Development of a parametric-based indirect aircraft structural usage monitoring system using artificial neural networks

S. C. Reed
Airworthiness and Structural Integrity Group
QinetiQ
Farnborough, UK

ABSTRACT

The development of a parametric-based indirect aircraft structural usage monitoring system using artificial neural networks is described. Flight parametric data, captured during Operational Loads Measurement have been used to predict strains or stresses at key structural locations for several military aircraft types, using mapping relationships determined by artificial neural networks. A framework for the development of a neural network-based structural usage monitor is discussed and the basic architecture of the multilayer perceptron artificial neural network is described. Additionally, results from case studies are presented. It is concluded that this technology could provide the basis for accurate, cost-effective structural usage monitoring systems across the range of military aircraft types and roles.

NOMENCLATURE

ANN	artificial neural network
ARD	automatic relevance determination
B	MLP bias vector
DAU	data acquisition unit
FLM	flight loads measurement
GR Mk1	ground attack and reconnaissance mark 1
Hz	hertz (event frequency)
IAT	individual aircraft tracking
MLP	multi-layer perceptron artificial neural network
MSE	mean squared error
NzW	normal acceleration conditioned by mass
ODR	operational data recording (rotary wing)
OLM	operational loads measurement (fixed wing)
p	roll rate
P	MLP input
PCA	principal components analysis
PTB	Tucano tailplane monitor location port side
purelin	linear activation unit (output layer)
P3B	Tucano port inner wing monitor location
P9B	Tucano port outer wing monitor location
RAF	Royal Air Force
RPROP	resilient back propagation training algorithm
SHAUNN	structural health and usage neural networks
SPC	sortie profile codes
S/s	samples per second (data capture frequency)
tansig	tangent sigmoid activation function
TMk1	trainer mark 1
u/c	undercarriage
UK	United Kingdom
W	MLP weight vector
W3L	Jaguar port outer wing monitor location
W5R	Jaguar starboard inner wing monitor location
Φ	MLP output vector

Subscripts

I	data line in the training set
j	specific weight/bias
M	MLP second hidden layer node
N	MLP first hidden layer node
n	number of weights and biases
OUT	MLP output node
w	weight/bias value.

Paper No. 3078. Manuscript received 8 February 2006, first revision 29 August 2006, second revision 17 November 2006, accepted 8 February 2007.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

1.0 INTRODUCTION

Military aircraft often operate, from necessity, in highly fatigue-damaging environments and close to limit load conditions; hence structural usage monitoring is essential to ensure that operations remain within acceptable levels of risk. However, for the majority of current fleets, limit-load exceedence monitoring relies on incidents being reported by aircrews and structural usage monitoring is only undertaken for the wing and those areas of the structure where strain correlates well with normal acceleration. For other structural areas, usage has generally been tracked in terms of flying hours or number of landings (for undercarriages).

With the ever-increasing capital and through-life support costs of military aircraft fleets, maximising an aircraft's reliability, availability and its safe fatigue life is paramount. Today, avionic and weapons systems and even engines can be replaced or upgraded and therefore, increasingly, the safe fatigue life of the aircraft structure will define the economic life-limit of the fleet.

However, within UK operational loads measurement (OLM) programmes[1,2,3] it has been shown that recording flying hours alone, or counting landings, can bear little correlation to fatigue damage accrual for these structural regions. Consequently, many military fleets have suffered significant costs of ownership[4,5] due to major fatigue inspections, component replacement, modifications or repairs and associated reductions in fleet availability. Furthermore, in some cases unforeseen structural failures have occurred[6]. Whilst a usage monitor for all critical structural locations would not negate the accrual of fatigue damage, it could provide a more accurate indication of structural risk of usage differing from design assumptions. It would also allow potentially increased safe-lives, optimised initial inspections and more efficient directed inspections thereafter.

A fleet-wide direct strain-based system would offer a solution in principle for most structural usage monitoring requirements and such a system is being introduced with the UK Eurofighter Typhoon fleet[7]. However, these systems are costly to develop and support; hence alternative, more cost-effective solutions should be investigated. Furthermore, the mechanisms of using a parametric-based individual aircraft tracking (IAT) system are well established in the UK; fatigue meters have been the backbone of UK military IAT systems for the past 40 or 50 years. Additionally, most of the inputs needed to predict strains are already available on a modern military aircraft, through data bus systems. Flight control, weapons management and fuel systems all need similar data; hence, in principle, an indirect structural usage monitoring system, using these parametric inputs, offers a cost-effective solution. Also, an indirect system can still have a role to play for strain-based fleet-wide monitoring systems. Parametric predictions could be used to provide back-up facilities for failed strain sensors pending repair. However, there are drawbacks and issues to address with indirect systems. Determining the relationship between parameters and strains, stresses or loads can be extremely complex. Also, methods identifying when those relationships have changed have to be developed. These issues are among those addressed within this paper.

The concept for the structural usage monitoring system described here would be to use data captured from the flight loads measurement (FLM) and operational loads measurement (OLM) (fixed-wing aircraft) or operational data recording (ODR) (rotary-wing aircraft) aircraft. An artificial neural network (ANN) based-system would be trained to predict the strains from flight parameters. Testing and validation of the ANN-based monitor would be undertaken by comparison between predicted and measured strains and calculated fatigue damages, using data not included in the training set; this would demonstrate that the ANN had achieved a good generalised solution. Once certified, such a system would be used to predict the strains seen on aircraft in the fleet not fitted with strain sensors, using the parametric inputs. Throughout the life of the fleet, data from the fleet OLM/ODR aircraft would be used for continued validation of the network. ANNs provide an attractive basis for the development of cost-effective indirect monitoring system because many of the relationships between parameters and strains are often non-linear and difficult to identify with more traditional regression techniques.

Researchers in the aeronautical arena have published work using ANNs and similar mathematical techniques to identify loading actions, or to supplement sparse flight test data. Azzam *et al*[8,9] described the use of a mathematical network to predict loads. Azzam *et al*[9] and Wallace *et al*[10] described co-operative work programmes involving Smiths Aerospace, the UK Ministry of Defence and BAE Systems and reported, among other successes, excellent correlation between measured and predicted strains at a number of structural locations for the Tornado combat aircraft, over a large number of test flights. Azzam[8] also described the development of methods to predict damage in helicopter components and the prediction of high frequency events, such as buffet loading on the fin of a fixed-wing aircraft. Jacobs and Perez[11] used a hybrid cascade ANN to augment sparse buffet data by predicting the power spectral densities of surface pressures on a fighter aircraft vertical tail, from parametric data. Kim and Marciniak[12] reported the use of a back-propagation ANN to predict strains in the empennage of a Cessna 172P general aviation aircraft during discrete manoeuvres. Hill *et al*[13] described the use of a back-propagation ANN to synthesise strain and fatigue damage in several Lynx helicopter components. Furthermore, Manry and co-authors[14] applied ANNs to flight load synthesis, using Bell helicopter data. Levinski[15] used ANN technology to synthesise loads on the vertical tail of a F/A-18 aircraft, using wind tunnel pressure data. Wang[16] used a recurrent ANN to predict loads during discrete manoeuvres for a fighter aircraft. Reed and Cole[17] and Reed[18] reported the development of an ANN-based parametric fatigue monitor for the wing and tailplane of a military trainer aircraft. With the exception of the work reported by Azzam and his associates[8,9,10] and Reed and Cole[17,18] the majority of the methods used were developed to predict discrete loading actions or manoeuvres rather than to predict the usage of the aircraft throughout its operational envelope.

Within the following sections, the architecture of the ANN core of the structural health and usage neural network (SHAUNN) is explained. Thereafter, a generic process for developing a SHAUNN system is outlined. Also, results from case-study demonstrator programmes are reported. Finally, conclusions are drawn and recommendations for further work are made. It is noteworthy that software development methods and practices are not discussed within this paper; a range of comprehensive standards and guidance, appropriate to the safety criticality of the software, already exists and is familiar to software developers.

2.0 ARTIFICIAL NEURAL NETWORKS

2.1 ANN definition

There is no universally accepted definition of what constitutes an ANN. However, a useful definition of an ANN is provided by Gurney[19]:

"… an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns."

In essence, ANNs have the characteristics that they can learn from experience, generalise from examples and identify underlying information from noisy data. Broadly, ANNs can be divided into two categories depending upon how they learn: supervised learning networks and unsupervised learning networks.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

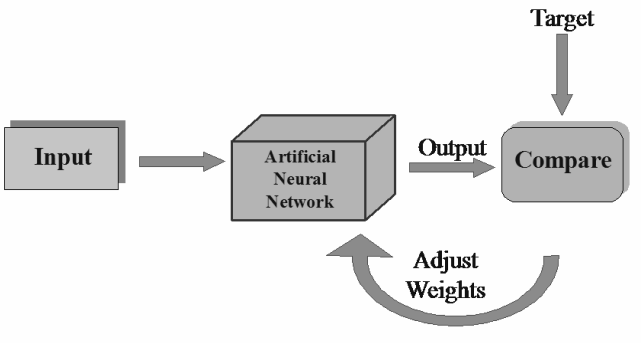


Figure 1. Supervised learning schematic for an artificial neural network.

2.2 Supervised learning ANNs

Supervised learning is the method proposed for the core of this application. In this technique the ANN is presented with input data (flight parameters) and examples of the required output data, or target (strains). The inputs are multiplied by weights and biases (discussed later) within the network, which are adjusted by iteration to reduce an error function. This error function is based upon the difference between the desired output and that predicted by the ANN. The concept is simple but the complexity is in the training algorithm that is required to converge the training to a minimum-error solution efficiently. Well-known examples of supervised learning ANNs include multi-layer perceptrons (MLP), used here, and radial basis functions. A simple schematic illustrating supervised learning is reproduced in Fig. 1.

2.3 Unsupervised learning ANNs

Unsupervised learning techniques allow the ANN to determine the characteristic relationships within the data without reference to output or target examples. Information about the characteristic features of input data is created during the learning process and stored in the weights. Output signals describe relationships between the current input signals and the weight vectors. Well-known unsupervised techniques include Kohonen networks and learning vector quantisation (which incorporates both unsupervised and supervised architecture).

2.4 Multilayer perceptron architecture

The multilayer perceptron (MLP) is one of the most commonly used ANNs and is now almost regarded as a traditional method[19]. The MLP was the primary ANN method used in the SHAUNN and the key features of this system are described in the following sections.

2.5 Weights and biases

A schematic of one of the MLP ANNs used for this work is provided in Fig. 2. The input nodes are P_1 …. P_x. These nodes represent the input parameters (processed flight parameters in this case). The inputs are each linked to the first hidden-layer neurons by weights (**W**) and a bias (**B**) is applied to each of the neurons. These weights and bias can be thought of as analogous to the '$y = mx + c$' equation in linear regression. The weight can be considered as the 'm' and the bias as the 'c'.

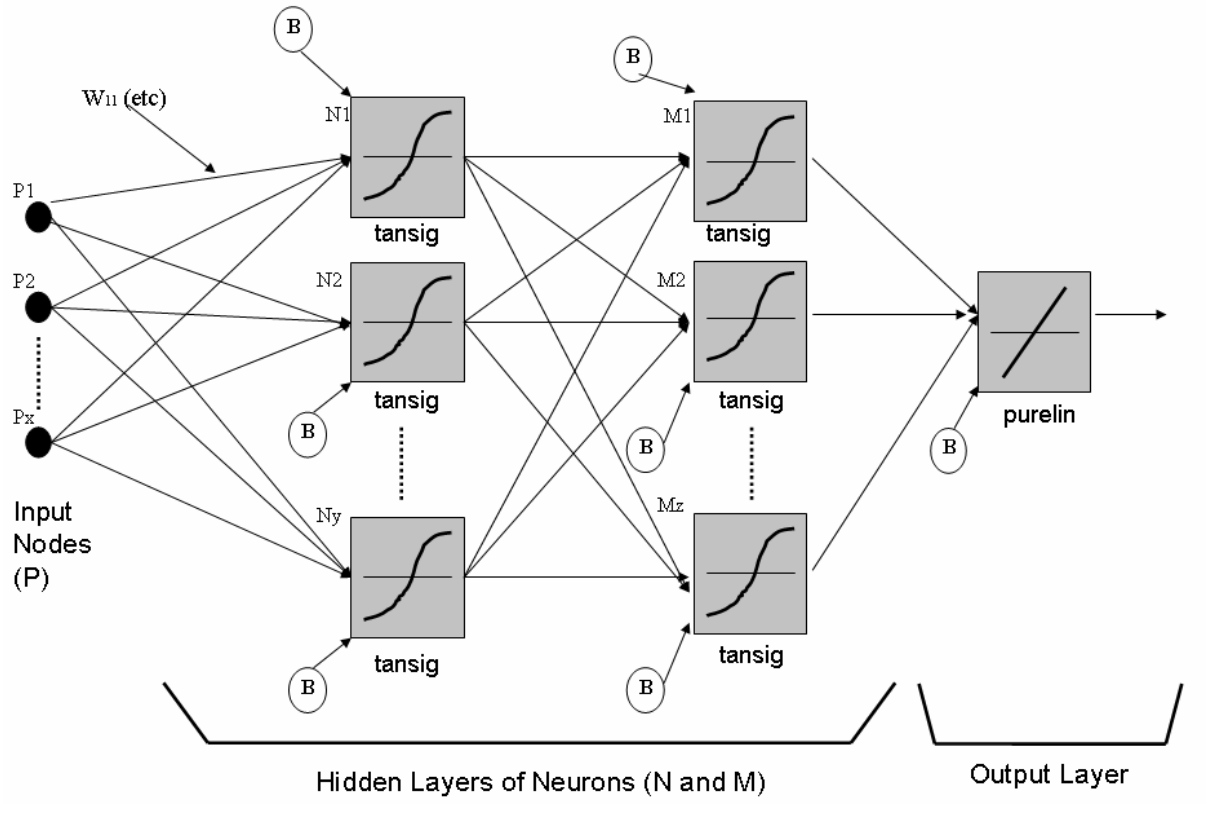


Figure 2. Example SHAUNN MLP architecture.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

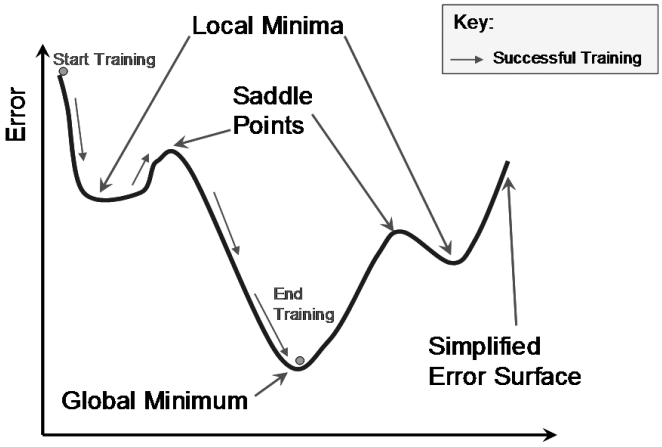


Figure 3. Simplified schematic of error surface.

