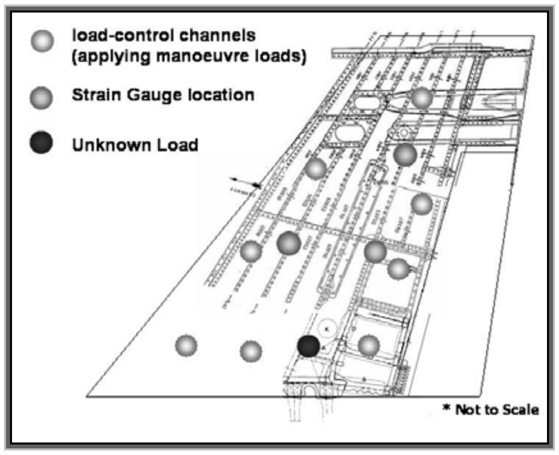

Figure 14. Locations of load actuators, strain gauges and 'unknown load'.

Each load L_1, L_2,…., L_n corresponds to the input loading applied at known locations 1,2,…,n corresponding to (X_1,Y_1), (X_2, Y_2),….,(X_n,Y_n) on the fin surface (assuming 2 dimensional surface).

If the coordinates for the locations of the loads are used as input in training the neural network then the network can also model the relationship between the location of the applied load and the resultant strain. So effectively the location of the load changes while the locations of the gauges is unchanged.

$$\text{Intput} = \begin{bmatrix} \mathbf{A_1} & \mathbf{A_2} & \cdots & \mathbf{A_i} \end{bmatrix}$$

$$\text{Where: } \mathbf{A_i} = \begin{bmatrix} \varepsilon_{1,1} & \varepsilon_{1,2} & \cdots & \varepsilon_{1,t} \\ \varepsilon_{2,1} & \varepsilon_{2,2} & \cdots & \varepsilon_{2,t} \\ \vdots & \vdots & \vdots & \vdots \\ \varepsilon_{n,1} & \varepsilon_{n,2} & \cdots & \varepsilon_{n,t} \\ \mathbf{x_i} & \mathbf{x_i} & \mathbf{x_i} & \mathbf{x_i} \\ \mathbf{y_i} & \mathbf{y_i} & \mathbf{y_i} & \mathbf{y_i} \end{bmatrix}$$

$$\text{Output} = \begin{bmatrix} \mathbf{B_1} & \mathbf{B_2} & \cdots & \mathbf{B_i} \end{bmatrix}$$

$$\text{Where: } \mathbf{B_i} = \begin{bmatrix} \mathbf{L_{i,1}} & \mathbf{L_{i,2}} & \cdots & \mathbf{L_{i,t}} \end{bmatrix}$$

In order to train the network to include actuator locations the original training data set was modified.

The input matrix consisted of the three gauge responses repeated and added to the location vector for each of the eight known load actuators over a 10 second period. The output is the sum of all eight load actuator outputs for the same time period. This allowed the network to learn the loading corresponding to the strain gauge data and the actuator position.

Figure 15 shows the networks ability to estimate the manoeuvre loading at a location outside of the training data set.

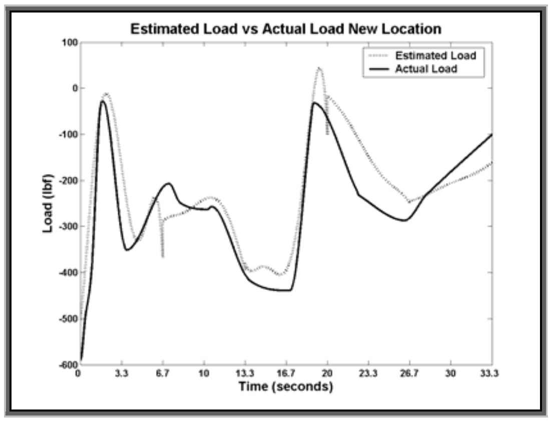

Figure 15. Estimated load at 'new location' compared to known actual load at that location.

CONCLUSIONS & FURTHER RESEARCH

Results show that neural networks are capable of estimating the aerodynamic loads inversely from the strain gauge data. The network cannot model an exact relationship between strain gauges and loads and the results contain a significant amount of error. The results are still useful however; because the actual relationships between response data and aerodynamic loading are complex and once trained the network has the potential to provide a good estimate of the aerodynamic loading. This may be suitable for structural health monitoring conditions where the exact load values are not as important as knowing the level of loading and frequencies.

Further work must be carried out to examine whether the errors can either be eliminated or to determine whether or not they have a significant effect on the potential applications of the network.

The selection of training data plays a large role in the ability of the network to accurately predict the aerodynamic loading. It is important that the network is trained with data that represents the full spectrum of expected loading conditions.

It would be better if the network could automatically identify and separate manoeuvre and buffet load components without prior filtering. More work could be done to modify the network or to combine more than one neural network in order to achieve this.

Overall, the results obtained so far are promising and neural networks do have the potential for enhancing existing structural health monitoring practices through the inverse determination of aerodynamic loads from structural response data.

ACKNOWLEDGEMENTS

This project could not have been carried out without the support of Defence Science Technology Organisation (DSTO), Australia which provided, along with financial assistance, access to data and technical resources required in order to carry out much of the work presented here and carried out by the DSTO/RMIT Centre of Expertise in Aerodynamic Loading (CoE-AL). In particular, authors would like to thank Dr. Douglas Sherman, Functional Head, Aircraft Loads from the Platform Sciences Laboratory for his vision, suggestions, support, initiative and thorough guidance of the work. The authors would also like to thank Dr. Chris Blanksby for helpful discussions.

REFERENCES

1. S. Shkarayev, R. Krashantisa, A. Tessler, An Inverse Interpolation Method Utilizing In-Flight Strain Measurements for Determining Loads and Structural Response of Aerospace Vehicles, *University of Arizona, Tucson, AZ 85721, & Analytical and Computational Methods Branch, M/S 240 NASA Langley Research Center, Hampton, VA 23681-0001, 2000.*

2. X. Cao, Y. Sugiyama, Y. Mitsui, Application of Artificial Neural Networks to Load Identification, *Computers and Structures,* **69**, 1998, pp. 63-78.

3. E. Sofyan and P. M. Trivailo, Solving Aerodynamic Load Inverse Problems Using a Hybrid FEM - Artificial Intelligence*, 3rd Australasian Matlab Users Conference [organised by CEANET and MATHWORKS], Crown Towers, Melbourne, Australia*, 09-10 Nov. 2000. - 19 pp.

4. L. Molent and S. Inan, Specifications for an Unified Strain and Flight Parameter Based Aircraft Fatigue Usage Monitoring System, *DSTO International Conference on Health and Usage Monitoring, Melbourne*, 19-20 Feb. 2001, pp. 53-67.

5. T. Troudet and W. Merrill, A Real Time Neural Net Estimator Of Fatigue Life Neural Networks, *IJCNN International Joint Conference*, vol. 2, 1990, pp. 59 -64.

6. D. A. Sofge, Structural health monitoring using neural network based vibrational system identification**,** *Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems*, pp. 91 -94.

7. T. A. P. Lopes, Neural networks on fatigue damage prediction**,** *International Conference on Neural Networks*, vol. 1, 1997, pp. 183-187.

8. S. S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed., Prentice Hall, New Jersey, USA, 1999. – 842 pp.