DOT/FAA/AR-01/80

Office of Aviation Research
Washington, D.C. 20591

Improved Methodology for the Prediction of the Empennage Maneuver In-Flight Loads of a General Aviation Aircraft Using Neural Networks

December 2001

Final Report

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof. The United States Government does not endorse products or manufacturers. Trade or manufacturer's names appear herein solely because they are considered essential to the objective of this report. This document does not constitute FAA certification policy. Consult your local FAA aircraft certification office as to its use.

This report is available at the Federal Aviation Administration William J. Hughes Technical Center's Full-Text Technical Reports page: actlibrary.tc.faa.gov in Adobe Acrobat portable document format (PDF).

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/AR-01/80		

4. Title and Subtitle	5. Report Date
IMPROVED METHODOLOGFY FOR THE PREDICTION OF THE EMPENNAGE MANEUVER IN-FLIGHT LOADS OF A GENERAL AVIATION AIRCRAFT USING NEURAL NETWORKS	December 2001
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
David Kim and Laure Pechaud	

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Embry-Riddle Aeronautical University 600 S. Clyde Morris Blvd. Daytona Beach, FL 32114	RPD-510
	11. Contract or Grant No. 98-G-025

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department of Transportation Federal Aviation Administration Office of Aviation Research Washington, DC 20591	Final Report
	14. Sponsoring Agency Code ACE-110

15. Supplementary Notes

The FAA William J. Hughes Technical Center Program Monitor was Mr. Thomas DeFiore.

16. Abstract

This research presents and documents an improved methodology for the prediction of empennage maneuver loads of a general aviation aircraft. Previous research using Neural Networks and angular accelerometers achieved predictions of strains to within ±50με of the measured strains in 100% of the cases for the vertical tail and 93% for the horizontal tail. The improved methodology is based on maneuver recognition using Neural Networks with the focus on both cost saving and improving the horizontal tail predictions. The recorded data are first classified by data clusters corresponding to known maneuvers then analyzed using Neural Networks trained for those maneuvers. This weighted sum approach successfully predicted strains to within ±50με of the measured strains in 100% of the cases for the horizontal tail. Angular accelerometer signals were replaced by numerically differentiated rate-gyro signals for the Neural Networks, and the predictions were found to be comparable resulting in considerable cost savings for the required minimum instrumentation.

17. Key Words	18. Distribution Statement
Flight loads, Maneuver recognition, Rate gyros, Angular accelerometers, Empennage, Flight recorder, General aviation	This document is available to the public through the National Technical Information Service (NTIS) Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	36	

TABLE OF CONTENTS

		Page
EXECUTIVE SUMMARY		vii
1.	INTRODUCTION	1
1.1	Overview	1
1.2	Previous Research	2
1.3	Current Approach	2
2.	BACKGROUND THEORY	3
2.1	Basis of Neural Network Architecture	3
2.2	Network Operation	4
2.3	Backpropagation Neural Network	5
2.3.1	Backpropagation Architecture	5
2.3.2	Backpropagation Training Algorithm	6
2.4	Unsupervised Neural Network, Self-Organizing Maps	8
2.4.1	Self-Organizing Map Architecture	8
2.4.2	Self-Organizing Maps Training Algorithm	9
2.5	Learning Vector Quantization Neural Network	10
2.5.1	Learning Vector Quantization Architecture	11
2.5.2	Learning Vector Quantization Training Algorithm	11
2.5.2.1	LVQ1	11
2.5.2.2	Other Versions: LVQ2, LVQ2.1, and LVQ3	12
2.6	Training Parameters	12
3.	TESTING AND DATA PROCESSING	13
3.1	Experimental Setup	13
3.2	Data Collection	14
3.2.1	Collected Data and Testing Procedure	14
3.2.2	Preparation of Data for Analysis—Data Filtering	15
4.	DISCUSSION OF RESULTS	15

4.1 Fatigue Data Consideration 15
4.2 Results From Suppression of the Angular Accelerometer 17

4.2.1 Utilization of the Signal from Rate Gyroscopes 17
4.2.2 Findings 19

4.3 Results From Improvement of the Predictions by Maneuver Classification 19

4.3.1 Neural Networks Specific to Each Maneuver 20
4.3.2 Classification 20

4.3.2.1 Evaluation of the Results 20
4.3.2.2 Classification Algorithm 20
4.3.2.3 Findings 21

5. CONCLUSIONS AND RECOMMENDATIONS 23

6. REFERENCES 24

APPENDIX A—NEURAL NETWORKS TESTED

LIST OF FIGURES

Figure		Page
1	Typical Architecture of a Network	4
2	Supervised Training	5
3	Backpropagation Neural Networks, Fully Interconnected	5
4	Principle of Backpropagation Training	6
5	Self-Organizing Map Architecture	8
6	Connections From the Input Layer to a Hidden PE *Yh*	9
7	Principle of SOM Training	10
8	Instrumentation Setup	13
9	S-N Curve of an Al 2024-T3 Sheet	16
10	Horizontal Tail Load During Maneuvers	17
11	Angular Accelerometer Signals	18
12	Rate-Gyro Signals After Numerical Differentiation	18
13	Classification Algorithm	21

LIST OF TABLES

Table		Page
1	Instrumentation Description	13
2	Minimum and Maximum Strain Increments Recorded	14
3	Sample of Neural Networks Predictions Trained on Rate-Gyro Signals After Numerical Differentiation	19
4	Parameters Used for the NN of Each Maneuver	20
5	Summary of Classification	22
6	Summary of Predictions After Classification	22
7	Summary of Predictions Previously Published in Reference 3	23
8	Summary of Classification and Prediction for Each Maneuver	23

LIST OF ABBREVIATIONS

α	Learning rate
β	Frequency estimation
γ	Conscience factor used to determine bias
δo & δh	Correction factors for output and hidden processing elements
vih	Weights of the Neural Network connections between input and hidden layers
ωho	Weights of the Neural Network connections between hidden and output processing elements
CG	Center of gravity
CG_Nz & CG Ny	Normal and lateral accelerations at center of gravity
DRL	Dutch roll maneuver
F	Transfer function
KIAS	Indicated airspeed in knots
NN	Neural Networks
PE	Processing element
PPL	Push-pull maneuver
r	Learning coefficient ratio
SOM	Self-organizing map
SSL & SSR	Side slip left & right
STGL & STGR	Stabilized level turn left & right
Tail_Nz & Tail Ny	Normal and lateral accelerations at empennage section
Xi	Input processing elements
Yh	Hidden processing elements
Zo	Output processing elements

EXECUTIVE SUMMARY

This research presents and documents an improved methodology for the prediction of empennage maneuver loads of a general aviation aircraft. Previous research using Neural Networks and angular accelerometers achieved predictions of strains to within ±50με of the measured strains in 100% of the cases for the vertical tail and 93% for the horizontal tail. The improved methodology is based on maneuver recognition using Neural Networks with the focus on both cost saving and improving the horizontal tail predictions. The recorded data are first classified by data clusters corresponding to known maneuvers then analyzed using Neural Networks trained for those maneuvers. This weighted sum approach successfully predicted strains to within ±50με of the measured strains in 100% of the cases for the horizontal tail. Angular accelerometer signals were replaced by numerically differentiated rate-gyro signals for the Neural Networks, and the predictions were found to be comparable resulting in considerable cost savings for the required minimum instrumentation.

1. INTRODUCTION.

1.1 OVERVIEW.

With recent improvements in flight data technology, airframe maintenance based on individual aircraft tracking has become feasible. For aging aircraft, this could potentially increase aircraft safety while keeping operating costs down. Aircraft operated regularly under severe conditions, such as low altitude surveying, patrolling, and flight training, would be checked in time to avoid unexpected failures, while airplanes flown in a much less severe environment would benefit from less frequent inspection cycles. The knowledge of the loads experienced by each aircraft is, therefore, essential to individual aircraft tracking. This would provide the information necessary to compile accurate fatigue load spectra and flight profile information for the aircraft. Future airplane designers could refer to this database to better estimate operational loads encountered in service and accurate predictions of airframe fatigue life could be made. If necessary, designers could then modify the design or require specific inspection cycles to ensure airworthiness of the aircraft.

Strain gages provide a direct method of measuring structural loads. However, the installation of strain gages on aircraft already in service is labor intensive and, most likely, cost prohibitive for many small aircraft operators. An alternate method in which instrumentation can be easily installed in a remote location, such as within the airplane center of gravity (CG) envelope, or in a readily accessible space, such as the baggage compartment, which can provide equivalent data, seems more practical. Thus, a methodology to estimate structural loads using remote sensors with the desired accuracy comparable to strain gages needs to be developed.

While extensive research has been completed on aircraft wing loads, work on empennage loads is still rather scarce. Since 1993, DeFiore and Kim [1] have been working on developing a methodology for the prediction of the empennage in-flight loads of small aircraft so that an appropriate flight loads database can be compiled. Initial phase of this work was to predict the strains in horizontal and vertical tail spars and verify these with strain gage measurements installed in the aircraft. Once the method has been validated, predictions can be extended to other locations of the structure including panels. The predicted strains can then be used to calculate the structural loads using traditional methods such as Skopinski, et al. [2]. The instrumentation must be simple, rugged, easy to install and maintain, and inexpensive in order to be accepted by the general aviation community.

In recent years, there have been good advances in the prediction methodology [3], but further improvement in accuracy of the predictions is sought particularly for the horizontal tail. Previous work treated all the data collected during maneuvers with a single Neural Network for vertical tail and for horizontal tail. In this study, the classification of the data by maneuvers is sought for improving the accuracy of predictions. If maneuvers can be identified from the data, then a network specifically developed for that maneuver can be used to achieve better accuracy of prediction. Only empennage maneuver loads are addressed in this report.

1.2 PREVIOUS RESEARCH.

The most comprehensive project to date on maneuver and gust loads on general aviation aircraft is the National Aeronautic and Space Administration and Federal Aviation Administration (NASA/FAA) VGH study [4]. The VGH recorder provided a time-history of the indicated airspeed, pressure altitude, and normal acceleration near the center of gravity of the airplane and, while the data are being used for the fatigue analysis of empennage structures, it is more applicable for wing structures. A total of 42,155 hours of data was collected on 105 airplanes. Final documents [5 and 6] are fatigue evaluation reports for small and commuter airplane certification under Part 23 of the Code of Federal Regulations (CFR) and are based on the analysis of these data.