2.6 Hidden layer activation functions

A MLP is a feed-forward network and the hidden layer (or layers in the illustrated example in Fig. 2 contains a number of neurons that receive weighted data from each input node and pass it to the next layer node, via an activation function. The function of each of these neurons is akin to a mechanical non-linear gearbox. The relationship between the input and output is non-linear but constant. Therefore a particular input always produces the same output and the shape of the non-linear activation, or squashing function, can be varied to suit the data. In the network illustrated below a tansig function was used, which is symmetric about zero. Alternative activation functions, such as the non-symmetric-about-zero logsig functions, are also used in MLP networks. Various other shaped functions can be used as long as they are continuous differentiable monotonically non-decreasing functions.

2.7 Output layer

Data from each neuron in the final hidden layer is passed to the output layer. In the example illustrated in Fig. 2, the output layer is a simple linear summation (purelin) of the final hidden layer, again with a bias, to a single output. More than one output unit can be incorporated and alternative functions can be used depending upon the output requirement, such as a logsig function if the output is required in the range 0-1.

2.8 Network initialisation

The weights and biases within the network are set to initial conditions before the training process begins. The choice of initial weights will influence whether the ANN reaches a global minimum in the error function or just a local minimum[20]. Furthermore, the speed at which it reaches this minimum will be governed partially by the value of the initial weights. Hence, it is common to use randomised initial weights and biases, often in the range –0·5 to 0·5. As the initial weights and biases can have such a significant effect on the final convergence of the ANN, it is usual to train a model a number of times using a seeded weight randomisation function and identify the model that produces the best generalised solution from testing with unseen data.

2.9 Training the ANN

2.9.1 Training algorithm aims

Once the ANN architecture is established, the network needs to be trained. Training in this application is a supervised learning process. The aim of the training algorithm is to reduce the error function to a minimum. This error function is usually the difference between the desired output (or target) and that predicted by the ANN, often with the addition of a regularisation term to prevent weights getting too large. Training the ANN can best be visualised by using a simplified two-dimensional error surface, as illustrated in Fig. 3. Here the aim is to reach the global minimum value in the error surface without getting stuck in local minima or saddle points.

Errors are fed back through the weights and biases using a method called back-propagation. (Hence, a MLP is often referred to as a feed-forward back-propagation algorithm). It was the development of a method of feeding back errors through the network that signalled the leap forward in neural network applications seen in the last 15-20 years.

A detailed discussion of the plethora of ANN training algorithms presented in the literature is beyond the scope of this paper. Readers wishing for a more detailed and mathematical explanation of ANN training algorithms are recommended to read Fausett[21], Hagan *et al*[22], Haykin[20], Bishop[23], Riedmiller and Braun[24] or Mackay[25].

In the vast majority of methods used, training, test and validation data sets are used within the training process. The training data set is used to optimise the weights and biases within the model. The statistically similar test data set is used to ensure the solution is not over fitting to the training data and to ensure the solution will still produce good generalisation. Thereafter, the validation data set is used for independent assessment of the performance of the model, with data not seen during the training and testing process.

2.10 Adaptive and non-adaptive systems

The training methods described here are non-adaptive prediction methods; after training, the weight and bias coefficients are fixed. Alternatively, in adaptive prediction methods the system continues to train itself when it is in service use. Without formal control over the training data set, certification of such a mechanism for a safety related applications, such as an aircraft usage monitoring system, would be highly unlikely and hence these adaptive networks are not considered suitable as the basis of structural usage monitoring systems.

2.11 Recall mode

Once trained, a non-adaptive ANN, such as that illustrated in Fig. 2, would have fixed weights and biases. Hence, an array or matrix equation would be written as in Equation (1):

$$\Phi = \sum \left[Tansig \left\{ \mathbf{W}_M * \left(Tansig \sum \left(\mathbf{W}_N * \mathbf{P} + B_N \right) \right) + B_M \right\} + B_{OUT} \right] \quad \ldots (1)$$

where $\mathbf{\Phi}$ is the output, Tansig is the non-linear transfer function, $\mathbf{W}_M$ and $\mathbf{W}_N$ are the weights for the hidden layers M and N, $\mathbf{P}$ is the input data (parameters), and $\mathbf{B}_N$, $\mathbf{B}_M$ and $\mathbf{B}_{OUT}$ are the biases for the *N*, *M* and output layers.

Within this section, the basics of the ANN core of the SHAUNN have been described. In the following section a development process for using a SHAUNN system within an aircraft usage monitor is detailed.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

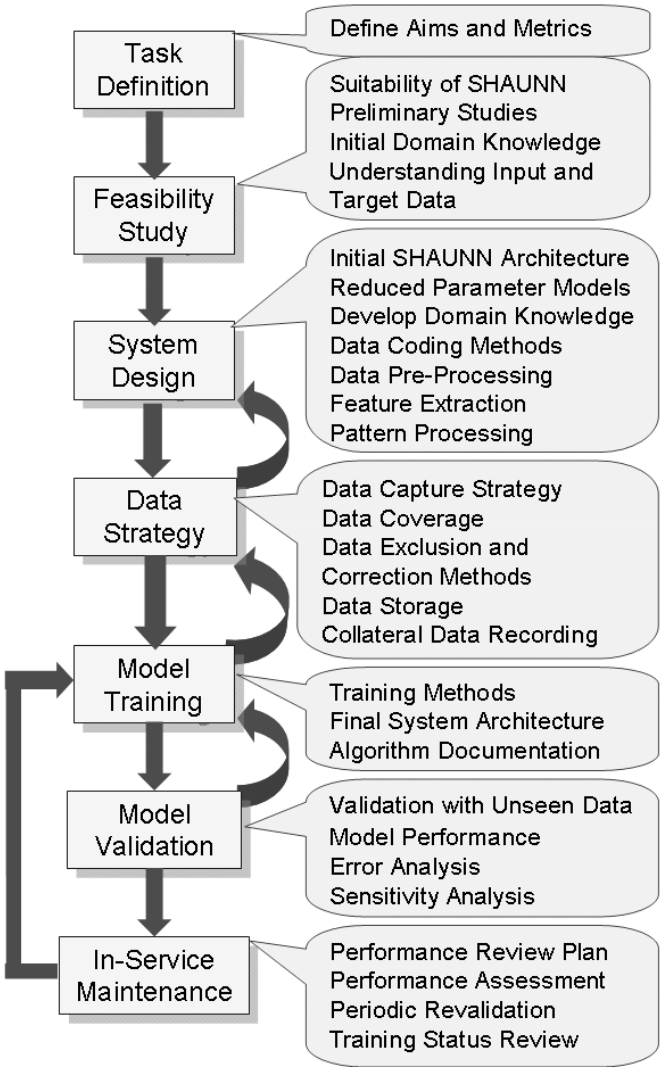


Figure 4. Example SHAUNN development schematic.

3.0 DEVELOPING STRUCTURAL HEALTH AND USAGE NEURAL NETWORKS

3.1 Development process

The high-level schematic at Fig. 4 provides an example framework within which a SHAUNN-based monitoring system could be developed. Throughout the remainder of this paper, in the interests of brevity, the process of predicting, or mapping, strains, stresses or loads from parametric data is described as mapping strains from parameters; additionally, reference to OLM data should be interpreted as OLM/ODR/FLM data.

3.2 Task definition

The task definition should be clear and concise and should include the identification of the structural monitoring locations, the monitoring method and process, the required performance metric for the system, the data coverage required and any *a priori* constraints on the system. These constraints might include: limitations on the quantity, quality, accuracy or resolution of the parametric or strain data available for training, testing and validation, or the presence of high-frequency structurally significant loads in monitor locations. Additionally, it may not be practical to position strain sensors at the key structural locations due to high strain gradients, local geometry, or the effects of maintenance traffic in the areas concerned; these limitations should be understood at the outset and accounted for in the design.

3.3 Feasibility studies

3.3.1 Suitability of SHAUNN

A feasibility study should be used to identify the provisional complexity requirements for the monitoring system and to identify whether a SHAUNN-based monitor is required at all. The aim should be to accord with Occam's Razor[26] wherever possible; i.e., if all things are equal then the simplest solution should be chosen. If one or more of the flight parameters show a high correlation with strain, an ANN-based system may not be necessary and traditional regression techniques may be sufficient to provide a viable mapping solution. It will be more difficult and costly to certify an ANN-based monitoring system than a parametric system using traditional regression techniques and hence the use of ANNs should be justified (see Case Studies).

3.3.2 Initial domain knowledge

Initial domain knowledge should be captured during this phase of the task. Domain knowledge is the specialist or a priori knowledge appropriate to the system. Hence the SHAUNN prediction model can be used to determine the unknown relationships between parameters and strains, rather than learning a parametric data input relationship already known to the system designer. For an aircraft usage monitoring system, domain knowledge could be obtained from a variety of sources including: aircraft loads models, equations of motion and previous aircraft parametric monitor inputs. An illustration of the effect of applying domain knowledge to the system is provided in Case Study 1.

3.3.3 Understanding input and target data

Work within the feasibility study should consider the source specifications of the parameter and strain data instrumentation. For projects where data have already been captured, it may not be possible to influence the specification of the data but, in these cases, the limitations of the data should be identified.

The success of the monitor will ultimately be determined by the accuracy of the strains predicted from flight parameters but most likely assessed in terms of fatigue damage. However, the accuracy requirements have to be consistent with the source data (i.e. strains and parameters) and a building-block approach to identifying expected accuracy should be used. At the base level, ideally strain sensors need to be located in regions of low strain gradient. Where this is not possible, an understanding of the local strain field is essential and even tighter location accuracy requirements should be invoked for sensor placement. Where data are to be sourced from a number of aircraft, which is highly desirable, the location of the strain sensors needs to be repeatable and controlled. Calibration of the sensor outputs is essential. All parametric data also needs to be sourced from calibrated sensors with appropriate accuracy requirements.

Reliability of sensors should be included within this analysis process. Electrical resistance strain gauge systems have suffered reliability problems on some OLM programmes[27]. Failure of these

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

sensors has been due to a variety of reasons: poor surface preparation, poor installation, poor sensor protection, vulnerable locations, unsympathetic maintenance activity, sensor drift and moisture ingress. OLM strain sensor data will be captured routinely from a small proportion of the fleet for the life of the aircraft and these data would be used to provide continuous validation and retraining data for the SHAUNN and hence analysis of sensor system reliability performance is essential. Expected reliability of proposed parametric sensors also should be established. This is particularly important where the sensor is used to provide data to support the SHAUNN system alone. Where structural monitoring systems can piggy-back upon other systems, such as mission or flight control systems, then sensors failures will be identified and repaired relatively quickly. However, where sensor failures have no direct operational effect, prompt repair is less likely. It is highly unlikely that a structural monitoring system failure would be a 'no-go'[28] item for a military aircraft. Hence, it is inevitable that operations with failed or degraded systems will occur and will accrue additional penalties for unmonitored flying (1·5 times the average damage for the recorded sortie type for UK military aircraft). With 1% of the fatigue life of a military fleet estimated in 1998 as costing in excess of £100m[29], the financial penalty for prolonged periods of unmonitored flying is extremely significant.

Additionally, the interaction between the sensor, data bus and data acquisition unit (DAU) should be carefully specified for new designs, or understood for extant systems. The rate at which data appear on the data bus may not necessarily reflect sample rates or instrument bandwidths as data may be stale. Additionally, for DAU integrated within mission systems data may be sampled at opportunity, depending on the processor work load. Anti-aliasing filters within the DAU can affect the synchronisation of data depending upon the filter architecture, frequency content in the data and sample rate. Furthermore, where data are collated from a number of aircraft systems, time synchronisation methods and limitations should be identified and understood. All these potential sources of error should be understood and taken account of in the design and in the formulation of design accuracy requirements; there is no mileage in setting unachievable accuracy requirements. Azzam[8,9] refers to this process as establishing 'truth margins'.

3.4 System design

3.4.1 Initial SHAUNN architecture

The system design phase should be used to identify the initial proposed architecture of the system, as developed from the feasibility study. Additionally, the system domain knowledge should be developed further and data coding and data manipulation methods established. For a MLP ANN, definition of the initial number of hidden layers and the initial number of neurons in each layer, as well as connectivity with the network and definition of the activation functions will be required. The size of the hidden layer (or layers) is a trade-off between accuracy and generalisation. The number of neurons in the hidden layer must be sufficient to represent a function or classify data adequately but low enough to allow generalisation.

In practice, the most suitable architecture for a SHAUNN monitor will take considerable iteration to identify and it is likely that a great many models will be needed to identify optimum architectures. There is no foolproof method for determining the number of neurons in the hidden layer or layers a priori. Many researchers[20,23,30,31] have proposed methods for starting this largely iterative process. These are largely heuristic 'rules' and the optimum solution will be very dependent upon the data sets in question.

The risk of over fitting the solution to the training set is well documented in the literature[19-23] and several methods are described to reduce this risk. These include: restricting the number of neurons in the model, using early stopping techniques to identify when the error in predicting a statistically similar data set to the training set is increasing while the error on the training set continues to reduce, and the use of Bayesian methods[32]. Many of these methods have been experimented with during this work and are reported in the attached case studies.

