Prediction of Landing Gear Loads Using Machine Learning Techniques

E. CROSS*, P. SARTOR†, K. WORDEN* and P. SOUTHERN†

ABSTRACT

This work aims to establish if significant correlations exist between flight parameters recorded on production aircraft and the loads induced in the landing gear by employing accurate nonlinear regression models developed using machine learning techniques. The mathematical modelling approach used in the development of the regression model employs both classical Multi-Layer Perceptron (MLP) and Bayesian MLP neural networks. The MLP neural networks in this work were developed using landing gear drop test data. The inputs from the drop test data include shock absorber travel, tyre closure, shock absorber pressure, wheel speed, drop carriage accelerations, landing gear accelerations, while the initial output target to be predicted is the landing gear side stay load. To demonstrate the fidelity of the model and avoid issues with overfitting to the data, the landing gear drop test data was divided into training, validation and test data sets, which did not overlap. The performance of the neural network is defined by the Mean-Square Error (MSE) between model predictions and the measured targets. In the preliminary model development, the MSE for the classic MLP implementation was 8.53% for the testing set, which is a very encouraging result. The Bayesian MLP was also found to perform well. In conclusion, the neural network developed at this preliminary stage has performed well for the prediction of the side stay load in the drop test data.

Elizabeth Cross, University of Sheffield, Department of Mechanical Engineering, Mappin Street, Sheffield, S1 3JD, Email: e.j.cross@sheffield.ac.uk
Pia Sartor, Messier-Bugatti-Dowty, Messier-Dowty Ltd., Cheltenham Road East, Gloucester, GL2 9QH, Email: pia.sartor@safranmbd.com

INTRODUCTION

There are several areas of interest in aircraft landing gear structural health monitoring: understanding the current operating environment of landing gear in order to allow for an improvement in the evaluation of fatigue design criteria, surveillance of the landing gear fleet in order to detect overload occurrences (and equally to indicate which occurrences were not overloads), and ultimately to allow the certification of the landing gear to be based on the actual life experienced in service.

Different approaches can be taken when determining in-service landing gear loads, such as the use of kinematics (accelerations, velocities and displacements) or the use of force measurements (pressure or strain). This work aims to establish if correlations exist between flight parameters recorded on production aircraft (such as accelerations, velocities, displacements) and the loads induced in the landing gear. The approach taken employs accurate nonlinear regression models developed using machine learning techniques.

One of the key benefits of using machine learning techniques is that minimal additional aircraft instrumentation is required to establish the loads on the landing gear. Ideally all of the required information will be available from the aircraft systems. An additional benefit of this approach is that it could easily be expanded to the aircraft maintenance monitoring system. This leads to a method of in-service loads monitoring that is advantageous in terms of weight, system complexity and reliability.

This paper describes the machine learning technique that is used to predict landing gear loads from drop test data. The mathematical modelling approach that is used in the development of the regression models is first discussed. In this work, regression models are developed with a Multi-Layer Perceptron (MLP) neural network and a Bayesian MLP. The results of applying the preliminary models to the drop test data are then explained and finally, future plans for model development and testing are discussed.

MACHINE LEARNING

The term machine learning, in the context of this work, applies to the gaining of knowledge about the relationships between recorded flight parameters and landing gear loads *from data* (in this case collected from a drop test). Specifically in this paper, neural networks are trained to attempt to predict the landing gear side stay load, as a test case, from other measured parameters.

Neural networks are a biologically inspired way of creating complex and nonlinear functional forms for modelling the relationship between variables in a data set. The use of neural networks in the machine learning community is common and the theory behind them well established (see for example [1, 2]) hence, few details will be given here.

The architecture of a simple neural network is illustrated in Figure 1, where the network is comprised of units which are arranged in an input layer, hidden layers and an output layer. Inputs to the network, which are each individually weighted, w, are fed into units along with a bias, b, which then feed into other units arranged in layers. The bias for each unit in a network can be thought of as an extra weight for each unit and will be referred to as such in the remainder of this paper.