In the Netherlands, researchers confirmed the findings by Kim, et al. that the data from the vertical acceleration at the center of gravity of an aircraft correlated well with wing loads but not with tail loads [7]. The relationship between the aircraft motion parameters (lateral and vertical accelerations; pitch and yaw rates) and tail loads appeared to be lacking. Hence, they measured the loads due to bending moments directly with strain gages for their study, a method amenable to large aircraft. In a cooperative program with Fokker Aircraft, a KLM Fokker 100 was instrumented by the Dutch National Aerospace Laboratory (NLR). It was equipped with nine strain gage bridges (three shear and six tension bridges), an aircraft monitoring system, and a data recorder. The tail load spectra were developed from the data collected.

In the U.S., the Navy developed a method for fatigue life evaluation using a Structural Data Recording System (SDRS) that recorded up to 20 flight variables, including normal acceleration, roll rate, altitude, angle of attack, wing sweep angle, and Mach number [8]. The method used Neural Networks (NNs). It was used to predict strains on the B.L.10 longeron of an F-14. The Neural Network was trained on a file recorded during a series of maneuvers and tested on a file recorded during maneuvers typical of fleet operations. The correlation of predicted to measured strains was 0.97 for the training file and 0.93 for the validation file. Similarly, the U.S. Navy also developed a SDRS system to track fatigue loads of dynamic components in helicopters [9]. From 1993 to 1995, 42 SDRS units were installed in an AH-1W fleet, and a total of 2,836 hours of flight data were recorded. A fleet average mission spectrum was created and component fatigue lives were calculated. Cost analysis showed that substantial savings was possible with the SDRS.

More recently, Neural Networks were also used to study fatigue crack growth. Crack-propagation noise is recorded using acoustic emission transducers. Environmental and structural noises distort acoustic emission signals beyond easy recognition. Hill, et al. [10] showed that Neural Networks could be used effectively to classify acoustic emission signals for aluminum structures. Classification of the acoustic signals was accomplished by an unsupervised Neural Network known as Self-Organizing Maps (SOM).

1.3 CURRENT APPROACH.

The objective of this work is to improve the methodology for the prediction of the empennage in-flight maneuver loads of general aviation aircraft. Kim, et al. were successful in developing a

methodology for the prediction but it required the use of angular accelerometers that are relatively expensive for general aviation use and the predictions for the horizontal tail were not as accurate as for the vertical tail.

In 1993, Kim found that multivariate regression analysis was not sufficient for predicting empennage flight loads. The analysis showed that, while there was a relatively strong correlation between airplane lateral accelerations (CG_Ny and Tail_Ny) and the strains in the vertical tail, the correlation was very weak between airplane vertical accelerations (CG_Nz and Tail_Nz) and horizontal tail strains. The major reason was due to the presence of nonlinear relationships between remote sensor data and strain gage measurements caused by time delays in maneuver loads. This time delay (phase lag between sensor and strain gage measurements) varied from maneuver to maneuver. Kim hypothesized the existence of some nonlinear relationship between the accelerations near the CG of the aircraft and the strains in the empennage structure. To solve this problem, Neural Network analysis was chosen. Neural Network analysis is a powerful tool for searching relationships of this type where there is an unknown, possibly nonlinear, relationship between a set of input parameters (data collected at the aircraft CG) and the outcome (strains in the empennage).

In this work, it is hypothesized that predictions can be improved with Neural Networks developed for a particular maneuver and applied to that maneuver than from a Neural Network trained on the composite of maneuvers and applied to a specific maneuver. In general, the training of a Neural Network is sensitive to the specificity of the data. If it is possible to separate clusters of data as specific maneuvers with Neural Networks, after separation of the data by maneuvers, a different maneuver-specific Neural Network could be applied to each of the data clusters to obtain the loads with improved accuracy.

For the previous work, the hardware required for the predictions included three angular accelerometers. Angular accelerometers are too expensive for general aviation use and significant cost savings is possible if relatively inexpensive solid-state rate gyros can be substituted for the angular accelerometers. To validate this approach, angular acceleration information was provided to Neural Networks both directly after numerical differentiation of the rate-gyro signals and indirectly with rate-gyro signals, allowing the Neural Networks to self-determine the best relationship. Numerical differentiation of the rate-gyro signals provided a slightly better result and is presented in this report.

2. BACKGROUND THEORY.

2.1 BASIS OF NEURAL NETWORK ARCHITECTURE.

Neural Networks represent an attempt to copy the working human brain. NNs belong to the field of artificial intelligence but differ from expert systems. No initial knowledge or explicit rules for processing the knowledge have to be coded by the programmer. A NN learns the rules by itself and is very useful in problems of prediction, classification, object or word recognition, optimization, and filtering.

Each functional part of the brain finds its equivalent in NNs. In the brain, neurons receive and process signals from other neurons through input structures called dendrites. If activated sufficiently by this signal, the neuron will emit an output signal through a structure made of different "branches" called an axon. The output signal is transmitted from the branches of the axon to different neurons' dendrites (input structures) through connections called synapses.

In NNs, neurons become processing elements (PEs). The processing performed by a neuron, due to input signals that it receives, is represented by a transfer function (TF). This TF can be a threshold function that only transmits information when the input signal reaches a minimum level. The output path of a PE, which would be an axon for the brain, is connected to the input path (dendrites) of other PEs by connection weights that correspond to the synaptic strength of neural connections.

Signals on the input lines to a PE are modified by the weights prior to being summed. Then, the weighted summation becomes the input to a TF. Most common are sigmoid and hyperbolic tangent functions. The choice of TFs depends mostly on the required characteristics (continuity of the function and its derivative, etc.) and on the range of input data and expected answers. Details about TFs can be found in references 11 and 12.

Usually, NNs consist of many PEs joined together and organized in layers. A NN is composed of an input layer, which receives all the external data, and an output layer, which holds the response of the network to a given input. Between those two layers are the hidden layer(s). In figure 1, circles represent the PEs.

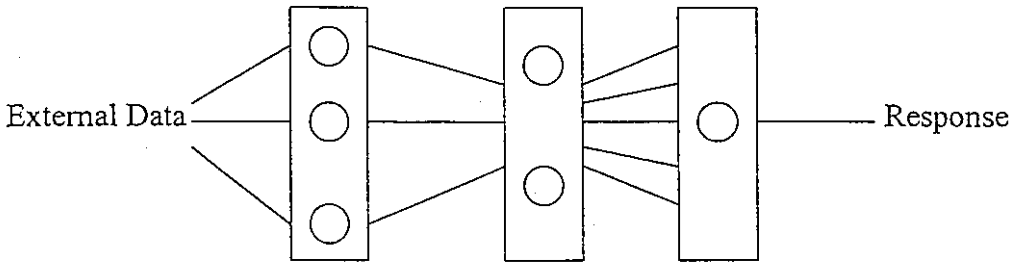

FIGURE 1. TYPICAL ARCHITECTURE OF A NETWORK

2.2 NETWORK OPERATION.

In as much as humans are first taught to speak, count, read, and recognize people, subsequently, they expand on this knowledge to situations that are new. A similar process is used in NNs. In the first phase, the network is trained on a training set: it generates rules. In the second phase, the network applies the rules learned to a new set of data (the validation set).

In supervised training, the correct answer is given to the network. By comparing its answer to the correct answer, the network corrects the rule that it was using through modification of input weights of the PEs (figure 2).

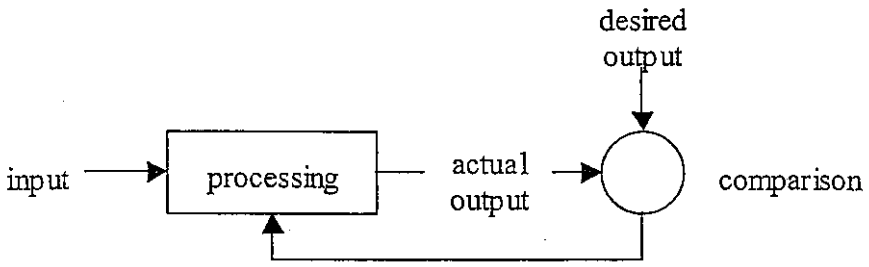

FIGURE 2. SUPERVISED TRAINING

In unsupervised training, the correct answer is not provided to the network. The most that can be expected from this model is to identify groups with similar patterns, a process that is called "clustering." The SOM is an application of this method.

2.3 BACKPROPAGATION NEURAL NETWORK.

The backpropagation, or back-error propagation, method is perhaps the most widely used NN. In this work, it is used for prediction of strains and classification of flight maneuvers. The training process is supervised. It follows a simple concept: if the NN gives a wrong answer, the weights are modified until the error is minimized.

2.3.1 Backpropagation Architecture.

Backpropagation NNs are usually layered with each layer fully connected to the layer below and above. PEs of the same layer are not interconnected. In figure 3, the weights of connections are denoted, *vih*, between input PEs (*Xi*) and hidden PEs (*Yh*), and, *ωho*, between hidden PEs (*Yh*) and output PEs (*Zo*).

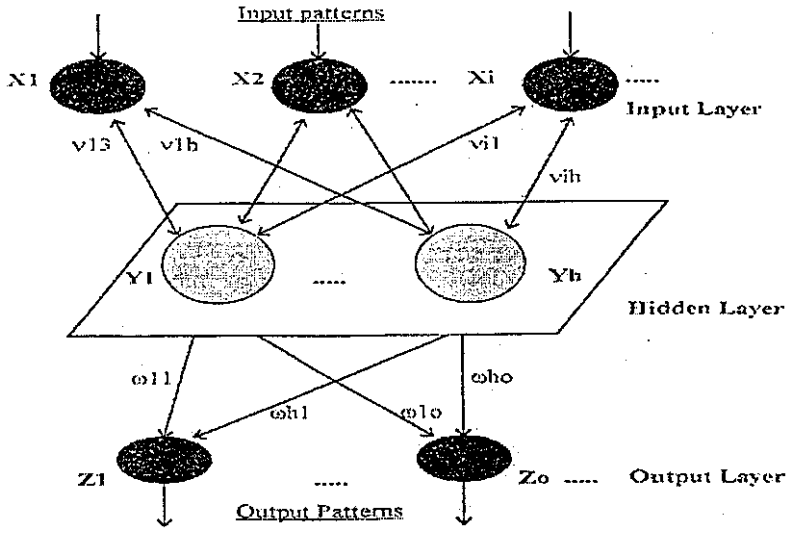

FIGURE 3. BACKPROPAGATION NEURAL NETWORKS, FULLY INTERCONNECTED

Initial values of the weights have to be set before the training can begin. If they are too small, the application of a transfer function to the weighted sum will induce very small inputs to the hidden and output layers. If initial values are too large, inputs to the hidden and output layers will be large and may fall in the saturation region of the transfer function. In that region, the derivative of the transfer function is close to zero and may prevent the algorithm from converging. To solve this dilemma, a common procedure is to assign random values between -0.5 and 0.5 to the weights.