3.4.2 Reduced parameter set models

Development of reduced parameter set models should be considered to prevent loss of the monitor due to failure of several of the less significant input parameters. Methods such as automatic relevance determination (ARD)[23,33] can be used to identify those inputs with the greatest significance. This method uses a Bayesian framework to identify the relative significance of an input to the network output based upon the overall variance in the weights relating to each input parameter (Case Study 1).

3.4.3 Further development of domain knowledge

The initial domain knowledge incorporated into the feasibility study should be developed further during the system design phase. Furthermore, differences in sortie types, flight phase or state of the aircraft may necessitate the development of several different models, if the underlying relationships between flight parameters and strains within these flight phases or states vary significantly.

3.4.4 Data pre-processing methods

Data pre-processing tasks are defined here as data cleansing and data normalisation. Data cleansing is used to detect and repair or exclude anomalous data. Maximum use of integrity indicators in the data streams, such as synchronisation words and bit checks should be used to confirm data validity. Thereafter, deterministic validation rule methods incorporating engineering, or domain knowledge, should be developed. More sophisticated methods, such as outlier analysis and novelty detection[34] methods should also be considered.

Experience has shown that, while well written anomaly detection algorithms will capture most of the anomalies in the data, they are unlikely to be fool proof. Hence, there is no substitute for 'eyeballing' of the training data by experienced engineers. Furthermore, great care should be taken to ensure that 'good' data are not rejected as being anomalous because they do not match pre-conceived ideas of the expected data sets. All corrections to data should be documented and preserved along with the data. Finally, the anomaly detection algorithms applied to the development data sets should be consistent with those used in service.

Data normalisation methods are used to match the range of the input data to the non-linear activation functions in the hidden layers and to prevent scale error effects due to the large variation in range of the inputs to the network. A variety of methods are used, including normalisation to zero mean and unit variance (often termed z-score scaling), or between predefined input limits, such as 0·1 and 0·9. The method used to normalise the data should also be consistent with any dimensionality reduction methods employed.

3.4.5 Feature extraction methods

Feature extraction refers to the process of extracting distinguishing features from the data. In this work this process has been undertaken by using two distinct approaches.

The first has been to view the input data from an engineering perspective and select significant elements of the data sets for inclusion in the training data, such as high damage rate flight segments. Additionally, it is desirable to reduce redundancy from within the data. This process is termed dimensionality reduction and can be achieved in a number of ways. Input variables can be removed that carry the least predictive power. In practice, identification of the

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

least predictive variables in a multidimensional space can be problematic, although significances can be determined using methods such as ARD[33],discussed previously. Alternatively, high-dimensional input data can be projected onto a lower dimensional space. Principal component analysis (PCA)[35] is a well-known tool used for this process. Input data are resolved into principal components and resolved components that contribute most to the variance of the data are preserved (Case Study 2 and 3). PCA is a linear transformation of the input data and as such there is a risk of loosing non-linear relationships within the data due to the transform. Alternative non-linear transformation methods, such as Sammon mapping, are also discussed in the literature[36].

The second method of feature extraction is to use an auto-associative network[36]. This method involves constructing a MLP ANN where the target output is the same as the input. The hidden layer (or layers) containing less neurons than the input or output layers acts as the feature extractor. The inputs to the prediction method are then generated by entering the full input vector into the network and extracting the outputs of the hidden layer with the least number of neurons.

3.4.6 Pattern processing methods

The pattern processing undertaken by the core SHAUNN will either be a regression task or a classification task. Regression tasks dominate the case studies presented here, where the SHAUNN is used to predict strains from flight parameters.

3.5 Data strategy – training, testing and validation data

3.5.1 Data-capture strategy

The data strategy should include identification of the data requirements, data sources, estimated data quantity requirements and data limitations. An ANN-based monitor will require large quantities of data throughout every stage of its development. Experience has shown that the capture of sufficient quantities of OLM data to undertake usage analysis generally takes far longer than originally anticipated. This can be due to delays in contracting, embodiment, equipment un-serviceability and operational reasons and this should be taken into account when developing a data capture strategy.

For aircraft entering service it will probably be necessary to develop provisional SHAUNN models, using flight loads measurement (FLM) data. FLM flying is unrepresentative of in-service flying but it can exercise areas of the flight envelope not routinely visited during in-service operations. Care should be taken to understand any differences in the data provided from FLM as compared to OLM data, such as gauge locations, signal conditioning methods and flight parameter instrumentation fits. It is essential, as far as practicable, that the SHAUNN data are drawn from throughout the flight envelope, to ensure all relationships between input and output parameters are captured. In practice, there may well be regions of the flight envelope where flight data cannot be captured, even from FLM data, and hence why data coverage indicators are required (following section).

Once sufficient OLM becomes available, additional model training should be undertaken and, if required, the fatigue damage algorithms should be re-run with revised predicted strains. This should be undertaken as part of the usage monitor validation exercise.

3.5.2 Data coverage

The population of the training, testing and validation data sets should be described within the data strategy. Care should be taken to ensure that the various aircraft configurations (e.g. stores, fuel tanks, role fits) are captured adequately within the data sets. Specialist knowledge, particularly from loads experts within aircraft design teams, is important for understanding type-specific information. Data should also be collected from a number of aircraft. Additionally, as part of the validation process (discussed in later sections), where practicable, the models should be tested on data captured from aircraft not used in the training data set, to illustrate model generalisation (Case Study 1). The statistics of the training, testing and validation data should be identified and documented. These statistics should be used to establish the 'properties' of the training, testing and validation data sets for future comparison with in-service data.

There is no straightforward method of determining how many data are required to develop a SHAUNN monitor. The number of training cases (input/output pairs) is defined by the complexity of the application, the intrinsic dimensionality of the data, the resolution required and the frequency distribution of the data. Additionally, the quality of the data and presence of ambiguous inputs also should be considered. Most importantly, evidence of the coverage of the structural monitor from training data should be provided.

Identification of the data coverage required and achieved is not a trivial matter for a multidimensional problem. Hence, simplification should be made to visualise this coverage. For some locations, such as wing-root bending, several key parameters should drive the strain prediction (e.g. NzW and roll rate (p)). For such locations, a three-dimensional envelope density plot could be used to provide an indication of data coverage. For more complex relationships, similar plots using the first two principal components of the data could provide similar information.

3.5.3 Data exclusion and correction methods

Methods used to check the integrity of the data should be applied across all data sets and documented accordingly. Assessment of the quality of the data to be presented to the model is essential. All raw data should be stored and a record of all data anomalies and corrective action taken should be retained. Flight data from any source will not be perfect and where there is noise in the data then a larger quantity of data will be required to train the model.

3.5.4 Data storage

The development of a SHAUNN-based monitor will necessitate the storage of significant quantities of raw and processed flight data. It is essential that a data storage facility is designed such that these data can be retrieved and all processes applied to the data from its raw origins can be identified and reproduced.

3.5.5 Collateral data collation

All data used in the development of SHAUNN models will have associated collateral information. This information will include the source of the data, flight date, sortie type, role or stores fit, aircraft flight control or engine control software standard etc. These data, which can provide essential information to the system designer, particularly in the processes of selection of development data and the analysis of prediction errors, should be captured, collated and accessible throughout the development process.

3.6 Model training

3.6.1 Training methods

The large number of training algorithms documented in the literature leads to the conclusion that there is no universal training algorithm and the most appropriate method will depend on many factors, including the dimensionality of the problem and the complexity of

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

the error surface. The performance of several frequently used training algorithms is provided in Case Study 3. Whichever method is chosen by the designer, it should be documented clearly, together with supporting material, including the model performance, methods used to prevent over fitting and indications of the health of the network.

For SHAUNN models, a randomisation function may be used to set the initial weights for training. In this case, a number of models are required as some may get stuck in local minima on the error surface. Thereafter, only the best performing model (using the test data) would be used to compute the performance metric using the validation data. Alternatively, Mackay[32] extols the virtue of forming a committee of the best performing models using a test data set and using the average prediction of these models (Case Study 1).

3.6.2 Final system architecture

In practice, the process of developing the SHAUNN model will be highly iterative. However, eventually the final system architecture will be identified that best meets the specification for the structural monitor. At this point the SHAUNN system architecture should be clearly documented, including architecture and algorithms used for data conditioning, anomaly detection, pre-processing and post processing methods.

3.7 Model validation

3.7.1 Validation with unseen data

Validation of the model using data independent of the training and test set is required. Although the test set is independent of the training set, it will have been used to select the best performing models during development. Validation data should be obtained from a number of aircraft and should, where practicable, include data from aircraft not used to populate training or test data sets. Model validation with these data will illustrate the model transferability within the fleet (Case Study 1).

3.7.2 Model performance

The performance of the SHAUNN monitoring system, using the validation data set, should be clearly documented. The performance metric of the monitor will be compared with the requirements specification and these requirements are likely to be expressed in terms of cumulative fatigue damage. However, the performance of the model in predicting strains (i.e. strain correlation coefficient) and fatigue damage accrual on an individual flight and through-flight basis should also be undertaken. Analysis of these data will assist greatly in the understanding of the source of errors produced by the model.

3.7.3 Error analysis

One of the key challenges in certifying a neural-network based usage monitor will be providing evidence of the confidence in the predictions made. To obtain this confidence, an understanding of errors generated in design, validation and in-service phases of the monitor development is required. These errors may be caused by one or more of three factors: the data may not contain sufficient information to characterise adequately the mapping relationships; the model may have failed to capture the relationships adequately; model validation data may include relationships not contained in the model's training data. Identification of the prime source or sources of error will allow iteration of the model inputs, model architecture or training data set to minimise this error.

Therefore, for a SHAUNN monitor to be deployed, it is essential that the models provide an indication of the confidence or confidence interval in the prediction or the likely error in the output. There are a variety of methods published in the literature[31] for identifying confidence intervals. Also, methods of producing error bars within a Bayesian training method[32] are used (Case Study 1).

3.7.4 Sensitivity analysis

Although anomaly detection and data novelty detection processes should be in place it is still necessary to have an appreciation of the sensitivity of the SHAUNN monitor to input variations. Sensitivity analysis[30] is a process whereby small perturbations are applied to each input in turn, while keeping the remainder constant, and identifying the output variation, often leading to production of a sensitivity coefficient for each input-output pair.

3.8 In-service maintenance

3.8.1 Performance review plan

Before the monitor enters service it is essential that a performance assessment mechanism and a performance review plan are identified. Anomaly reporting mechanisms have to be extended to cater for situations where the software is operating correctly but the prediction is questionable but quite possibly credible. A continued monitoring plan is therefore required. Within this plan the measures already described for data anomaly, confidence intervals and data novelty should be detailed and methods for undertaking trend analysis and setting performance warning levels should be specified. This should not, in principle, be significantly more onerous than the measures needed to supervise the performance of 'conventional' parametric structural monitoring systems.

3.8.2 Continual performance assessment

One of the key issues for in-service support will be to identify when structural monitoring models require retraining. Until a system is in use it is not possible to identify how often retraining will be needed. If well-optimised models are produced initially, retraining should only be necessary when either significant change in usage occurs or if structural configuration changes in the aircraft affect the underlying flight parameter to strain relationships. However, measures should be in place to support continual performance assessment to identify when review of the models should be undertaken. Within this continual performance assessment, novel data should be identified and this should be coupled with identification of predictions with low confidence. The use of warning levels, based upon confidence and novel data criteria, to identify model review requirements should be included. Trending of results for similar sortie types using criteria such as strain range, damage accrual and steady state values should also be undertaken.

3.8.3 Periodic revalidation

The continual performance assessment system should be designed to trigger warnings when retraining of the SHAUNN monitor may be required. However, this system will be dependent upon the appropriate setting of trigger values. Therefore, in addition, periodic revalidation of whole usage monitoring process should be planned and undertaken. Usage monitor validation is one of the prime aims of OLM programmes and UK military requirements[37] mandate periodic or continuous capture of OLM data. Hence, these OLM data should be used for the periodic planned re-validation of the SHAUNN-based monitor.

In this section of the paper a framework for developing a SHAUNN-based fatigue monitor has been described. This

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

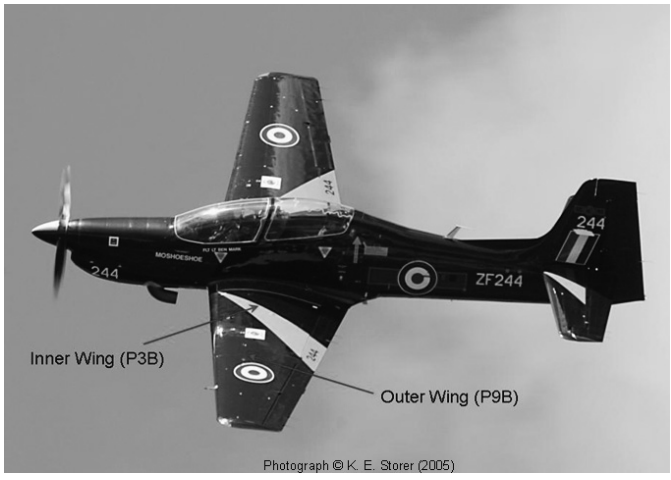


Figure 5. Tucano TMk1 wing strain mapping locations.

framework[38] has since formed the basis of a new inclusion in the UK Defence Standard for the Design and Airworthiness Requirements for Service Aircraft[39]. In the remaining sections, a selection of case studies is presented in which many of the techniques described in the previous sections have been used.

4.0 CASE STUDY 1 – MILITARY TRAINER INNER AND OUTER WING MONITOR FEASIBILITY STUDY

4.1 Case study aim

The aim was to predict bending strains for the inner and outer wing of the Tucano TMk1 military trainer. The target validation performance was an average correlation coefficient between predicted and measured strains of 0·995 and cumulative fatigue damage prediction within 10% of that calculated from the measured strains.

4.2 Training data

4.2.1 Data source

The data were sourced from the RAF Tucano TMk1 OLM aircraft ZF406 and ZF512.

4.2.2 Strain mapping locations

The port rib 3 inner wing bending bridge (P3B) and the port rib 9 outer wing bending bridge (P9B) were selected as the monitoring location (Fig. 5).

4.2.3 Flight parameter data set

The flight parameter data set available is illustrated in Table 1. Parameters were sampled at 16 samples per second (S/s), except normal acceleration which was sampled at 32 S/s.

4.2.4 Data set limitations

The data did not include outside air temperature; hence an international standard atmosphere was assumed for Mach number calculations. Roll and pitch rate were estimated by integrating differential accelerations. Angle-of-attack was estimated from a lift curve slope equation, with validation limits.

4.2.5 SHAUNN justification

Justification for developing a SHAUNN model was undertaken by correlating the mass corrected normal acceleration (NzW) with the measured strain time history. An average correlation for NzW to strain for the data set was 0·927, which was significantly below the target requirement of 0·995.

4.2.6 Application of domain knowledge

Analysis was undertaken to identify the effects of applying domain knowledge (Table 2). Acceleration terms were conditioned by mass (from Newton's Second Law of Motion), Mach number terms were included, dynamic pressure was also added and used to condition control deflections (lift being proportional to dynamic pressure and control surface deployment). Inner wing models without domain knowledge produced average strain correlations of 0·986, compared with 0·995 for the models with domain knowledge.

4.2.7 Training data set feature extraction

A range of ANN models were trained on data from ZF406 and their performance tested on data from ZF406 and ZF512, to provide evidence of model generalisation and transferability. Initially performance was identified in terms of the Pearson correlation coefficient between the measured and predicted strain values in the data set. Better performance was obtained from models trained on aggressive flights; hence, aerobatic sorties, at a range of masses, were used as the final training set. A feature extraction method was used to select highly damaging segments from within these flights.

Table 1
Parameter and discrete values available from Tucano OLM data

Airspeed	Altitude	Elevator Angle	Aileron Angle	Rudder Angle	Tail Normal Acceleration
Port Normal Acceleration	Normal Acceleration	Lateral Acceleration	Engine Torque	Fuel Total	Heading
Flap Position	Weight on Wheels (Discrete)	Airbrake Selection (Discrete)	U/C Position (Discrete)	Compass Warning (Discrete)	Engine >80% Switch (Discrete)

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

Table 2
Tucano input parameters – domain knowledge applied

Symbol	Combined Parameter
NzW	Normal Acceleration x Mass
PAILxDYP	Port Aileron Deflection x Dynamic Pressure
NzWMach2	Normal Acceleration x Mass x Mach^2
NzWMach	Normal Acceleration x Mass x Mach
RACCxW	Roll Acceleration x Mass
PACCxW	Pitch Acceleration x Mass
RRATxDYP	Roll Rate x Dynamic Pressure
RRATxDYP /TAS	Roll Rate x Dynamic Pressure / True Airspeed
MACH	Mach Number
MACH^2	Mach Number^2
DYP	Dynamic Pressure
ALPHxDYP	Angle of Attack x Dynamic Pressure
FLAPxDYP	Flap Angle x Dynamic Pressure
WOW	Weight on Wheels

4.3 Model generation

The models were created using Neuromat® software[40]. The training data set consisted of 20,000 examples with 14 input parameters. A single hidden layer of tansig neurons and a purelin output neuron was used; training was undertaken using Mackay's Bayesian framework[32]. Committees were formed from the seven best performing models from the over 250 models produced from various weight-initiation seeds and hidden-layer complexities. The input parameter significances for models, calculated using automatic relevance determination, are reproduced in Fig. 6 and Fig. 7. Models were identified by the number of neurons in the hidden layer (A-Z representing 1-26 neurons) and the seed value (1-9).

The models were dominated by airspeed-orientated parameters (dynamic pressure and Mach number) and less so by the normal acceleration orientated parameters. This is most likely because the data still contained a significant number of examples where the aircraft was flying straight and level or on the ground and strain variations would be primarily associated with dynamic pressure changes around the aircraft.

4.4 Results

Performance of the inner and outer wing models using the 8-flight validation data in terms of strain correlation and fatigue damage comparison is produced in Table 3. The fatigue damage comparison was made by rainflowing[41] the measured and predicted strain time histories and calculating the Miner's sum damage using a L65 Aluminium Alloy S-N Curve for a low-load transfer joint, using data from ESDU 89046(42).

An example of a validation run for the inner wing model, for flight 5122049, is reproduced in Fig. 8.

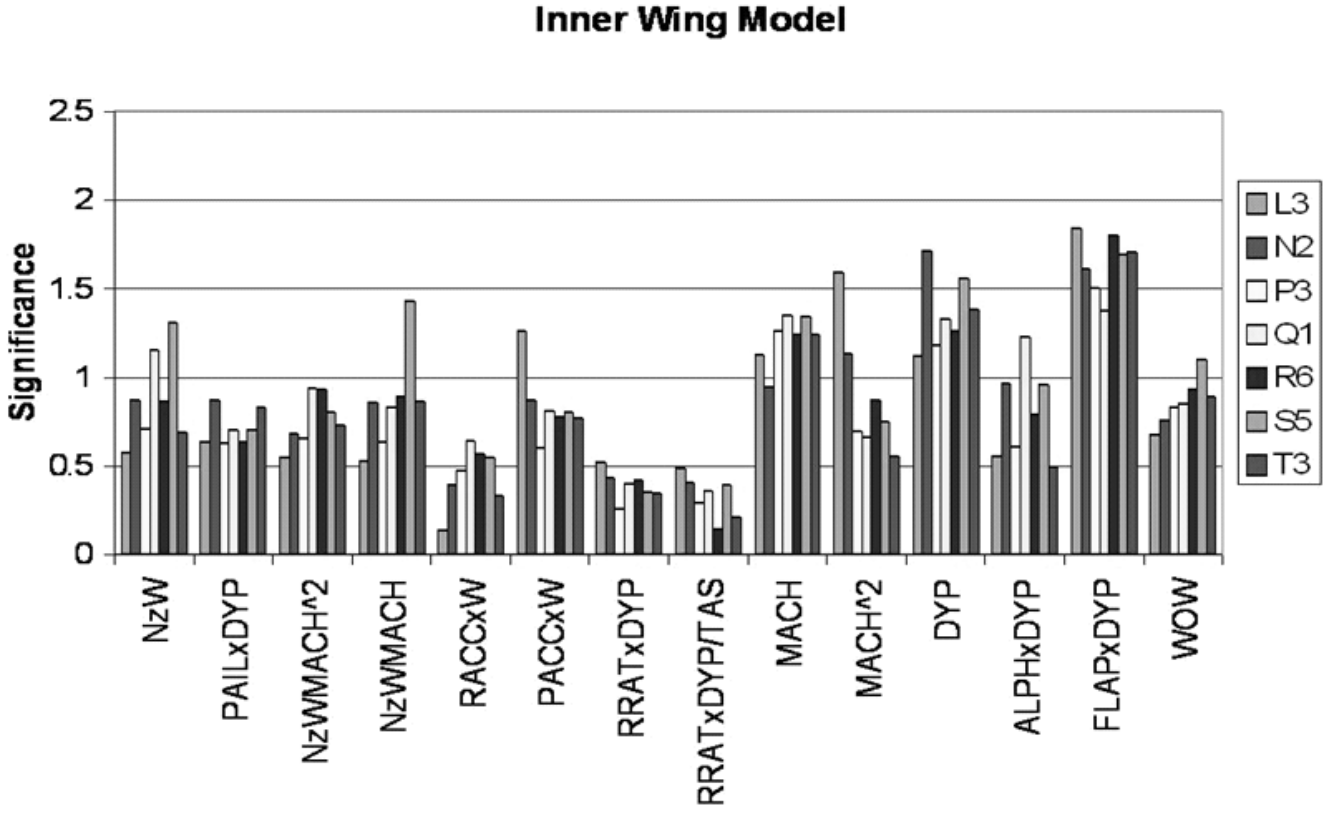


Figure 6. Inner wing (P3B) input parameter significances for model committee.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

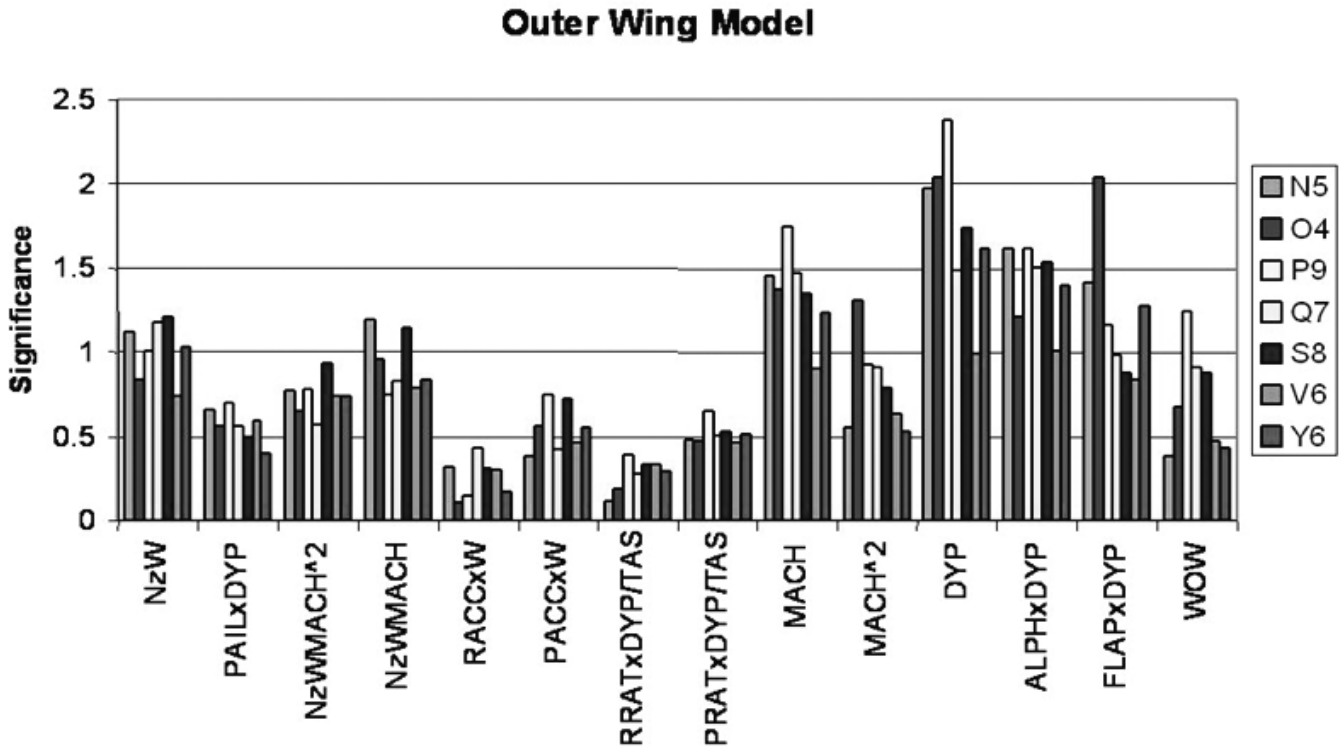


Figure 7. Outer wing (P9B) input parameter significances for model committee.

The average correlation between predicted and measured strains, using validation data, for the inner wing was 0·995 and 0·997 for the outer wing; these values meet the performance metric aims. The lowest correlation was 0·992 for flight 4061768 on the inner wing. This was an early flight in the OLM programme and several changes were made to the parametric calibration equations during this period.

Table 3
Inner wing (P3B) and outer wing (P9B) validation performance

Inner Wing Model Validation Flight	Strain (P3B) Correlation Coefficient	Damage from Predicted Strain (x1E6)	Damage from Measured Strain (x1E6)	Predicted / 'Measured' Damage
4061768	0.992	139.5	108.6	1.28
4061956	0.994	130.3	122.8	1.06
4061962	0.995	259.2	241.5	1.07
5122028	0.994	74.7	105.1	0.71
5122045	0.997	206.1	214.9	0.96
5122049	0.994	240.1	216.6	1.11
5122127	0.996	232.0	186.6	1.24
5122137	0.996	625.9	659.7	0.95
	Average	Total	Total	Total
	0.995	1907.8	1855.8	1.03

Outer Wing Model Validation Flight	Strain (P9B) Correlation Coefficient	Damage from Predicted Strain (x1E6)	Damage from Measured Strain (x1E6)	Predicted / 'Measured' Damage
4061768	0.997	9.3	8.7	1.07
4061956	0.996	10.4	9.7	1.07
4061962	0.998	22.9	21.6	1.06
5122028	0.994	6.3	4.9	1.29
5122045	0.996	15.2	15.4	0.99
5122049	0.997	19.3	20.2	0.96
5122127	0.998	10.6	12.6	0.84
5122137	0.998	51.5	57.1	0.90
	Average	Total	Total	Total
	0.997	145.5	150.1	0.97

Although validation would require a larger data set, the cumulative fatigue damage error was a 3% over prediction for the inner wing and a 3% under prediction for the outer wing. This level of accuracy was within the target aims of the programme (±10%).

5.0 CASE STUDY 2 – COMBAT AIRCRAFT INNER AND OUTER WING MONITOR FEASIBILITY STUDY

5.1 Case study aim

The aim was to predict stresses for the inner and outer wing of the RAF Jaguar GR Mk1 combat aircraft. The target validation performance was an average correlation coefficient between predicted and measured stress of 0·995 and cumulative fatigue damage prediction within 10% of that calculated from the measured strains. The Jaguar was considered a far more complex aircraft than the Tucano (Case Study 1). The wing is fitted with spoilers, combat slats and a large range of external stores configurations are flown.

5.2 Training data

5.2.1 Data source

The data were sourced from the RAF Jaguar GR Mk1 OLM aircraft.

5.2.2 Strain mapping locations

The starboard inner wing (W5R) and the port outer wing (W3L), both sampled at 32S/s, were selected as the monitoring location (Fig. 9).