The input layer must have a number of PEs equal to the number of input variables. PEs in the output layer must be as many as the predicted variables for the prediction problem and as many categories as possible for classification problems. The best number of PEs in the hidden layer is found by trial and error. The number of hidden layers is determined by the type of problem. For example, a linear problem requires no hidden layer while problems that are more complex necessitate one or more hidden layers.

2.3.2 Backpropagation Training Algorithm.

A NN is given an input that propagates forward from the input layer through each hidden layer to the output layer. The output PE provides the network's response. When the network corrects its internal parameters, the correction mechanism starts with the output PE and propagates backward ("backpropagates") through the hidden layers to the input layer. At a glance, the training of a backpropagation NN consists of three phases as shown in figure 4.

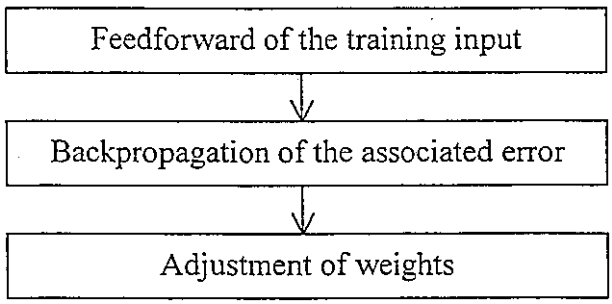

FIGURE 4. PRINCIPLE OF BACKPROPAGATION TRAINING

The adjustment of weights follows a learning rule. A description of different types of rules can be found in reference 13. Only one of them, the Delta-Bar-Delta rule, is explained here.

During training, each PE receives a weighted sum of inputs to which it applies a transfer function, F. The result is named the activation:

$$yh = F(\Sigma xI\, vih) \qquad \text{for a PE in the hidden layer}$$

$$zo = F(\Sigma yh\, \omega ho) \qquad \text{for a PE in the output layer}$$

Each output PE compares its output value "*zo*" with its target value "*to*" to determine the associated error for that pattern. Based on this error, correction factor δo and δh are computed using the following equations

$$\delta h = (to - zo) * F'(\Sigma yh\ \omega ho) \qquad \text{for an output PE}$$
$$\delta h = \Sigma(\delta o\ \omega ho) * F'(\Sigma x\ i\ vih) \qquad \text{for a hidden PE}$$

From these two equations, it can be seen that δh can only be calculated once δo has been determined; the error is backpropagated. The derivative of the transfer function, F, is needed. Hence, for the ease of solution, F is chosen such that it is continuous and easily differentiated. The logistic and hyperbolic tangent transfer functions are the most commonly used in backpropagation NNs.

A large value for δ indicates that a large correction should be made to the incoming weights. Its sign reflects the direction in which the weights should be changed. δ is used to distribute the error from the output PE to the input layer through the hidden layers. After all the δ factors have been determined, the weights for all layers are adjusted simultaneously. The adjustment to the weight ωho, (from hidden PE *Yh* to output PE *Zo*) is based on the factor δo and the activation value *yh* of the hidden PE *Yh*:

$$\Delta \omega ho = \alpha\, \delta o\, yh$$

In the same manner, the adjustment to the weight vih (from input PE *Xi* to hidden PE *Yh*) is calculated as

$$\Delta vih = \alpha\, \delta h\, xi$$

α is the learning rate selected by the operator before the training begins. Its value ranges from 0 to 1. The learning rate affects the duration of the training; the larger α is, the larger the correction of the weights per training pass. The network is at risk to oscillate about the solution without converging. Conversely, if α is small, the weights are corrected by small amounts at each period of the training, and it takes too much time to converge. The value of α can be varied for more efficient learning. In this case, the learning rate α would begin at a relatively large value and would be decreased over the learning process. The initial value of α is set by choice, typically between 0.2 and 0.5. The value of α is then decreased as iterative training continues. Acceptable rates of change are

$$\alpha t = \alpha 0(1 - t/T)$$

where t is the current training iteration, and T is the total number of training iterations to be completed, or

$$\alpha t = r * \alpha$$

where r is the learning coefficient ratio. To determine if α should be modified, a transition point is defined: it is the number of iterations that must occur before α is reduced, an iteration being the processing of one input data set. The transition point should be at least equal to the number of training data sets.

In order to determine if the training has been achieved, tests are performed periodically on the training and validation sets. When the results start degrading, training is stopped and the weights are held at their values.

2.4 UNSUPERVISED NEURAL NETWORK, SELF-ORGANIZING MAPS.

In unsupervised training, no additional information is necessary other than the input vector. Neural Networks have to identify groups of similar input patterns. Teuvo Kohonen developed this technique between 1979 and 1982. The concept is to consider the output cells as occupying positions in space. Cells close to each other are known as neighbors. The positions of cells are mapped to their behavior so that cells with similar input become neighbors. In the SOM, clusters aggregate geometrically within the network output layer. This method was tested to recognize and categorize maneuvers from the data collected during flights.

2.4.1 Self-Organizing Map Architecture.

A SOM typically consists of three layers: an input layer, a hidden or Kohonen layer, and an output layer. The hidden layer allows the SOM to complete its task. The input layer is fully connected to the bidimensional Kohonen layer. PEs are interconnected in the Kohonen layer but not in the input or output layers (figure 5).

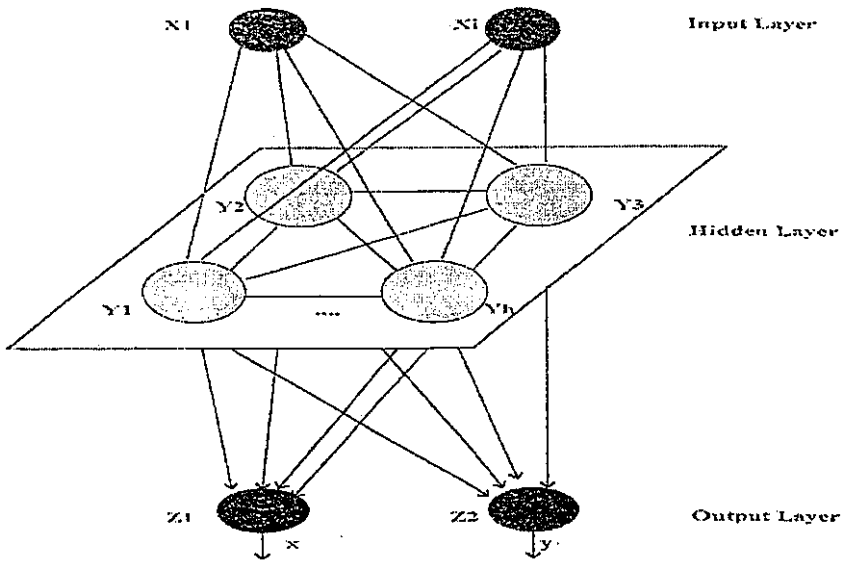

FIGURE 5. SELF-ORGANIZING MAP ARCHITECTURE

The capability of a SOM to classify data is dependent on the number of PEs. Even though the optimum number of PEs in the hidden layer cannot be determined precisely, it does increase with the number and complexity of inputs.

2.4.2 Self-Organizing Maps Training Algorithm.

When an input pattern is presented to the NN, each input PE transmits a signal to the hidden layer. Hidden PEs then sum their inputs and compete to find a single winning PE. The Euclidean distance, D, of each hidden PE is calculated and the PE with the smallest D is selected. The Euclidean distance measures the difference between the desired signal and the input pattern. It is defined as

$$D = \| X - Wh \| = \sqrt{\sum(i = 1..n)(xi - \omega ih)^2}$$

The suffix h refers to a PE of the hidden layer and i to a PE of the input layer. n is the number of input values, X is the vector of inputs, and W is the set containing the corresponding weights:

$$X = (x1, x2, x3, ..., xn)$$
$$W = (\omega 1h, \omega 2h, \omega ih, ..., \omega nh)$$

Those quantities are presented in figure 6.

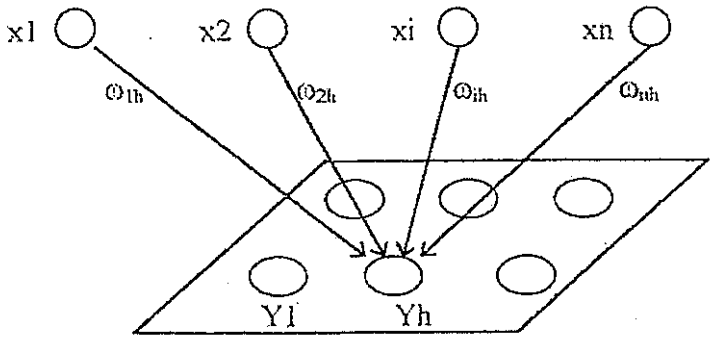

FIGURE 6. CONNECTIONS FROM THE INPUT LAYER TO A HIDDEN PE *Yh*

The selected PE, or the winning PE, denoted by c, is chosen such that

$$Dc = \| X - Wc \| = minh \{\| X - Wh \|\}.$$

If two PEs have the same value Dc, then by convention, the unit with the lower index h is chosen. The output of the winning PE is translated into a coordinate by output neurons. These coordinates represent the input signal to the topological map.

Once a winner has emerged, the PEs in the immediate vicinity (neighborhood Nc) of the winner adjust their weights to more closely resemble the winner. The size and shape (square, diamond,

circle) of the neighborhood must be determined before the training begins. Weights of the PEs are updated as follows:

$$\Delta\omega ih = \alpha(xi - \omega ih) \quad \text{if PE } h \text{ is in } Nc$$
$$\Delta\omega ih = 0 \quad \text{otherwise}$$

and

$$(\omega ih)\text{new} = (\omega ih)\text{old} + \Delta\omega ih$$

where α is the learning rate.

Next, the size of the neighborhood needs to be specified. Typically, its initial width is relatively large and is decreased during training iterations. For example, if Nc has a square shape of width "d" the value of d is decreased according to the equation

$$d = d0\ (1 - t/T)$$

where $d0$ is the initial value of d. The final neighborhood size is chosen to be either one or two. It contains only the winning PE or the winning PE and its immediate neighbors. At a glance, the training of a SOM is represented by three (phases see figure 7).