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

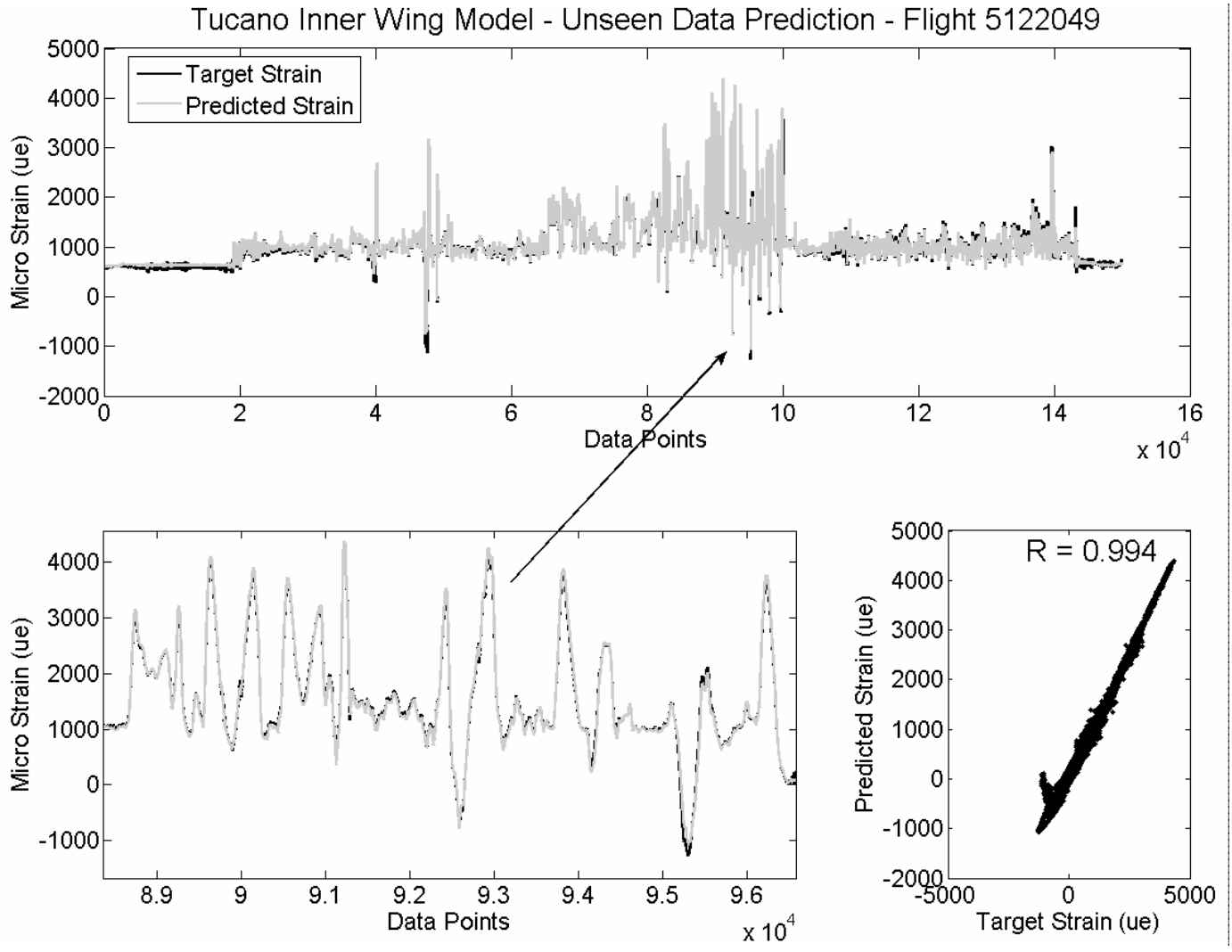


Figure 8. Performance of inner wing model (P3B) on validation flight 5122049.



Figure 9. Jaguar GR Mk1 wing strain mapping locations

5.2.3 Flight parameter data set

The flight parameters were sampled at 8 S/s, except normal acceleration port and starboard (16S/s) and tailplane position indicators which were sampled at 32 S/s.

5.2.4 Data set limitations

The data did not include outside air temperature and an international standard atmosphere was assumed. No through-flight fuel mass data were available; hence, fuel mass variation was estimated. Additionally, there were no pitch rate or pitch acceleration data. The use of simple semi-automatic data anomaly detection algorithms and graphical visualisation of the data identified significantly more anomalies in the raw Jaguar OLM data than had been seen in the Tucano data.

5.2.5 Application of domain knowledge

Domain knowledge was applied to the flight parameters and a 21-parameter input data set was constructed (Table 4).

5.2.6 Training data set

Models were trained on data from six severe flights, of differing profiles and stores configurations. Feature extraction methods were used to select highly damaging segments from within these flights and 100,000 line training and test data sets were concatenated.

5.2.7 Data pre-processing

The training data were normalised to zero mean and unit variance. The dimensionality of the input data was reduced from 21 to 10 principal components, using PCA (2% variance cut off).

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

Table 4
Jaguar parameter set – domain knowledge applied

Serial	Identification	Combined Parameters
1	ALT	Altitude
2	ALPH	Angle of attack
3	COM_MOVE*DYP	Combat slat moving (discrete) x dynamic pressure
4	COMBAT*DYP	Combat slat position x dynamic pressure
5	DYP	Dynamic pressure
6	FLAP*DYP	Flap position x dynamic pressure
7	MACH	Mach number
8	MACH^2	Mach number squared
9	NY*W	Lateral acceleration at cg x mass
10	NZ*W	Normal acceleration at cg x mass
11	NZ*W*MACH	Normal acceleration at cg x mass x Mach number
12	NZ*W*MACH^2	Normal acceleration at cg x mass x Mach^2
13	PAIL*DYP	Port aileron position x dynamic pressure
14	RACC*W	Roll acceleration x mass
15	RRAT*DYP/TAS	Roll rate x dynamic pressure / true airspeed
16	RUDD*DYP	Rudder position x dynamic pressure
17	SPOILP*DYP	Port spoiler position x dynamic pressure
18	SPOILS*DYP	Stbd spoiler position x dynamic pressure
19	TPLP*DYP	Port tailplane position x dynamic pressure
20	TPLR*DYP	Stbd tailplane position x dynamic pressure
21	WOW	Port main landing gear weight-on-wheels (discrete)

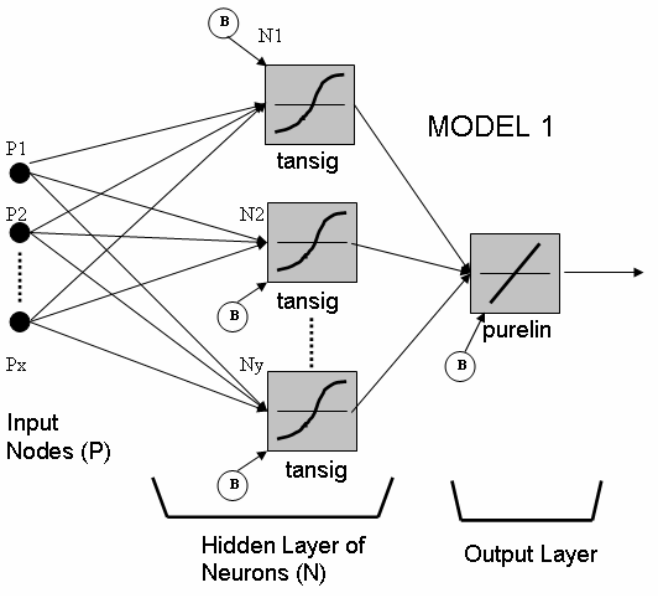


Figure 10. Jaguar Model 1 – SHAUNN.

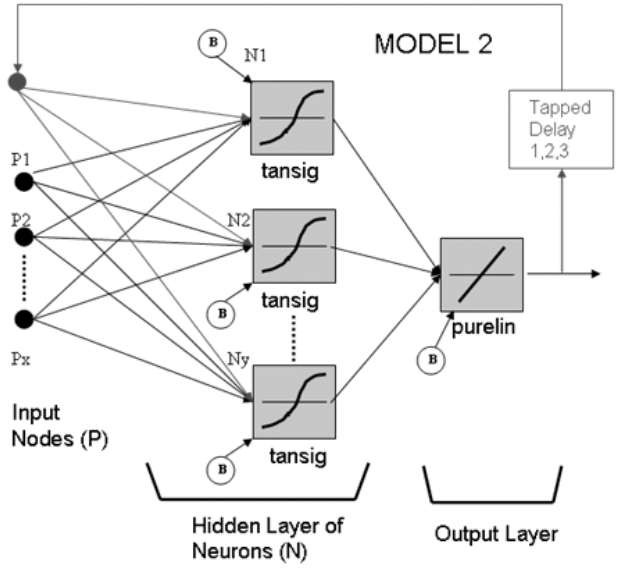


Figure 11. Jaguar Model 2 – Recurrent SHAUNN.

5.3 Model generation

The models were created using the Mathworks Matlab® Neural Network Toolbox[43]. Several model configurations were experimented with, generating around 40 models. Of these, two models were used to predict the validation data sets for each location. Both models had single hidden layer of 25 tansig neurons and a purelin output neuron (Figs 10 and 11) and were trained using the quasi-Newtonian Levenberg-Marquardt[20] algorithm. Model 2 (Fig. 11)

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

Table 5
Jaguar wing models – validation performance

	Inner Wing (W5R)							
	Model 1 - Non Recurrent				Model 2 - Recurrent			
Flight	Stress Correlation	Damage from Predicted Stress (x1E6)	Damage from Measured Stress (x1E6)	Fatigue Damage Ratio	Stress Correlation	Damage from Predicted Stress (x1E6)	Damage from Measured Stress (x1E6)	Fatigue Damage Ratio
048	0.992	164	134	1.22	0.992	162	134	1.21
097	0.974	311	302	1.03	0.974	311	302	1.03
189	0.992	95	93	1.02	0.988	110	93	1.18
383	0.981	330	314	1.05	0.981	330	314	1.05
448	0.990	180	144	1.25	0.992	190	144	1.32
	Average	Total	Total	Total	Average	Total	Total	Total
	0.986	1080	987	1.09	0.985	1103	987	1.12

	Outer Wing (W3L)							
	Model 1 - Non Recurrent				Model 2 - Recurrent			
Flight	Stress Correlation	Damage from Predicted Stress (x1E6)	Damage from Measured Stress (x1E6)	Fatigue Damage Ratio	Stress Correlation	Damage from Predicted Stress (x1E6)	Damage from Measured Stress (x1E6)	Fatigue Damage Ratio
048	0.996	47	43	1.09	0.995	55	43	1.28
097	0.985	27	46	0.59	0.987	25	46	0.54
189	0.996	19	21	0.90	0.995	19	21	0.90
383	0.995	54	78	0.69	0.993	58	78	0.74
448	0.996	57	47	1.21	0.992	62	47	1.32
	Average	Total	Total	Total	Average	Total	Total	Total
	0.994	204	235	0.87	0.992	219	235	0.93

was a recurrent network and contained a tapped delay[41], whereby the input data were augmented by the last three output predictions, to investigate whether time-history information was significant. Training was conducted using an error-activation method to reduce the risk of over-fitting and the training was stopped if the error in the test set increased five times, while the error in the training set continued to reduce.

5.4 Results

Performance of the inner wing and outer wing models, in terms of stress correlation coefficient and fatigue micro damage, using a five-flight validation data set is produced in Table 5.

An example of a validation run for the inner wing model 1, for flight 048, is reproduced in Fig. 12.

The average correlation between predicted and measured stresses, using the limited validation data, for the inner wing models was 0·986 and 0·985 respectively and for the outer wing models was 0·994 and 0·992 respectively. Although these values were outside the target stress correlations of 0·995, the performance in fatigue damage terms was better than would be expected from such stress correlations. The inner wing models over predicted cumulative fatigue damage by 9% and 12% respectively and the outer wing models under predicted damage by 13% and 7% respectively, marginally outside the target metrics. There was no appreciable difference between the performance of the non-recurrent and recurrent models.

Investigation of these errors revealed that the models had captured the character of the stress time histories well (Fig. 12) but offsets were apparent between predicted and measured values. This was considered most likely an effect of over simplification of the mass distribution of the different stores configuration. Also, the variation in mass through the sortie had to be estimated. It was considered that larger training data sets, stores configuration particular models and improvements to the mass assumptions, would lead to improvements in the model predictions.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

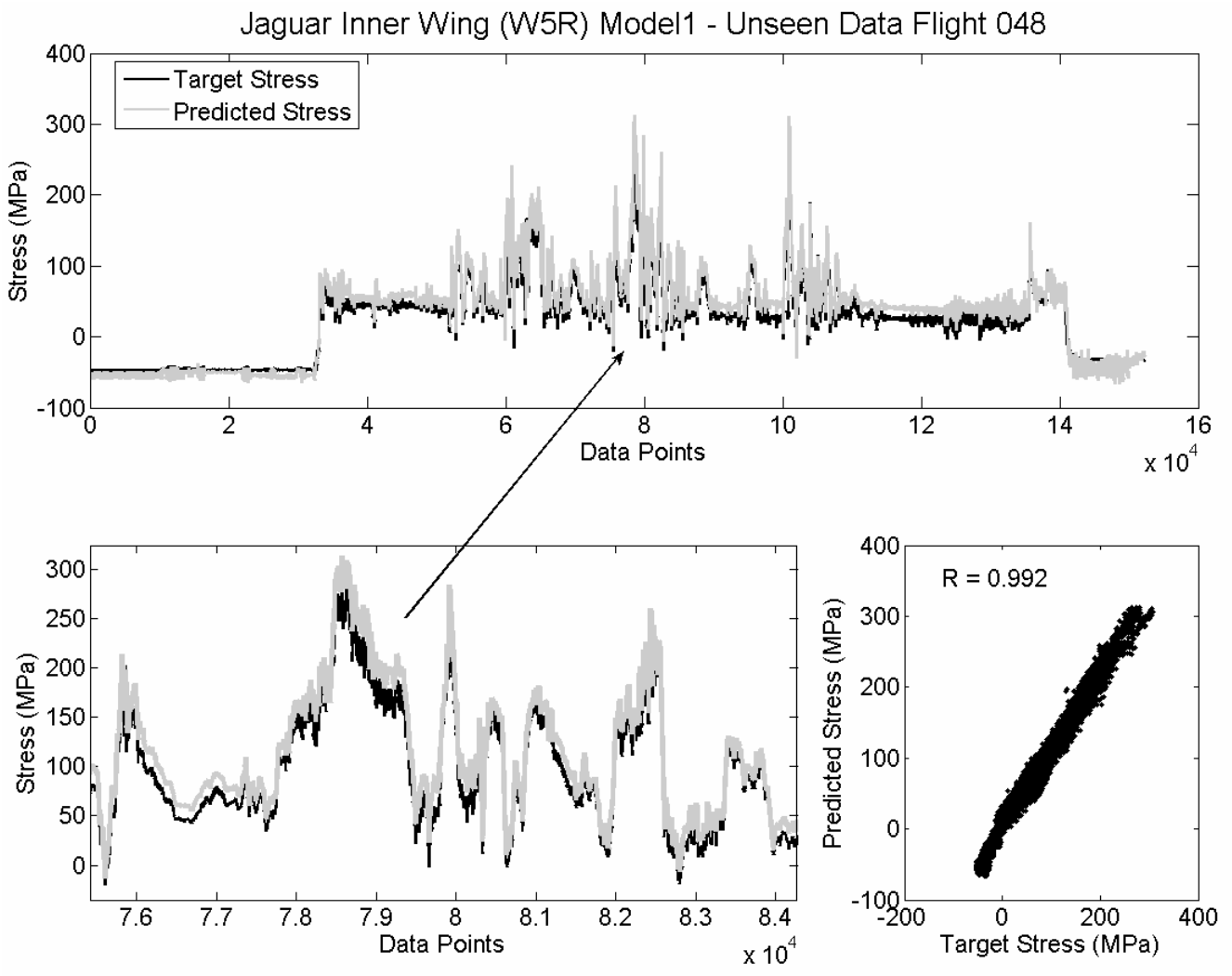


Figure 12. Performance of inner wing model 1 on validation flight 048.