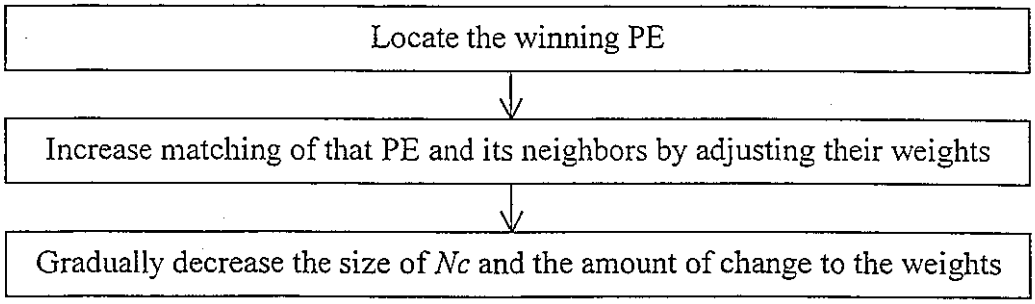

FIGURE 7. PRINCIPLE OF SOM TRAINING

Once the NN has been trained, the weights of all the PEs are held constant.

2.5 LEARNING VECTOR QUANTIZATION NEURAL NETWORK.

A method similar to SOM that uses supervised training is the Learning Vector Quantization method (LVQ). This method is used for problems of classification. It is a variation of the Kohonen SOM method that merges self-organization and supervised training. Its purpose is classification, and several versions of the LVQ method are available.

2.5.1 Learning Vector Quantization Architecture.

Like in SOM, LVQ models have three layers: an input layer with as many PEs as there are input parameters, a hidden or Kohonen layer, and an output layer with a PE for each category. Once again, the number of PEs in the hidden layer is determined by trial and error. However, each category of the output layer has corresponding same number of PEs in the hidden layer.

2.5.2 Learning Vector Quantization Training Algorithm.

2.5.2.1 LVQ1.

Other versions of LVQ NN are based on the method described here. Like in SOM, a global winning PE is first determined. It has the smallest Euclidean distance from the input vector:

$$Dc = \| X - Wc \| = minh \{ \| X - Wh \| \}$$

The calculation of the Euclidean distance for the global winner does not involve any bias factor.

In the simplest version of LVQ, only the weights of the winning PE are modified. The main difference with the SOM lies in the type of modification performed on the weights. The category of the input vector is known, and each PE of the hidden layer is assigned to an output PE, i.e., to a category. The winning hidden PE, h, may or may not be in the category of the input vector. The modification of the winner's weight is:

$\Delta\omega ih = \alpha(xi - \omega ih)$ if PE h is in the correct category
$\Delta\omega ih = -\alpha(xi - \omega ih)$ if PE h is not in the correct category

and

$$(\omega ih)\text{new} = (\omega ih)\text{old} + \Delta\omega ih$$

Hence, if the PE is in the good category, its weight becomes closer to the input signal, and if the PE is a wrong category, its weight diverges from the input signal. Here again, α is the learning rate and is reduced depending on the learning rate ratio.

In this simple form, LVQ1 algorithm presents a major inconvenience. Some PEs tend to win each time, while others have very few chances to win. An option called LVQ1 with conscience [13] permits control of how often each PE wins. It is similar to the mechanism described previously. LVQ1 with conscience determines not only a global winner among all the classes but also an in-class winner.

The frequency with which the PE h wins is given by

$p(h)=(1-\beta)*p(h)$ if h is not the in-class winner for this step
$p(h)=(1-\beta)*p(h)+\beta$ if h is the in-class winner for this step

In LVQ models, β is called the frequency estimation. The in-class winner is determined by using a biased Euclidean distance

$$B(h)=\gamma[N*p(h)-1]$$

where N is the total number of PE in the hidden layer and γ is the conscience factor. In LVQ1 with conscience, the in-class winner PE is always moved towards the input signal, even if it is in the wrong class.

2.5.2.2 Other Versions: LVQ2, LVQ2.1, and LVQ3.

In other versions, not only the winning PE, but also the closest PEs from the winning PE are modified. In LVQ2, the second winner is modified if it is in the right category while the winner is not. However, this correction happens only if the second winner is close enough to the winner (at a distance ω). In LVQ2.1, the winner and second winner are both modified if they are within a distance ω, as long as any one of them is in the correct category while the other is not. Finally, in LVQ3, the two first winners are modified as long as they are within the distance ω.

2.6 TRAINING PARAMETERS.

Training parameters control the changes that occur in the hidden layer during the learning period. The learning coefficient α, the learning ratio, and transition points have been mentioned previously, but two more parameters need to be explained: β and γ. They control the frequency $p(h)$ that PE Yh is allowed to win. This controlling mechanism keeps track of how often a PE wins and adjusts the network so that losing PEs are encouraged to win. This allows for better separation of categories in the output map. The frequency $p(h)$ with which the PE h becomes a winner is updated at each transition point:

$$p(h)=(1-\beta)*p(h) \qquad \text{if } h \text{ is not the winner for this step}$$
$$p(h)=(1-\beta)*p(h)+\beta \qquad \text{if } h \text{ is the winner for this step}$$

γ is used to determine a bias term Bh that is added to the Euclidean distance function for the PE Yh. It favors PEs that have not won recently and thus encourages all the PEs of the hidden layer to be utilized.

$$Bh=\gamma[N*p(h)-1]$$

where N is the total number of PEs in the hidden layer. Initially, $p(h)$ is set to $1/N$, so $Bh = 0$. If a PE Yh wins often, $p(h)$ is high, Bh is high, and a high number is added to the Euclidean distance D so that the PE has fewer chances to win. On the other hand, if a PE Yh does not win often, $p(h)$, Bh, and D become smaller, and PE Yh has a greater chance to win.

3. TESTING AND DATA PROCESSING.

3.1 EXPERIMENTAL SETUP.

A Cessna 172P (C172P) was instrumented with air data transducers to record airspeed and altitude, two linear accelerometers near the aircraft CG, two linear accelerometers mounted on the tailcone bulkhead, three rate gyros and three angular accelerometers mounted on an instrument pallet in the baggage compartment, and a pitot-static flight test boom with angle-of-attack and sideslip vanes. The airplane was also instrumented with eight temperature compensating strain gages: two in the right wing front spar, four in the left horizontal tail stabilizer front spar, and two in the vertical fin front spar. For this work, only one gage is used for each tail, i.e., the gage on the port or left spar cap for the vertical tail (VTP), and the gage on the top spar cap for the horizontal tail (HTT), the others serving as backup gages. The instrumentation aboard the C172P is detailed in reference 3 and is included here as table 1 and figure 8.

TABLE 1. INSTRUMENTATION DESCRIPTION

Type	Qty	Model	Manufacturer
Linear Accelerometer	4	Columbia Research SA-107BHP	Columbia Research Laboratories, Inc.
Rate Gyro	3	Gyrostar ENV-05H-02	Murata Mfg. Co. Ltd.
Angular Accelerometer	3	Shaevitz, Inc.	Shaevitz, Inc.
Portable Computer	1	IBM PC Compatible	
Portable Data Acquisition System	1	Daqbook 216	IOTech, Inc
Strain Gage	4	Columbia Research Model 2681	Columbia Research Laboratories, Inc.

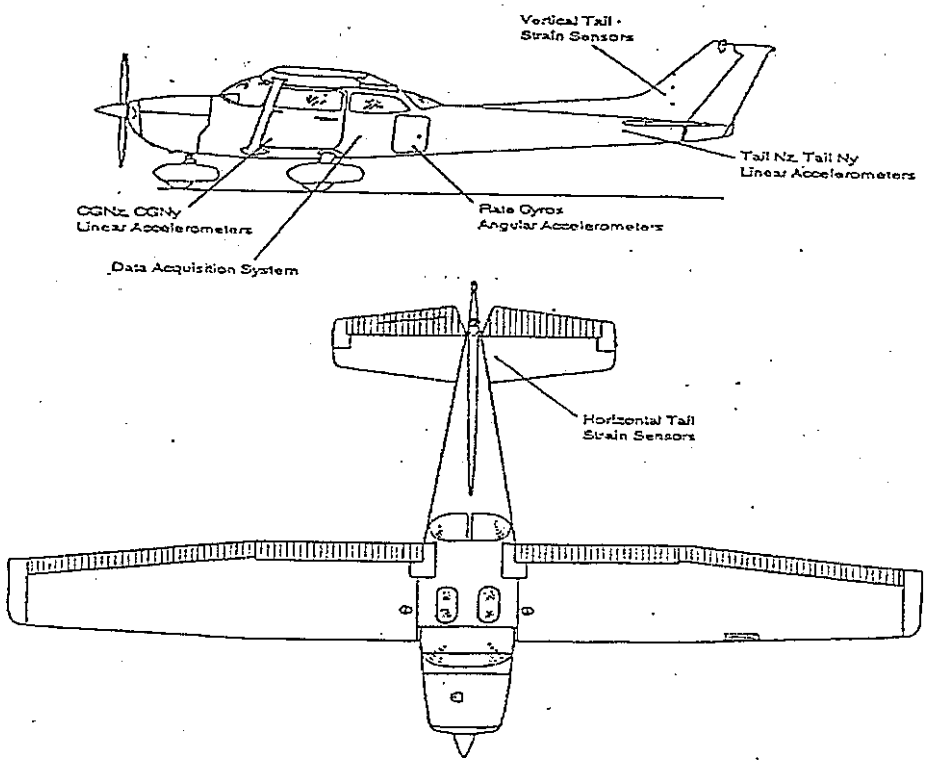

FIGURE 8. INSTRUMENTATION SETUP

3.2 DATA COLLECTION.

3.2.1 Collected Data and Testing Procedure.

Twenty-two channels were recorded using the data acquisition system *Iotech DaqBook 216 portable*: eight strain gage outputs, sideslip, angle of attack, airspeed, altitude, four linear accelerometers—two near aircraft CG (CG_Nz and CG_Ny) and two in the aircraft tailcone bulkhead (Tail_Nz and Tail Ny), three rate gyros (pitch, roll, and yaw), and three angular accelerometers (pitch, roll, and yaw). Data was collected in-flight using a laptop computer. The sampling rate was set at 200 Hz, while the recording duration varied from 30 seconds (6,000 records) to 5 minutes (60,000 records). Data collection is started manually by the flight test engineer at the pilot's signal. A few seconds of straight and level data are collected before and after each maneuver to verify that the instruments are operating correctly and to provide a baseline. The data are collected in binary format and converted to ASCII format for easy analysis. Detailed descriptions of the data acquisition system and testing procedure can be found in reference 3.

The methodology should be applied to any flight condition (speed, altitude, maneuver) within the flight envelope. However, for safety reasons, only a subset of the envelope was tested. Files containing the 22 channels were collected for flights at 3,000 ft for seven different maneuvers: dutch-roll, roll, sideslip left, sideslip right, stabilized-g turn right, stabilized-g turn left, and push-pull. Data was collected at 65, 80, and 95 KIAS. Maximum bank angles for roll and stabilized-g turns were 60 degrees, and a maximum of 3 g's was induced during push-pull maneuvers.

Table 2, taken from reference 3, shows the maximum and minimum strain increments in the empennage for each maneuver in the training and validation set for the NN.

TABLE 2. MINIMUM AND MAXIMUM STRAIN INCREMENTS RECORDED

	Vertical Tail		Horizontal Tail	
Maneuver	Maximum Strain Increment ($\mu\varepsilon$)	Minimum Strain Increment ($\mu\varepsilon$)	Maximum Strain Increment ($\mu\varepsilon$)	Minimum Strain Increment ($\mu\varepsilon$)
Dutch-roll	165	-178	112	-131
Roll	29	-119	95	-160
Sideslip left	4	-92	26	-98
Sideslip right	17	-68	82	-59
Push-pull	-5	-61	44	-97
Stabilized-g turn right	4	-71	32	-67
Stabilized-g turn left	14	-69	17	-63

From table 2, it can be seen that roll and dutch-roll maneuvers induce the greatest strain increments for the horizontal and vertical tails. Kim, et al. predicted the strains induced by those two maneuvers without considering the other maneuvers. The other maneuvers induced strains

only in the insignificant region (see section 4.1). The strain increments shown are relative to the baseline 1-g level flight condition.

In theory, the horizontal tail loads should be very small during pure sideslip maneuvers. However, the strains measured for these maneuvers were nevertheless relatively high. This is because as the airplane is maintained in a 1-g steady-state sideslip condition, the airplane tends to bank and lose altitude. To counter these tendencies, the pilot must apply opposite aileron and up elevator that increases the horizontal tail load.

3.2.2 Preparation of Data for Analysis—Data Filtering.

Initial work with the *IoTech* data acquisition system showed significant noise in the data. To filter the data, *DADisp 4.0* software, from DSP Development Corporation, was used to perform a frequency spectrum analysis. Signals corresponding to the pilot control inputs, i.e., maneuver loads, had frequencies of 1.5 Hz or less so a low pass filter with a cutoff frequency of 2 Hz was used with the *DADisp4.0*. Kim, et al. compared predictions obtained using filtered and unfiltered data to train the Neural Network. Filtered data yielded better predictions and were used for this work.

4. DISCUSSION OF RESULTS.

The primary objective of this research is to improve the percentage of correct predictions of significant strains in the horizontal tail. It is desired that 100% of the predicted strains in the significant region be within the tolerance band (see section 4.1). A secondary objective is to reduce the cost of the sensors required for the prediction. The methodology is required to be accurate and reliable, but also affordable for operators of small general aviation aircraft. The range of acceptable prices for the low-cost flight data recorder is estimated between $2,000 and $3,000. Since the price of an angular accelerometer can vary from $1,700 to $3,000 each, finding a suitable low-cost alternative is paramount to satisfy the design criteria. If it works, rate gyros would provide a low-cost alternative at less than $300 each.

4.1 FATIGUE DATA CONSIDERATION.

The search for a methodology to predict empennage in-flight loads is driven by the requirement to avoid aircraft structural failure due to fatigue loads. Low stresses induced on the airframe do not cause appreciable fatigue damage. An accurate prediction of those stresses is of no interest. Only stresses that cause failure below the endurance limit need to be investigated.

Figure 9 is the S-N curve of an Al 2024-T3 sheet [14]. For this material, the worst case for fatigue damage is for stresses over 5,000 psi with full stress reversal. The strain gages in the test aircraft are located approximately one-third semi-span outboard from the fuselage centerline. If the bending moment in the horizontal tail causes maximum stresses at the wing root, the minimum stress of concern at this location would be 5,000 psi. Assuming a linear relationship, the stress at the gage location would be approximately 1,600 psi. Thus, the insignificant region was defined conservatively as below 1,000 psi or below 100με.

Similarly, it is desired that the stresses be predicted as accurately as they are plotted in the S-N curve. Thus, a tolerance band of ±500 psi or ±50με was adopted as successful prediction criteria.

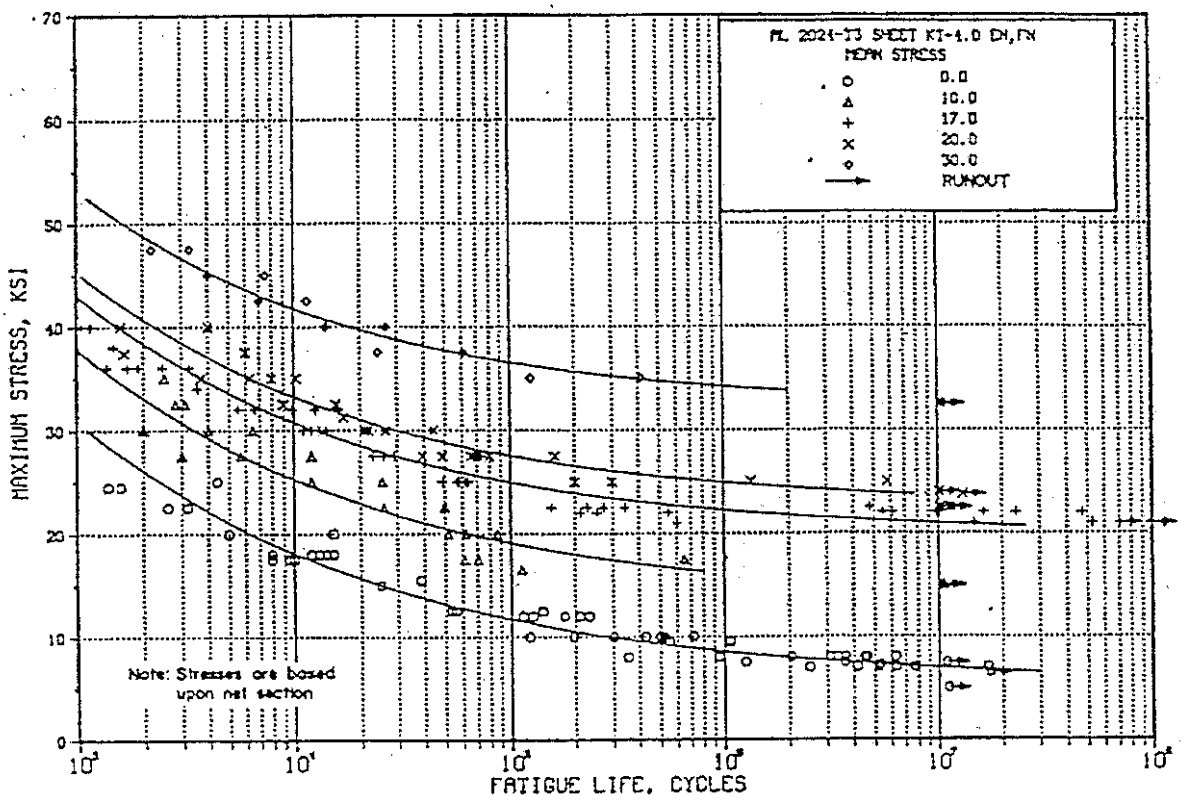

FIGURE 9. S-N CURVE OF AN Al 2024-T3 SHEET

Inspections of figure 10 show that, for the flight envelope tested, dutch-roll (DRL) and roll maneuvers were the most significant maneuvers for the horizontal tail. In comparison, the peak load shown during the push-pull (PPL) maneuver corresponds to a 3-g pull-up condition. The peak load for stabilized-g turns (STGL and STGR) occurs during the roll entry, and the peak load during sideslip (SSL and SSR) maneuvers occurs due to up elevator deflection required to maintain altitude; a condition that is not common during routine operations.

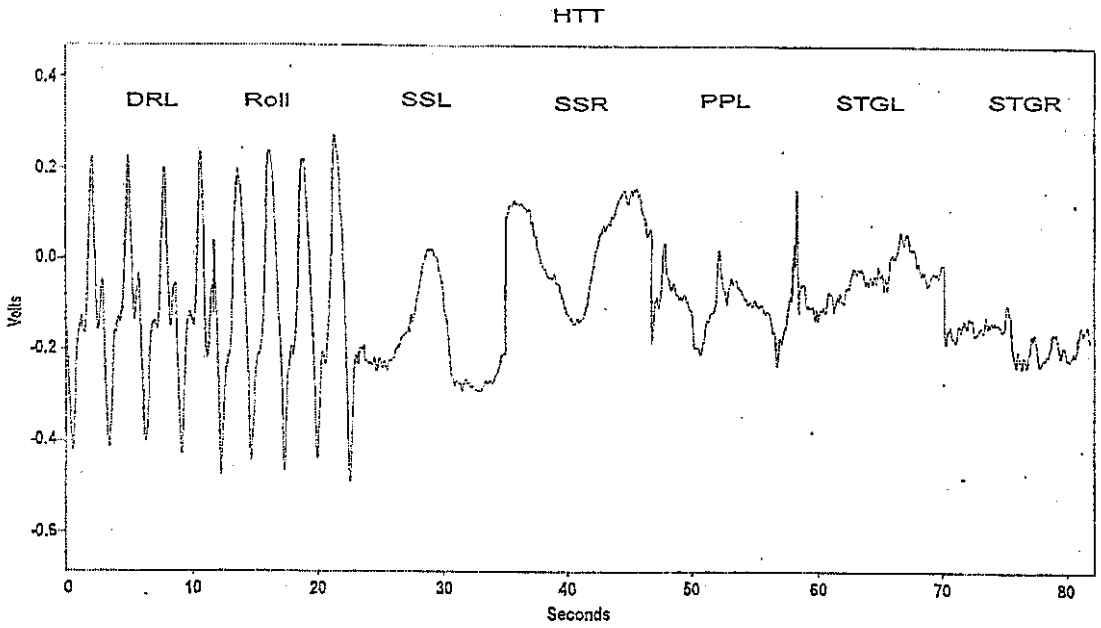

FIGURE 10. HORIZONTAL TAIL LOAD DURING MANEUVERS

Kim, et al. [3] used two sets of NNs for empennage load predictions: one for the horizontal tail and one for the vertical tail. For the horizontal tail, the best predictions were obtained when the NN was trained on angular accelerometer signals and vertical acceleration at the aircraft CG (CG_Nz). Data used for the training belonged to dutch-roll, roll, and push-pull maneuvers at 80 KIAS. For the vertical tail, the NN was trained on signals from the angular accelerometer along the x- and y-axes collected during roll and dutch-roll maneuvers. NNs used were of modular architecture, which is a generalization of backpropagation NNs. The learning rule was of the type Extended Delta-Bar-Delta, and the transfer function was the hyperbolic tangent. For the horizontal tail, 93.3% of the significant strains were predicted within ±50με of their measured value. Incorrect predictions were obtained when testing the NN on data from dutch-roll and roll maneuvers at 95 KIAS. In contrast, 100% of the strains were predicted correctly for the vertical tail.