6.0 CASE STUDY 3 – MILITARY TRAINER TAILPLANE MONITOR FEASIBILITY STUDY

6.1 Case study aim

The aim was to predict bending strains for the tailplane of the Tucano TMk1 military trainer. The target validation performance was an average correlation coefficient between predicted and measured strains of 0·995 and cumulative fatigue damage prediction within 10% of that calculated from measured strains. Predicting tailplane strains was considered significantly more complex than predicting wing strains and consequently the target performance criteria were considered challenging.

6.2 Training data

6.2.1 Data source

The data were sourced from the RAF Tucano TMk1 OLM aircraft ZF406.

6.2.2 Strain mapping locations

The port side tailplane bending bridge (PTB) (sampled at 128S/s) was selected as the monitoring location.

6.2.3 Flight parameter data set

The flight parameter data set available was the same as in Case Study 1 (Table 1).

6.2.4 Data set limitations

All the data limitations in Case Study 1 applied. In addition, analysis of the frequency content in the PTB revealed frequencies up to 30Hz.

6.2.5 SHAUNN justification

Justification for SHAUNN model development was undertaken by correlating the input parameters, with domain knowledge applied, against PTB (Fig. 13). Unlike wing monitoring locations, no significant correlations were identified. Regions of correlation on control surface plots were only apparent when the surfaces were deployed.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

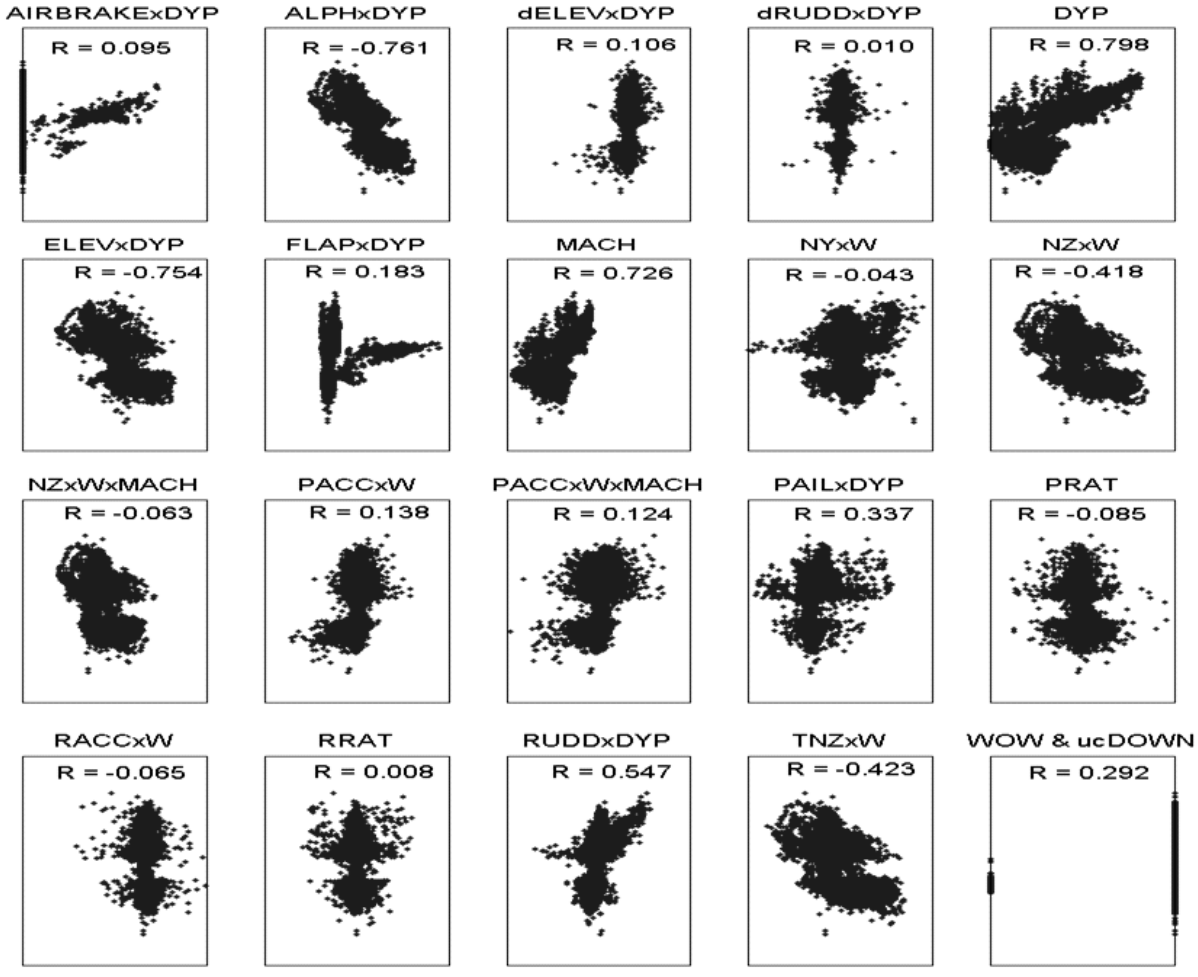


Figure 13. Tucano input parameters against PTB (target).

6.2.6 Application of domain knowledge

Domain knowledge was applied as in Case Study 1, with the addition of conditioned tail normal acceleration, rudder and tailplane control surface deflections and rates of deflection terms.

6.2.7 Training data set

It is not possible to predict directly events at frequencies up to around 30Hz from data sampled at 16S/s; therefore the training data was divided into its low frequency (manoeuvre) and high frequency (buffet) content. The manoeuvre data were predicted directly; whereas for buffet the fatigue damage rate was predicted. The fatigue damage rate was the fatigue damage calculated for each 1 second time slice. The manoeuvre content was separated out using an 8Hz low-pass filter. Data from seven flights were selected and concatenated into a 278,000-line training set and corresponding test set, using high strain range and low buffet damage content criteria.

For the buffet data, the progressive damage contribution from the high-frequency content was calculated from the difference between the damage for the full signal (128S/s) and the manoeuvre content. Seven flights were concatenated into a 330,000 line training set, with corresponding test set, using buffet damage rate criteria. Thereafter, the buffet damage rate for one second bursts was calculated and set as the target value. This process is illustrated in Fig. 14.

6.2.8 Data pre-processing

The training data were normalised to zero mean and unit variance. The dimensionality of the input data was reduced from 21 to 14 principal components using PCA.

6.3 Model generation

6.3.1 Training algorithm selection

In order to identify the most appropriate training algorithm, nine algorithms, selected from the Mathworks Matlab® Neural Network Toolbox[43], were trained using the Case Study 1 data set. Ten runs of each algorithm were undertaken; 15 hidden units in a single hidden layer were used and a target mean square error (MSE) of 0·003 was set. The training ran for a maximum of 10,000 epochs (passes of the data). The results are reproduced in Table 6.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

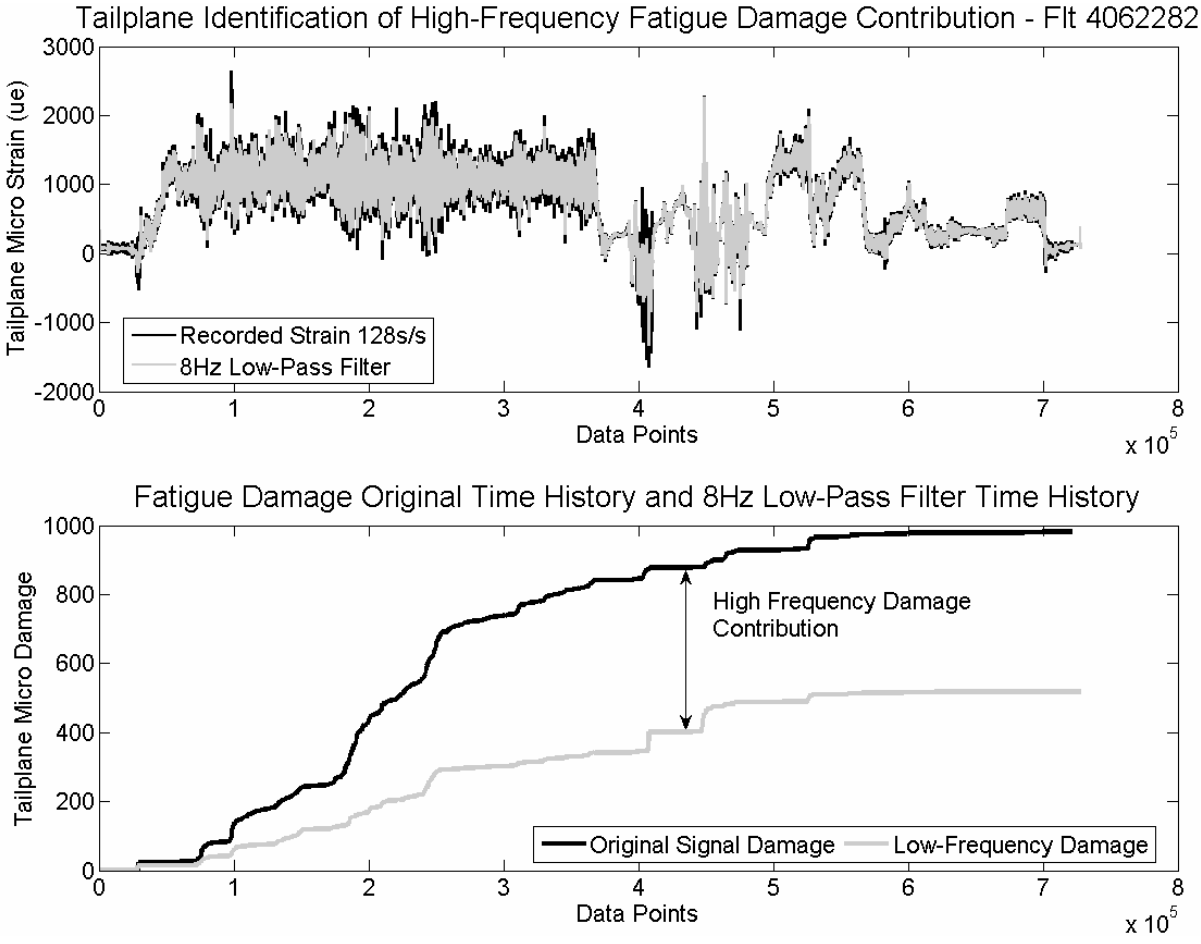


Figure 14. Extraction of high-frequency training set.

Note: Epochs = 10,000 denotes models that failed to reach a mean squared error of 0·003 before 10,000 Epoch limit.

The resilient back propagation (RPROP) algorithm (Algorithm 5) was the quickest algorithm to converge. The more complex quasi Newtonian Levenberg-Marquardt and the Bayesian algorithms converged in fewer epochs but the size of training set and likely complexity[43] of the models required for this problem suggested against the use of these algorithms for a feasibility study. Additionally, Riedmiller and Braun's excellent paper on RPROP's development[24] provided sufficient information to code this algorithm and hence it was used for this study.

6.3.2 Manoeuvre model architecture

Simple models, with one hidden layer and around 25 neurons, were unable to predict accurately the manoeuvre strains. Therefore models with two hidden layers, each of 25 tansig neurons, were used (Fig. 2). Single optimised models, rather than committees were used for this study. The best performing model (of 25) was designated as model manoeuvre and was trained for 5,000 epochs, using the coded RPROP algorithm. The gradient of the error function was almost zero at the end of training but the risk of over-fitting was low, due to the large size of the training data set, relative to the number of weights in the model.

6.3.3 Buffet model architecture

The buffet modelling problem was even more complex. The best performing buffet model (of 60), was identified as model buffet; it had two hidden layers of 35 tansig neurons and was trained for 1,000 epochs, using the RPROP algorithm. The gradient of the error function was almost zero at the end of training.

6.4 Results

6.4.1 Manoeuvre model

The average correlation between measured and predicted manoeuvre strains for the 38 flight validation data set was 0·991, compared with a correlation of 0·993 for the test set; this was outside the target of 0·995. However, the cumulative fatigue damage for the validation set was 4% below that calculated for the recorded manoeuvre strains, within the 10% aim. An example validation plot for a Model Manoeuvre prediction is presented in Fig. 15.