4.2 RESULTS FROM SUPPRESSION OF THE ANGULAR ACCELEROMETER.

To meet the cost objective of the instrumentation, it was necessary to find a suitable low-cost substitute to the relatively expensive angular accelerometers. To this extent, validation that the angular accelerometer data could be obtained from rate gyros was necessary.

4.2.1 Utilization of the Signal from Rate Gyroscopes.

Rate gyroscopes indicate rate of rotation, i.e., the angular velocity ω, about an axis, while angular accelerometers measure the rate of change of angular velocity with respect to time, $d\omega/dt$. If the signal from the rate gyro is smooth enough, it can be differentiated to provide the same information as the angular accelerometer. The differentiation of the rate-gyro signals can be

obtained by multiplying the difference between two succeeding samples by the sampling rate. A sample of signals from angular accelerometers (figure 11) and those obtained by differentiation of rate-gyro signals (figure 12) is shown below.

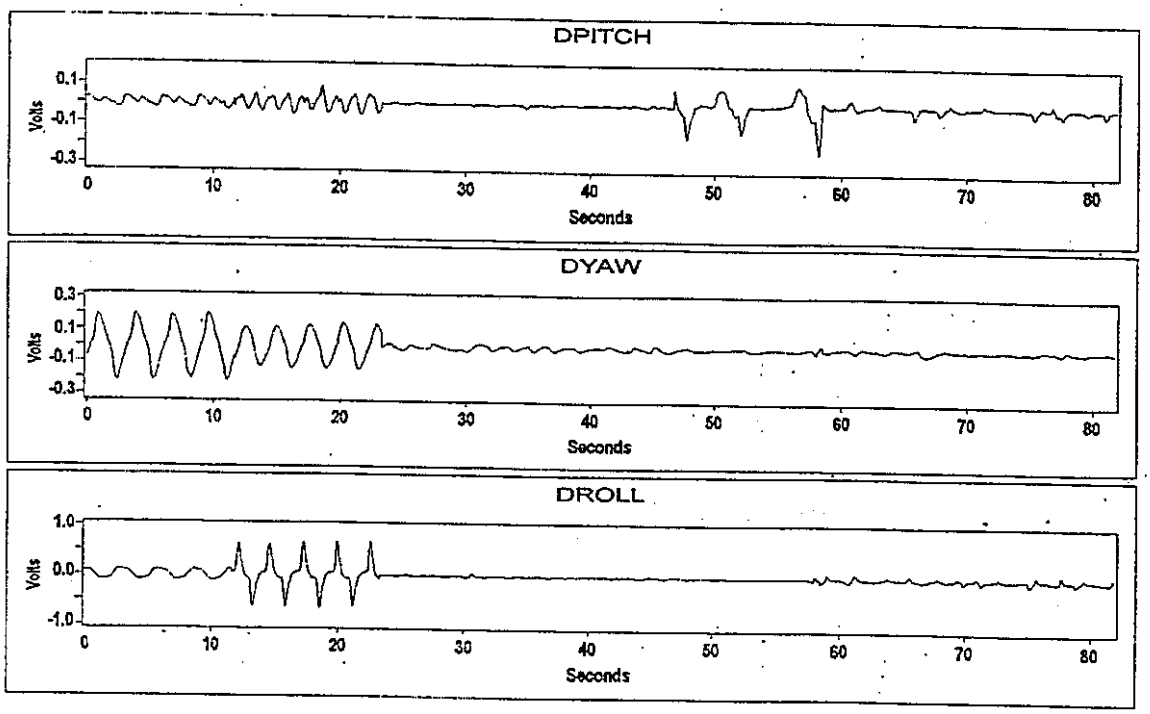

FIGURE 11. ANGULAR ACCELEROMETER SIGNALS

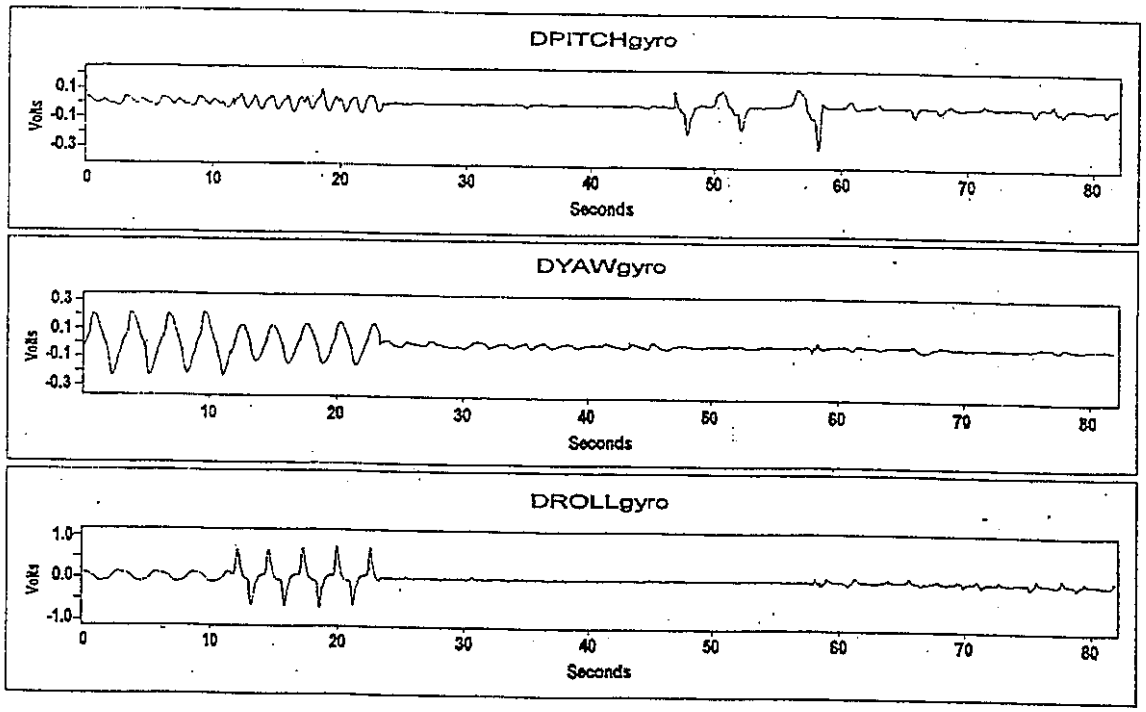

FIGURE 12. RATE-GYRO SIGNALS AFTER NUMERICAL DIFFERENTIATION

4.2.2 Findings.

Signals from the angular accelerometers were replaced by signals from the rate gyros after numerical differentiation. Results were comparable if not slightly better. Table 3 presents those results.

TABLE 3. SAMPLE OF NEURAL NETWORKS PREDICTIONS TRAINED ON RATE-GYRO SIGNALS AFTER NUMERICAL DIFFERENTIATION

Results Obtained With "Predict" Prediction of Horizontal Tail Maneuver	Entire Data File			Significant Stress Region		
	No. of Records	No. of Error >500 psi	Max Error	No. of Records	No. of Error >500psi	Max Error
Dutch-roll, 80 KIAS, Training set	5,801	0	37	384	0	18
Dutch-roll, 80 KIAS, Validation set	5,801	48	68	45	0	14
Dutch-roll, 95 KIAS, Training set	5,801	568	73	184	0	16
Dutch-roll, 95 KIAS, Validation set	5,801	456	74	730	167	74
Roll, 65 KIAS, Training set	5,801	0	48	313	0	23
Roll, 65 KIAS, Validation set	5,801	16	52	25	0	19
Roll, 80 KIAS, Training set	5,801	0	46	585	0	31
Roll, 80 KIAS, Validation set	5,801	54	59	297	0	23
Roll, 95 KIAS, Training set	5,801	40	60	1,021	19	53
Roll, 95 KIAS, Validation set	5,801	91	61	748	4	51
Total	232,040	17,511	(34.1)*	4,332	190	(32.2)

*Average error

In the significant region, the NN trained on the signals from the rate gyros after differentiation gave better predictions than those trained on the angular accelerometer data. The number of predictions outside of the tolerance band decreased from 275 (6.4%) to 190 (4.4%) records. The accuracy of the predictions was improved slightly. The average error of prediction decreased from 34.1με to 32.2με. Throughout the entire file, the results were similar. It was unexpected that better results were obtained with differentiated rate-gyro signals than with angular accelerometers. This is believed to be due to variations in the sensitivity and signal quality of the sensors and is not considered to be of much significance.

4.3 RESULTS FROM IMPROVEMENT OF THE PREDICTIONS BY MANEUVER CLASSIFICATION.

As previously stated, prediction of strains should be accurate for any flight condition. The seven maneuvers flown represent the main motions that can be encountered in flight. It was hypothesized that predictions obtained with a NN developed for each specific maneuver would be more accurate than the predictions obtained with a single NN trained on composite maneuvers. After validation of this hypothesis, the next step is to distinguish in the data file, the records belonging to each of the seven maneuvers and to apply the corresponding NN on the data clusters. Since the user has no interaction with the data acquisition system during the flight, the separation of data clusters by maneuver is performed postflight. NN is used to recognize the characteristics of each maneuver and automatically classify the records by maneuvers.

4.3.1 Neural Networks Specific to Each Maneuver.

Specific NNs were built and tested for each maneuver. The parameters modified for each NNs training were the input variables and speed of training. Results were generally close, and table 4 represents the best combinations of parameters obtained.

TABLE 4. PARAMETERS USED FOR THE NN OF EACH MANEUVER

	Input Parameters	Speed
Dutch-roll	CG_Ny, rate gyro differentiated	80 and 95 KIAS
Roll*	CG_Nz, rate gyro differentiated	65, 80, and 95 KIAS
Sideslip	CG_Ny, rate gyro differentiated	80 and 95 KIAS
Push-pull	CG_Nz, rate gyro differentiated	80 KIAS
Stabilized-g turn	CG_Ny, roll, yaw, pitch	80 KIAS

* For the roll maneuver, the NN used was also trained on the dutch-roll maneuver. It still gave very good results for roll and allowed the correct prediction of dutch-roll records that were improperly classified as roll.

4.3.2 Classification.

4.3.2.1 Evaluation of the Results.

Classification aims to discover significant regularities within the input data of different groups. The network is trained to recognize to which group or data cluster the data record belongs. Ideally, the classification NN should be able to classify correctly 100% of the records, but two important factors must be considered. Even if the NN attained 100% correct classification on the training set, it is very likely that some records of the validation set will be classified in the wrong cluster. During test flights, and in particular during an operational flight, pure maneuvers rarely exist, e.g., a roll maneuver induces unintentional sideslips. In this case, some records might be classified incorrectly, but the prediction of strains still needs to be correct. To solve this problem, during the NN training, binary vectors were assigned with each record. For example, if two classes had to be classified, records belonging to class 1 were assigned a binary vector (1,0) and those of class 2, a binary vector (0,1). During the recall phase, a record might be classified as (0.6, 0.4). In this case, applying the weighted sum approach, the strain was calculated as the adjunction of the strain that would be induced in class 1 multiplied by the weight 0.6 to the strain induced in class 2 multiplied by the weight 0.4.

4.3.2.2 Classification Algorithm.

Different methods of classification, such as SOM, backpropagation with two and three hidden layers and LVQ were tested. While LVQ gave very good results for classification, backpropagation NN was better suited for the load prediction problem.

In order to reduce the number of classes from seven to five, sideslip right and left as well as stabilized-g turns right and left were considered as one maneuver. The choice of input variables for the classification NNs was done by visual comparison of the traces of the data channels for

the different maneuvers. For example, the first NNs tried were expected to distinguish the seven (or five) maneuvers. It was thought that any maneuver could be resolved into a combination of three fundamental motions: roll, pitch, and sideslip. Thus, a NN with only three classes was built, but the predictions were not satisfactory.

To overcome this shortcoming, a different strategy was tried with much better success. Since dutch-roll can be considered as a coupled roll and sideslip motion, the five maneuvers were classified into two classes: one with roll, dutch-roll, and sideslip, and the other with push-pull and stabilized-g turns. Finally, since maximum errors of predictions were associated with the roll maneuver, this maneuver was isolated from the four other maneuvers as the first step of the classification. Then, the dutch-roll was separated followed by push-pull, sideslip, and stabilized-g turns. Figure 13 illustrates the classification algorithm.

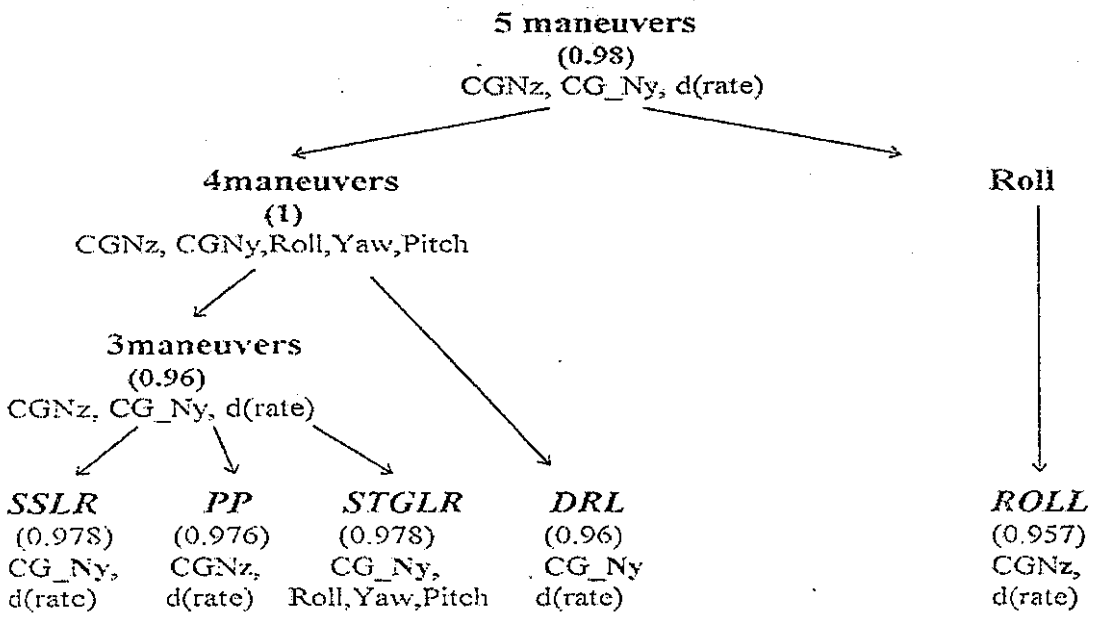

FIGURE 13. CLASSIFICATION ALGORITHM

The numbers in parentheses indicate the average rate of classification obtained. The term d(rate) represents the signals from the three rate gyros after numerical differentiation.

4.3.2.3 Findings.

Table 5 represents the classification obtained for each file. The first column indicates the maneuver corresponding to the files. The five following columns details how the files were classified. Numbers correspond to the percentages of records among the six files (three training, three validation) corresponding to each maneuver, classified in a given class.

TABLE 5. SUMMARY OF CLASSIFICATION

Maneuver	Classified as Dutch-Roll (%)	Classified as Roll (%)	Classified as Sideslip (%)	Classified as Push-Pull (%)	Classified as Stabilized-g Turn (%)
Dutch-roll	84	10	5	1	0
Roll	3	86	5	4	2
Sideslip left	2	6	86	5	1
Sideslip right	1	2	79	16	2
Push pull	0	1	5	92	2
Stabilized-g turn left	4	4	10	8	74
Stabilized-g turn right	5	4	14	20	57

Ten percent of dutch-roll was classified as roll. This explains why the NN applied to roll also had to be trained on dutch-roll. If the NN that separates roll from dutch-roll can be improved, a NN trained only on roll could be used for it.

Table 6 presents the errors of prediction obtained after classification. For comparison, table 7 gives the results published previously by Kim, et al. [3]. Table 8 summarizes the percentages of correct classification and the number of strains predicted with an error greater than 50με of their recorded values for each maneuver.

TABLE 6. SUMMARY OF PREDICTIONS AFTER CLASSIFICATION

Results Obtained With "Predict"Prediction of Horizontal Tail Maneuver	Entire Data File			Significant Stress Region		
	No. of Records	No. of Error >500 psi	Max Error	No. of Records	No. of Error >500 psi	Max Error
Dutch-roll, 80 KIAS, Training set	5,801	60	65	384	0	16
Dutch-roll, 80 KIAS, Validation set	5,801	67	61	45	0	9
Dutch-roll, 95 KIAS, Training set	5,801	0	43	184	0	13
Dutch-roll, 95 KIAS, Validation set	5,801	220	81	730	0	43
Roll, 65 KIAS, Training set	5,801	24	63	313	0	17
Roll, 65 KIAS, Validation set	5,801	105	82	25	0	24
Roll, 80 KIAS, Training set	5,801	16	55	585	0	34
Roll, 80 KIAS, Validation set	5,801	74	66	297	0	21
Roll, 95 KIAS, Training set	5,801	60	68	1,021	0	49
Roll, 95 KIAS, Validation set	5,801	78	66	748	0	42
Total	232,040	6,383 (2.75%)		4,332	0	(26.8)

TABLE 7. SUMMARY OF PREDICTIONS PREVIOUSLY PUBLISHED IN REFERENCE 3

Results Previously Published Prediction of Horizontal Tail Maneuver	Entire Data File			Significant Stress Region		
	No. of Records	No. of Error >500 psi	Max Error	No. of Records	No. of Error >500 psi	Max Error
Dutch-roll, 80 KIAS, Training set	5,801	0	49	384	0	45
Dutch-roll, 80 KIAS, Validation set	5,801	102	58	45	0	36
Dutch-roll, 95 KIAS, Training set	5,801	357	69	184	0	32
Dutch-roll, 95 KIAS, Validation set	5,801	604	82	730	202	82
Roll, 65 KIAS, Training set	5,801	0	40	313	0	30
Roll, 65 KIAS, Validation set	5,801	0	42	25	0	38
Roll, 80 KIAS, Training set	5,801	0	44	585	0	44
Roll, 80 KIAS, Validation set	5,801	58	55	297	0	39
Roll, 95 KIAS, Training set	5,801	103	66	1,021	34	55
Roll, 95 KIAS, Validation set	5,801	236	63	748	54	59
Total	232,040	16,950 (7.3%)		4,332	290	(46)

TABLE 8. SUMMARY OF CLASSIFICATION AND PREDICTION FOR EACH MANEUVER

Maneuver	Percentage of "Correct Classification"	Percentage of Errors of Prediction > 50με in Entire Files
Dutch-roll	84%	2.9%
Roll	86%	8.6%
Sideslip left	86%	4.3%
Sideslip right	79%	7.3%
Push-pull	92%	0.04%
Stabilized-g turn left	74%	0.15%
Stabilized-g turn right	57%	2.8%

The worst percentage of errors of prediction was from the roll maneuver. This is because the NN was also trained on dutch-roll. This choice was guided by the primary objective of this work, i.e., correct prediction of strains in the significant region. The predictions have improved significantly, i.e., the number of errors was reduced by two-thirds in the entire file. In the significant region, the region of primary interest, no error was greater than 50με achieving the desired degree of accuracy comparable to that of the vertical tail predictions.

5. CONCLUSIONS AND RECOMMENDATIONS.

Neural Networks (NNs) were chosen on the hypothesis that stresses induced in the empennage are related by some relationship, not necessarily linear or obvious, to the accelerations measured

near the aircraft center of gravity (CG). For the prediction of strains in the horizontal tail, it has been shown that the utilization of the signals of the rate gyros after numerical differentiation could be substituted for the signals from the angular accelerometers without losing accuracy. The results are expected to be similar for the vertical tail.