The flight-by-flight damage prediction, the cumulative Model Manoeuvre predicted damage, the damage from measured strains and the fatigue damage calculated for a ±5% variation in the

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

Table 6
Training algorithm performance

	Gradient Descent Backpropagation (1)			Backpropagation with Momentum (2)			Backpropagation and Variable Learning Rate		
Run	R	Epochs	Time(s)	R	Epochs	Time(s)	R	Epochs	Time(s)
1	0.998	10000	2591	0.998	10000	2508	0.999	1251	304
2	0.998	10000	2598	0.998	8004	2006	0.999	2314	570
3	0.998	4581	1184	0.998	10000	2486	0.999	1623	397
4	0.998	8140	2093	0.998	9343	2310	0.999	1266	307
5	0.998	5933	1529	0.998	10000	2491	0.999	1305	319
6	0.998	10000	2568	0.998	10000	2491	0.999	1485	360
7	0.998	5833	1504	0.998	7331	1833	0.999	1471	359
8	0.998	10000	2589	0.998	10000	2503	0.999	1018	249
9	0.998	10000	2551	0.998	8400	2107	0.999	910	220
10	0.998	9609	2477	0.998	10000	2513	0.999	1144	277
Ave	0.998	8410	2168	0.998	9308	2325	0.999	1379	336

	Backpropagation, Variable Learning Rate and Momentum (4)			Resiliant Backpropagation (5)			Conjugate Gradient Algorithm (6)		
Run	R	Epochs	Time(s)	R	Epochs	Time(s)	R	Epochs	Time(s)
1	0.999	311	75	0.999	76	20	0.999	62	38
2	0.998	216	52	0.999	80	21	0.999	69	39
3	0.998	557	134	0.999	75	20	0.999	51	32
4	0.998	388	92	0.999	56	15	0.999	54	34
5	0.998	545	131	0.999	57	15	0.999	45	29
6	0.998	451	108	0.999	64	17	0.999	45	29
7	0.999	334	81	0.999	64	17	0.999	60	38
8	0.998	442	106	0.999	64	17	0.999	48	31
9	0.999	1140	269	0.999	58	15	0.999	45	29
10	0.999	408	97	0.999	86	22	0.999	58	37
Ave	0.998	479	115	0.999	68	18	0.999	54	34

	Scaled Conjugate Gradient (7)			Levenberg-Marquardt Quasi-Newtonian Algorithm (8)			Bayesian Regularisation (with Levenberg Marquardt) (9)		
Run	R	Epochs	Time(s)	R	Epochs	Time(s)	R	Epochs	Time(s)
1	0.999	65	35	0.999	8	23	0.996	8	24
2	0.999	55	30	0.999	6	19	0.996	6	19
3	0.999	50	28	0.999	11	30	0.994	9	26
4	0.999	48	27	0.999	5	18	0.995	10	28
5	0.999	65	35	0.999	4	16	0.994	7	21
6	0.999	50	28	0.999	6	19	0.995	7	21
7	0.999	61	33	0.999	12	31	0.993	7	21
8	0.999	65	35	0.999	12	31	0.992	9	26
9	0.999	53	29	0.999	7	21	0.992	8	24
10	0.999	48	27	0.999	10	29	0.995	6	19
Ave	0.999	56	31	0.999	8	24	0.994	8	23

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

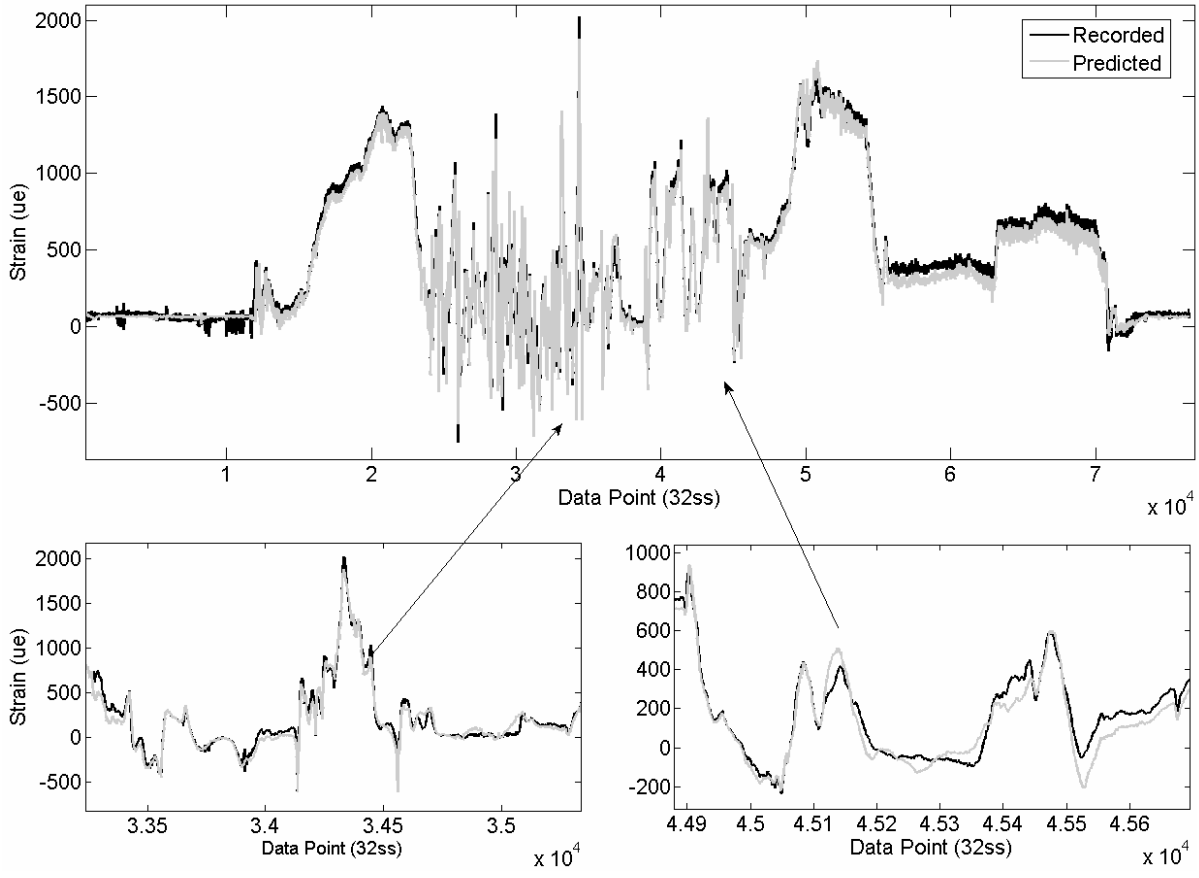


Figure 15. Example model manoeuvre validation flight 4061966 prediction.

measured strain, for the 38-flight validation set are presented in Table 6 and plotted in Fig. 16. The damage calculated for the ±5% variation in the measured strain was included to put the model prediction error in context of credible strain gauge and acquisition system errors.

6.4.2 Buffet model

The scatter in performance of the buffet model was greater than the manoeuvre model but this was expected due to the approximations in trying to predict an average damage rate for a time slice rather than a strain. Model buffet overestimated the cumulative damage for the 38 flight validation set by 11%. Table 7 and Fig. 17 show the flight-by-flight damage prediction, the cumulative model buffet predicted damage, the damage attributed to buffet from the measured strains and the damage calculated for a strain variation of ±5%, for the 38-flight validation set.

6.4.3 Combined model

The combination of the best performing models (Manoeuvre + Buffet) over-predicted cumulative damage for the 38 flight validation set by 4%, within the 10% target value. Combined results are presented in Table 7 and plotted in Fig. 18.

7.0 CONCLUSIONS

SHAUNN have the potential to produce sufficiently accurate, relatively cost-effective structural usage monitoring systems, across the range of aircraft types and roles.

Although certification of a SHAUNN-based monitor would be more complex than a 'traditional' monitor, the development framework outlined here could be used as the basis of a route to certification[38].

Further development of the methods outlined in the Case Studies should be undertaken, to improve accuracy and to validate further the methods using larger data sets.

ACKNOWLEDGEMENT

The author wishes to acknowledge QinetiQ Farnborough and the UK Ministry of Defence for giving permission to publish this paper. In addition, although it should be noted that the opinions presented within this paper are those of the author and do not necessarily represent the policy or practice of any organisation or group, these opinions have been influenced by discussions with a great many learned colleagues throughout the UK military aviation structures community and the Military Aircraft Structural Airworthiness Advisory Group in particular.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

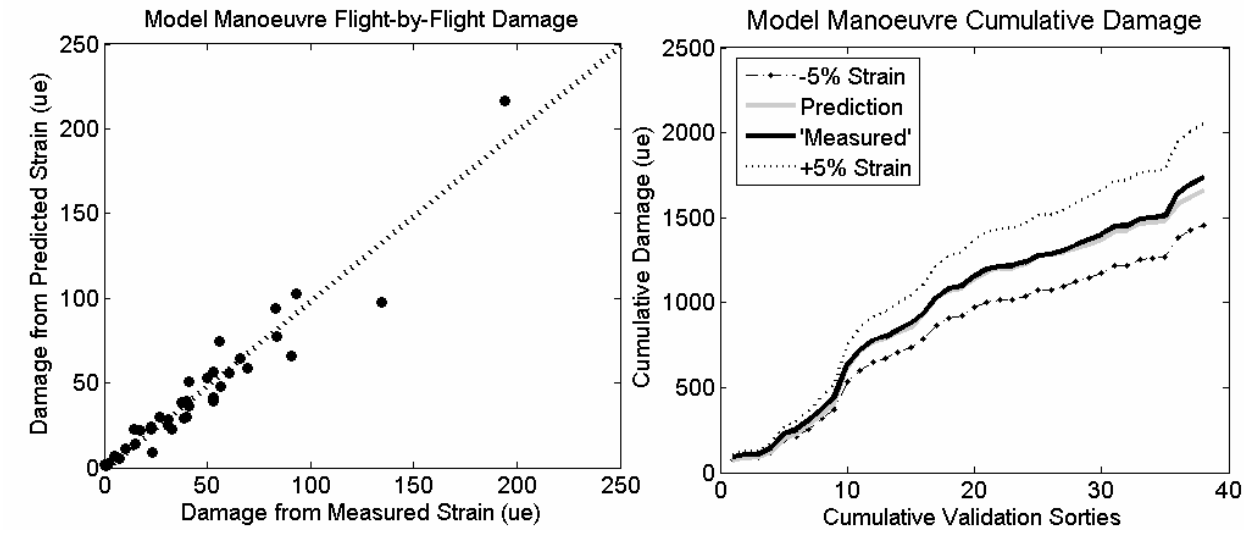


Figure 16. Model manoeuvre flight-by-flight and cumulative fatigue damage from validation data.

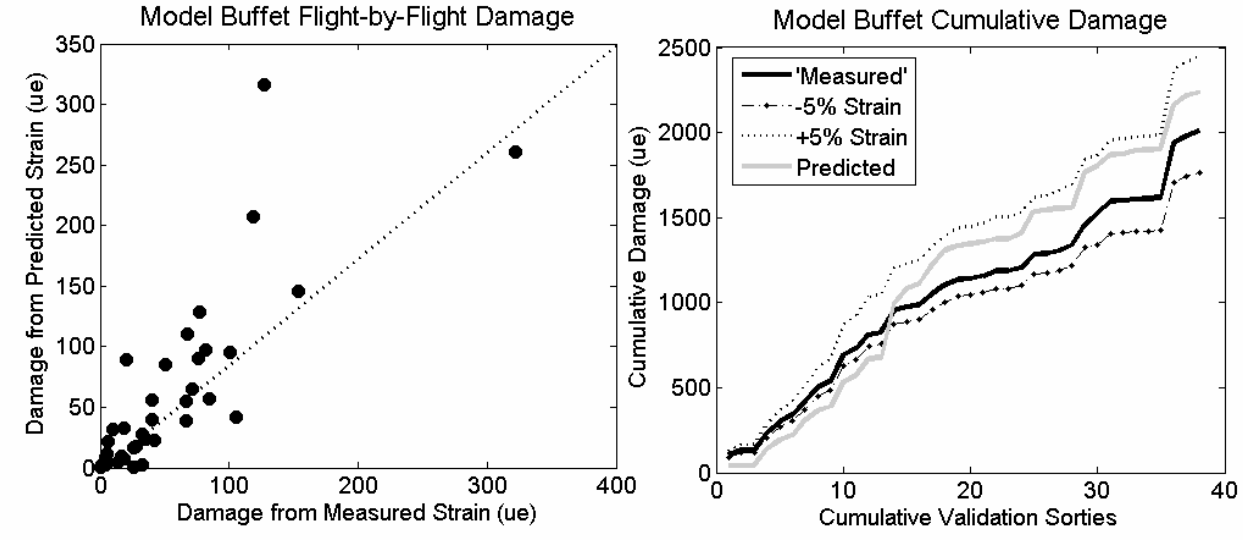


Figure 17. Model buffet flight-by-flight and cumulative fatigue damage from validation data.

The author wishes to acknowledge Wg Cdr R.K. Eckersley, formerly of the UK Ministry of Defence, Aircraft Structural Integrity branch, for having the foresight to fund some of the early work presented here, which has led to the development of a Defence Standard for the use of non-adaptive predictive methods in aircraft fatigue monitoring systems[38]. The guidance and support provided by Professor W.J. Staszewski and Dr. J.E. Moon throughout this work is also acknowledged. The author is also pleased to recognise the painstaking efforts of his colleague, Mr A.J. Mountfort who has steadfastly undertaken the technical review of the plethora of reports produced during this work; finally the permission given by Sqn Ldr K.E. Storer to use the aircraft photographs reproduced in this paper is acknowledged.