Classification of the records by maneuver was thought to be a solution for improving the accuracy of the predictions from the previously published work. NNs have been used to automatically classify the data by maneuvers. This classification of records by maneuvers, plus the application of a NN specific to that maneuver in conjunction with the weighted sum approach, allowed a significant reduction of the number of errors of prediction. For the horizontal tail, 97% of the strains for the entire data set were predicted within 50με of their recorded value. In the significant region, 100% of the strains were predicted within 50με. In comparison, previous methodology for the horizontal tail yielded only a 93% success rate within the significant region with a maximum error of 82με.

The Neural Network methodology provides a viable alternative to determining the strains induced by maneuvers in the empennage. Due to the nature of the sensors, it has an additional advantage in that rate gyros and angular accelerometer signals are much less susceptible to signal noise that is prone to strain gage measurements.

Recommendations for future work are:

a. Classification of sideslip and stabilized-g turns left and right with specific NN for each should be tested.

b. Effects of changes in altitude and airspeeds need to be investigated for high-performance airplanes with large altitude and airspeed operating envelopes.

c. An empennage load survey should be performed using the traditional method with strain gages and compared to predictions made using the NN methodology.

d. NN using power spectral density needs to be investigated for improvements in computational speed as well as maneuver and gust load separation.

e. The methodology should be tested on other aircraft and in particular on aircraft with different empennage configuration.

6. REFERENCES.

1. DeFiore, T. and D. Kim, "General Aviation Individual Airplane Load Monitoring Program," 3rd AIAA/FAA Joint Symposium on General Aviation Systems, Starkville, MS, 1994.

2. Skopinski, T.H., Aiken, Jr., W.S., and Huston, W.B., "Calibration of Strain-Gage Installations in Aircraft Structures for the Measurement of Flight Loads," NACA Report 1178.

3. Kim, D. and Marciniak, M., "A Methodology to Predict the Empennage In-Flight Loads of a General Aviation Aircraft Using Backpropagation Neural Networks," DOT/FAA/AR-00/50, Washington, D.C., February 2001.

4. Gabriel, E.A., DeFiore, T., Locke, J.E., and Smith, H.W., "General Aviation Aircraft-Normal Acceleration Data Analysis and Collection Project, Final Report," DOT/FAA/CT-91/20, Department of Transportation, Federal Aviation Administration, Washington, D.C., 1993.

5. Anonymous, "Fatigue Evaluation of Wing and Associated Structure on Small Airplanes," AFS-120-73-2, Federal Aviation Administration, Washington, D.C., 1973.

6. Anonymous, "Fatigue Evaluation of Empennage, Forward Wing, and Winglets/Tip Fins," ACE-100-01, Federal Aviation administration, Kansas City, MO, 1974.

7. Van Gelder, P.A., "In-Flight Tailload Measurements," Proceedings of the 18th ICAS Congress, AIAA, Washington, D.C. 1992, pp. 1058-1066.

8. Hoffman, M.E., "A Neural Network Prototype for Predicting F-14B Strains at the B.L.10 Longeron," NAWC Report No. NAWCADWAR-92042-60, Warminster, PA, 1992.

9. Moon Suresh and Hinger Carl, "Helicopter Dynamic Components Fatigue Tracking System," IAA9603, 1995.

10. Hill, E. and Marsden, M.L., "Classification of Fatigue Cracking Data in a Simulated Aircraft Fuselage Using a Self Organizing Map," Proceedings of the First International Conference on Nonlinear Problems in Aviation & Aerospace, S. Sivasundaram, editor, Embry-Riddle Aeronautical University Press, Daytona Beach, FL, 1997, pp. 405-410.

11. Dayhoff, J.E., "Neural Network Architectures: An Introduction," Van Nostrand Reinhold, NY, 1990.

12. Fausett, L., "Fundamentals of Neural Networks—Architecture, Algorithms, and Applications," Prentice-Hall, Inc., Englewood Cliffs, NJ, 1994.

13. Anonymous, "Neural Computing—A Technology Handbook for Professional II/PLUS and NeuralWorks Explorer," NeuralWare, Inc., PA, 1995.

14. Anonymous, MIL-HDBK-5E, Department of Defense, 1987.

APPENDIX A—NEURAL NETWORKS TESTED

The first objective was the improvement of the predictions for roll and dutch-roll maneuvers, especially at 95 KIAS. A hypothesis was that a network trained on 95 KIAS could accurately predict the strains at 95 KIAS. Since the files collected during flights contain airspeeds, it is possible to separate the data by speed (or ranges of speed), and apply the network corresponding to this speed. However, if trained on a composite file of roll and dutch-roll at 95 KIAS, Neural Network (NN) did not give correct results, even on the training file. An explanation for this is that at 80 KIAS, the difference between the maneuvers is not significantly pronounced so the network could find a general rule for roll and dutch-roll. However, as the airspeed increases, characteristics of each maneuver become more noticeable. NNs cannot find a single general relationship between the input signals and the strains for the composite file. Therefore, no classification by speed was attempted for the previously published unique NN approach.

NNs were also trained on a file composed of dutch-roll and roll maneuvers at 80 and 95 KIAS. The only file for which predictions remained outside of the tolerance band was the validation file for dutch-roll at 95 KIAS. In the significant region, the maximum error was decreased from 82με with Professional II/PLUS to 72με, and the number of records outside of the tolerance band decreased from 202 to 146 records.

When classification approach was first tested, coefficients given by the NN were not considered as "weight" coefficients, but as indicator of the class. For example, for a classification between two classes, if the resulting classification was (0.6, 0.4), the record was considered to belong to class 1, and the predicted strain associated to this record was obtained with the Neural Network of class 1. Predictions obtained in this simple way were unacceptable.

Many other different classifications were also tested. Results obtained on validation files for roll and dutch-roll at 95 KIAS are shown in tables A-1 to A-5.

TABLE A-1. RESULTS FOR ROLL AND DUTCH-ROLL AT 95 KIAS

Number of Classes	Classes	File Name	Max. Error in Entire File (με)	No. Errors >50με in Entire File	Max. Error in Significant Region (με)	No. Errors >50με in Significant Region
5	drl,roll,ppl,ss,stg	RL95V	128	822	117	431
		DRL95V	50	28	38	0
3	roll,ss,ppl	RL95V	98	226	67	3
		DRL95V	87	495	72	146
2	drl+roll+ss/ppl+stg	RL95V	100	56	100	9
		DRL95V	69	185	30	0

TABLE A-2. RESULTS FOR ROLL AND DUTCH-ROLL AT 95 KIAS

File	Roll	Dutch-Roll	Sideslip	Push-Pull	Stabilized-g Turn	Correct Classification (%)
RL65T	5423	18	157	126	77	93
RL80T	5315	109	21	212	144	91
RL95T	5460	30	48	133	130	94
DRL65T	550	5080	60	105	6	87.5
DRL80T	234	5367	186	14	0	92.5
DRL95T	428	5168	184	10	11	89
SSL65T	410	29	4877	485	0	84
SSL80T	542	213	4808	194	44	82
SSL95T	395	193	4746	281	186	81
SSR65T	22	0	4499	1290	0	77
SSR80T	189	41	4430	1034	107	76
SSR95T	51	41	4923	786	0	85
PPL65T	123	0	41	5496	141	94
PPL80T	41	1	95	5529	135	95
PPL95T	41	0	136	5619	5	97
STGL65T	68	0	1194	933	3606	62
STGL80T	164	428	214	471	4524	78
STGL95T	337	114	89	113	5148	89
STGR65T	265	389	937	2030	2180	37
STGR80T	215	15	381	826	4364	75
STGR95T	295	92	352	668	4394	75
RL65V	4423	272	414	397	294	76
RL80V	4011	355	937	424	74	69
RL95V	5151	155	280	116	99	88
DRL65V	595	4451	695	57	3	76
DRL80V	856	4577	325	34	9	78
DRL95V	783	4468	212	231	107	77
SSL65V	342	107	5223	129	0	90
SSL80V	308	1	5361	131	0	92
SSL95V	47	0	5049	579	126	87
SSR65V	186	54	3879	1138	544	67
SSR80V	143	43	4815	768	32	83
SSR95V	12	0	5101	517	171	88
PPL65V	106	0	177	5364	154	92
PPL80V	162	7	848	4701	83	81
PPL95V	45	0	364	5262	130	90
STGL65V	163	166	511	677	4284	73
STGL80V	425	337	956	447	3636	62
STGR65V	231	813	1036	1174	2547	44
STGR80V	159	188	1199	1048	3207	55
STGR95V	279	358	779	1272	3113	53

TABLE A-3. RESULTS FOR ROLL AND DUTCH-ROLL AT 95 KIAS

	Dutch-roll	Roll	Sideslip	Push-pull	SGT
Dutch-roll	29,111	3,446	1,662	451	136
Roll	939	29,783	1,857	1,408	818
Sideslip left	543	2,044	30,064	1,799	356
Sideslip right	179	603	27,647	5,523	854
Push pull	8	518	1,661	31,971	648
Stabilized-g turn left	1,361	1,357	3,366	2,814	25,908
Stabilized-g turn right	1,855	1,444	4,684	7,018	19,805

TABLE A-4. RESULTS FOR ROLL AND DUTCH-ROLL AT 95 KIAS

Maneuver	Classified as Dutch-Roll (%)	Classified as Roll (%)	Classified as Sideslip (%)	Classified as Push-Pull (%)	Classified as SGT (%)
Dutch-roll	83.6	9.9	4.8	1.3	0.4
Roll	2.7	85.6	5.3	4	2.4
Sideslip left	1.6	5.9	86.4	5.2	1
Sideslip right	0.5	1.7	79.4	15.9	2.4
Push pull	0.03	1.5	4.8	91.8	1.9
Stabilized-g turn left	3.9	3.9	9.7	8	74.4
Stabilized-g turn right	5.3	4.1	13.5	20.2	56.9

TABLE A-5. RESULTS FOR ROLL AND DUTCH-ROLL AT 95 KIAS

Maneuver	Percentage of "Correct Classification"	Number of Error of Prediction > 50με in Entire Files
Dutch-roll	83.6	1,000
Roll	85.6	2,990
Sideslip left	86	1,482
Sideslip right	79	2,520
Push pull	91.8	14
Stabilized-g turn left	74	51
Stabilized-g turn right	57	959

For completeness, different NN for sideslip left and right and for stabilized turn left and right should be tried. Improvement of predictions may be possible by sorting the data for each maneuver and speed and applying corresponding NNs.