REFERENCES

1. GELDER, M., ROBERTS, S., HOLFORD, D. and REED, S. Fatigue and operational loads monitoring of the Red Arrows, May 2000, Proceedings of the 20th Symposium on Aircraft Integrated Monitoring Systems, Garmisch-Partenkirchen, Germany.
2. HOLFORD, D.M. and STURGEON, J.R. Operational load measurement: a philosophy and its implementation, 1984, NATO Advisory Group for Aerospace Research and Development, AGARD Conference Proceedings No 375, Sienna, Italy.
3. WARD, A.P. Experience obtained from service fatigue monitoring exercises, 1984, NATO Advisory Group for Aerospace Research and Development, AGARD Conference Proceedings No 375, Sienna, Italy.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

Table 7
Model performance for 38 flight validation set

Flight Number	Predicted Damage Manoeuvre (x1E6)	Manoeuvre Damage from Measured Strain (x1E6)	Predicted / Damage from Manoeuvre Strains	Predicted Damage Buffet (x1E6)	Buffet Damage from Measured Strains (x1E6)	Predicted / Damage from Buffet Strains	Total Predicted Damage (x1E6)	Total Damage from Measured Strains (x1E6)	Total Predicted / Total Damage from Measured Strains
4061768	66.17	90.58	0.73	41.58	105.55	0.39	107.75	196.13	0.55
4061769	22.42	14.42	1.55	0.61	25.81	0.02	23.03	40.23	0.57
4061834	2.04	0.51	4.00	0.70	0.71	0.98	2.74	1.22	2.24
4061963	29.41	38.79	0.76	95.19	101.06	0.94	124.60	139.85	0.89
4061966	77.75	83.48	0.93	55.25	67.00	0.82	133.00	150.48	0.88
4061994	29.65	26.61	1.11	22.96	42.06	0.55	52.61	68.67	0.77
4062061	56.51	52.81	1.07	90.77	75.78	1.20	147.28	128.59	1.15
4062073	58.37	69.36	0.84	56.90	84.53	0.67	115.27	153.89	0.75
4062085	64.38	65.93	0.98	24.16	35.76	0.68	88.54	101.69	0.87
4062105	216.16	193.95	1.11	145.39	154.13	0.94	361.55	348.08	1.04
4062106	94.06	82.79	1.14	39.71	39.94	0.99	133.77	122.73	1.09
4062155	47.77	56.68	0.84	97.76	81.46	1.20	145.53	138.14	1.05
4062162	22.35	22.96	0.97	4.43	13.94	0.32	26.78	36.90	0.73
4062175	36.52	40.86	0.89	316.59	127.28	2.49	353.11	168.14	2.10
4062188	29.60	39.63	0.75	88.97	20.40	4.36	118.57	60.03	1.98
4062228	74.60	55.64	1.34	31.40	9.92	3.17	106.00	65.56	1.62
4062237	102.96	93.21	1.10	110.91	67.56	1.64	213.87	160.77	1.33
4062249	41.16	53.09	0.78	85.21	50.25	1.70	126.37	103.34	1.22
4062251	13.82	15.34	0.90	27.40	33.08	0.83	41.22	48.42	0.85
4062253	55.50	60.49	0.92	8.92	4.39	2.03	64.42	64.88	0.99
4062257	38.40	39.08	0.98	10.07	16.89	0.60	48.47	55.97	0.87
4062260	21.69	17.38	1.25	17.81	28.35	0.63	39.50	45.73	0.86
4062261	1.13	1.11	1.02	2.16	1.72	1.26	3.29	2.83	1.16
4062266	23.99	22.48	1.07	32.59	18.44	1.77	56.58	40.92	1.38
4062267	50.90	41.09	1.24	128.93	77.00	1.67	179.83	118.09	1.52
4062268	7.17	5.09	1.41	11.29	5.37	2.10	18.46	10.46	1.76
4062271	9.11	23.13	0.39	7.29	18.31	0.40	16.40	41.44	0.40
4062272	22.71	32.51	0.70	2.90	32.55	0.09	25.61	65.06	0.39
4062276	24.94	30.87	0.81	206.97	118.87	1.74	231.91	149.74	1.55
4062281	28.41	31.03	0.92	38.36	66.83	0.57	66.77	97.86	0.68
4062287	52.73	49.82	1.06	65.16	71.63	0.91	117.89	121.45	0.97
4062291	2.70	2.18	1.24	5.42	4.88	1.11	8.12	7.06	1.15
4062293	39.38	39.67	0.99	21.81	6.03	3.62	61.19	45.70	1.34
4062294	5.64	7.43	0.76	2.73	3.98	0.69	8.37	11.41	0.73
4062295	11.50	10.50	1.10	4.09	6.42	0.64	15.59	16.92	0.92
4062296	97.77	134.40	0.73	260.24	322.03	0.81	358.01	456.43	0.78
4062297	39.56	52.82	0.75	56.42	40.14	1.41	95.98	92.96	1.03
4062298	38.75	37.66	1.03	16.14	26.17	0.62	54.89	63.83	0.86
TOTAL	**1658**	**1735**	**0.96**	**2235**	**2006**	**1.11**	**3893**	**3742**	**1.04**

4. REED, S., HOLFORD, D., GELDER, M. and ROBERTS, S. Operational loads measurement of the Hawk aircraft in RAF service, June 2001, Proceedings of the 21st Symposium of the International Committee on Aeronautical Fatigue, Toulouse, France.
5. LINCOLN, J.W. Aging aircraft – USAF experience and actions, 15th Plantema Memorial Lecture June 1997, Proceedings of the 19th Symposium of the International Committee on Aeronautical Fatigue, Edinburgh.
6. YANISHEVSKY, M., DOUCHANT, A., BRETON, M., TURCOTTE, C. and SOVA, M. CT114 Tutor – Snowbird aerobatic aircraft safety by inspection coupon test program, June 2001, Proceedings of the 21st Symposium of the International Committee on Aeronautical Fatigue, Toulouse, France.
7. HUNT, S.R. and HEBDEN, I.G. Eurofighter 2000: An integrated approach to structural health and usage monitoring, 1998, AGARD RTO Proceedings 7, Exploitation of Structural Loads/Health Data for Reduced Life Cycle Costs, Brussels, Belgium.
8. AZZAM, H. A practical approach for the indirect prediction of structural fatigue from measured flight parameters, *J Aero Eng*, June 1997, **211**, (G1), pp 29-38.
9. AZZAM, H., HRBDEN, I., GILL, L., BEAVEN, F. and WALLACE, M. Fusion and decision making techniques for structural prognostic health management, February 2005, IEEE Aerospace Conference Paper No 1535.
10. WALLACE, M., AZZAM, H. and NEWMAN, S. Indirect approaches to individual aircraft structural monitoring, *J Aero Eng*, 2004, **218**, (G), pp 329-346.
11. JACOBS, J. H. and PEREZ, P. A Combined approach to buffet response analysis and fatigue life prediction, *An Assessment of Fatigue Damage and Crack Growth Prediction Techniques*, 1993, NATO AGARD Report 79, Papers presented at the 77th Meeting of the AGARD Structures and Materials Panel, 29-30 September 1993, Bordeaux, France.
12. KIM, D. and MARCINIAK, M. A methodology to predict the empennage in-flight loads of a general aviation aircraft using backpropagation neural networks, February 2001, DOT/FAA/AR-00/50, Washington, DC, USA.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press

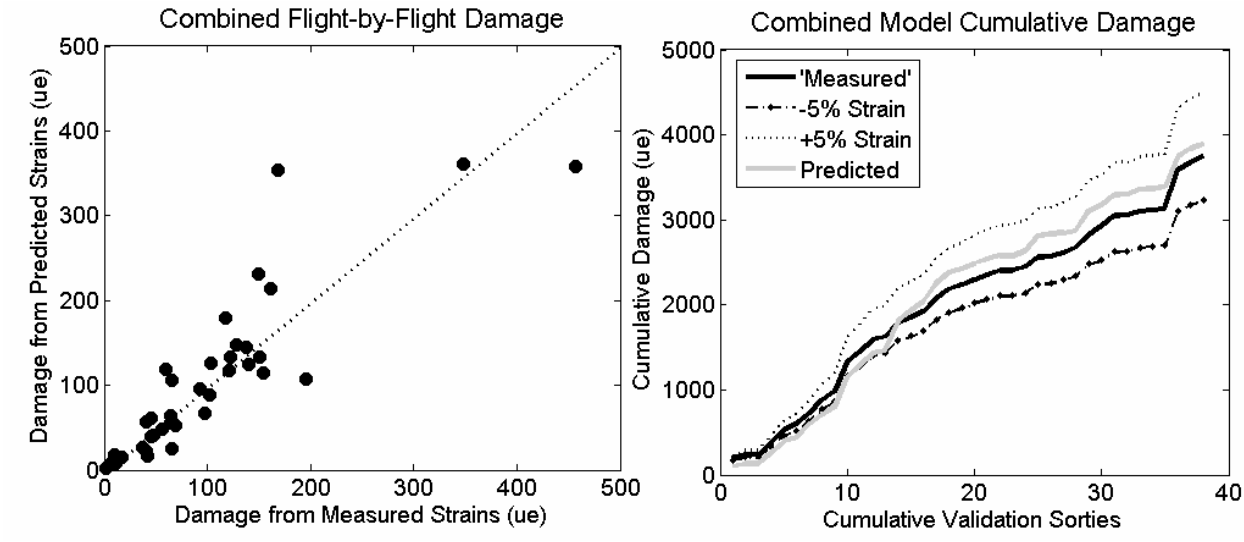


Figure 18. Combined model (manoeuvre + buffet) damage accrual for 38 flight validation set.

13. HILL, K., HUDSON, R.A., IRVING, P.E. and VELLA, A.D. Loading spectra, usage monitoring and prediction of fatigue damage in helicopters, May 1995, Proceedings of the 18th ICAF Symposium on Aeronautical Fatigue, Melbourne, Australia.
14. MANRY, M.T., HSIEH, C.H. and CHANDRASEKARAN. H. Near-optimal flight load synthesis using neural networks, August 1999, Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, Madison, Wisconsin, USA.
15. LEVINSKI. O. Prediction of buffet loads using artificial neural networks, Document DSTO-RR-0218, September 2001, Australian Defence Science and Technology Organisation.
16. WANG, J. Service Loads Prediction and Recorder Data Validation Using Artificial Neural Networks, July 2003, PhD dissertation, Institute of Aeronautics and Astronautics, National Cheng Kung University, Taiwan.
17. REED, S. and COLE, D. Development of a parametric aircraft fatigue monitoring system using artificial neural networks, May 2003, Proceedings of the 22nd Symposium of the International Committee on Aeronautical Fatigue, Lucerne, Switzerland.
18. REED, S. A parametric-based empennage fatigue monitoring system using artificial neural networks, June 2005, Proceedings of the 23rd Symposium of the International Committee on Aeronautical Fatigue, Hamburg, Germany.
19. GURNEY, K. *An Introduction to Neural Networks*, 1997, Routledge, London.
20. HAYKIN, S. *Neural Networks: A Comprehensive Foundation*, 1999, Second Edition, Prentice Hall, NJ, USA.
21. FAUSETT, L.V. *Fundamentals of Neural Networks, Architectures, Algorithms and Applications*, 1994, Prentice-Hall, NJ, USA.
22. HAGAN, M.T., DEMUTH, H.B. and BEALE, M.H. *Neural Network Design*, 1996, Brooks Cole.
23. BISHOP, C.M. *Neural Networks for Pattern Recognition*, 1995, Oxford University Press.
24. RIEDMILLER, M. and BRAUN, H. A direct adaptive method for faster backpropagation learning: The RPROP Algorithm, 1993, Proceedings of the IEEE International Conference on Neural Networks.
25. MACKAY, D.J.C. Bayesian interpolation, *Neural Computing*, 1992, **4**, (3), pp 416-447.
26. Occam's Razor (also spelled Ockham's Razor), is a principle attributed to the 14th-century English logician and Franciscan friar, William of Ockham. It forms the basis of methodological reductionism, also called the principle of parsimony or law of economy. In its simplest form, Occam's Razor states that one should make no more assumptions than needed, or given two equally predictive theories, choose the simpler.
27. CARON, Y. and RICHARD, Y. CF-188 Fatigue life management programme, May 1998, 1998 Specialists' Meeting on Exploitation of Structural Loads/Health Data for Reduced Life Cycle Costs, NATO Research and Technology Organisation Proceedings 7, Brussels, Belgium.
28. 'No-go' is a term used in UK military aviation to identify an item of equipment for which the aircraft cannot be launched in peacetime if this equipment is unserviceable.
29. CRONKITE, J.D. and GILL, L. Technical evaluation report on 1998 Specialists' Meeting on Exploitation of Structural Loads/Health Data for Reduced Life Cycle Costs, May 1998, NATO Research and Technology Organisation Proceedings 7, Brussels, Belgium.
30. TSOUKALIS, L.H, and UHRIG, R.E. *Fuzzy and Neural Approaches in Engineering*, 1996, John Wiley and Sons, pp 238.
31. SWINGLER, K. *Applying Neural Networks – A Practical Guide*, 1996, Morgan Kaufman Publishers, San Francisco, CA, USA.
32. MACKAY, D. J. C., *Information Theory, Inference and Learning Algorithms*, 2003, pp 527-534, Cambridge University Press.
33. NABNEY, I.T. *Netlab: Algorithms for Pattern Recognition – Advances in Pattern Recognition*, 2002, Springer Publishers.
34. WORDEN, K. Structural fault detection using a novelty measure, *J Sound and Vibration*, 1997, **201**, (1), pp 85-101.
35. MANLY, B.F.J. *Multivariate Statistical Methods – A Primer*, 1994, Second Edition, Chapman and Hall CRC Press, FL, USA.
36. STASZEWSKI, W.J., BOLLER, C., GRONDELL, S., BIEMANS, C., O'BRIEN, E., DELEBARRE, C. and TOMLINSON, G.R. *Damage Detection Using Stress and Ultrasonic Waves: in Health Monitoring of Aerospace Structures – Smart Sensor Technologies and Signal Processing*, 2004, pp 179-181, STASZEWSKI, W. J., BOLLER, C. and TOMLINSON, G.R. (Eds), John Wiley and Sons.
37. *Military Airworthiness Regulations*, July 2003, First Edition, UK Joint Service Publication, JSP553.
38. REED, S.C. Proposed Defence Standard 00-970 – structural monitoring systems using non-adaptive prediction methods, July 2006, UK Military Aircraft Structural Airworthiness Advisory Group, MASAAG Paper 107A.
39. Design and airworthiness requirements for service aircraft, 2006, Issue 4, UK MoD Defence Standard 00-970.
40. Neuromat, Neural Network Software, June 2001, Version 1.0.
41. Standard practices for cycle counting in fatigue analysis, 1985, American Society for Testing and Materials (ASTM), E1049-85, (Re-approved 1997).
42. Fatigue of aluminium alloy joints with various fastener systems – low load transfer, 1989, Engineering Sciences Data Unit (ESDU), Item 89046.
43. DEMUTH, H. and BEALE, M. *Neural Network Toolbox – Mathworks Matlab User's Guide*, 2001, Seventh edition.

https://doi.org/10.1017/S0001924000004474 Published online by Cambridge University Press