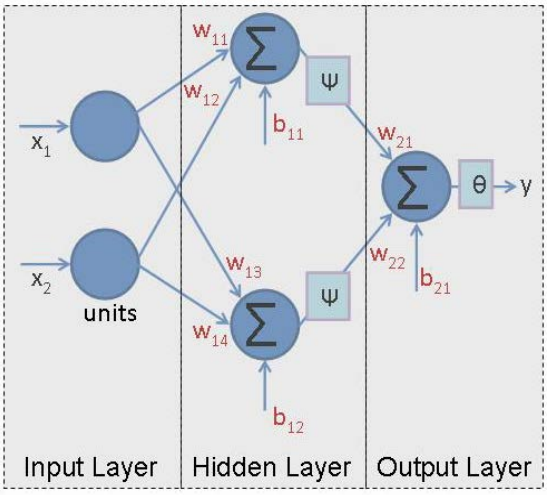

Figure 1. A typical two-layer network architecture with network weights and biases.

There are a number of common network architectures and in this work the regression model developed uses a Multi-Layer Perceptron (MLP), which has been shown to be a universal approximator [1].

For a fixed network architecture, network optimisation is achieved through the minimisation of the error function, which is dependent on the network weights. A number of algorithms are commonly used for the optimisation task with the simplest variants being versions of gradient descent; the current state-of-the-art is centred around second-order methods that require evaluation of a Hessian, and the predominant algorithms are Levenburg-Marquardt and scaled conjugate gradients (see [1] for more details).

A significant issue when utilising neural networks, and indeed other machine learning techniques, is the risk of overfitting to the training data which can lead to poor generalisation capabilities of the network when used as a predictor for an unseen data set. A common approach to avoid overfitting issues is to utilise, as well as a data set used to train the network, non-overlapping validation and testing sets of data. These additional data sets can be used to measure the network's performance on unseen data. In this work, a number of networks of varying complexities are trained using the training data set. The network which performs best on the non-overlapping validation set is then selected for further trials. To assess network performance in this work, a normalised Mean Square-Error taking the following form is used:

$$MSE = \frac{100\sum_{i=1}^{n}(y - f(x))^2}{n\sigma_y^2} \qquad \textbf{(1)}$$

where y is the measured target, $f(x)$ is the model prediction, σ_y^2 is the variance of the target data and n is the number of points in the test set.

Despite the modelling capability and sophistication of neural networks, there is one aspect that remains unsatisfactory. Even if a trained neural network does generalise well, if it were trained on a different dataset, the weights would more than likely be different, if only by a small margin. This uncertainty about what the true weights should be is dealt with in Bayesian inference by assigning a probability

distribution for the weights. This means that, for a neural network, all likely weight values are considered. Through the use of Bayes' rule, these distributions over all possible weight values can be used to obtain a distribution over the predicted values of the network. The mean of that distribution can be used as the value of the prediction, and the standard deviation can be used to give confidence intervals on this prediction.

As with all Bayesian techniques, a Bayesian MLP requires the specification of a prior, which incorporates one's beliefs about the weights before any data has been seen. In this work a Gaussian prior distribution is adopted, which enables the size of the weights to be restricted and fits with the idea that the generating function of interest should be smooth. Following [3], for a set of weights ***w***, in a network, the Gaussian prior distribution has the form:

$$p(\mathbf{w}) = \frac{1}{Z_w(\alpha)} \exp\left[-\frac{\alpha}{2} \|\mathbf{w}\|^2 \right] \quad \textbf{(2)}$$

where $Z_w(\alpha) = \int \exp\left[-\frac{\alpha}{2} \|\mathbf{w}\|^2 \right] d\mathbf{w}$ is a normalisation constant and α is a hyperparameter corresponding to the inverse variance of weight values that needs to be chosen. Next, the likelihood must be specified. With an ideal model, the prediction errors will simply depend on the noise of the measurements, which are usually modelled as Gaussian. In this case, the likelihood has the form:

$$p(D|\mathbf{w}) = \frac{1}{Z_D(\beta)} \exp\left[-\frac{\beta}{2} \sum errors^2 \right] \quad \textbf{(3)}$$

where $Z_D(\beta) = \int \exp\left[-\frac{\beta}{2} \sum errors^2 \right] dD$ is another normalisation term, D refers to the training data set, and β is another hyperparameter which represents the inverse variance of the measurement noise on the outputs. If the likelihood and priors have the form of Equations 2 and 3, the posterior distribution of the weights can be expressed as:

$$p(w|D) = \frac{1}{Z_S} \exp\left[-\frac{\beta}{2} \sum errors^2 - \frac{\alpha}{2} \|\mathbf{w}\|^2 \right] \quad \textbf{(4)}$$

where $Z_D(\alpha, \beta) = \int \exp\left[-\frac{\beta}{2} \sum errors^2 - \frac{\alpha}{2} \|\mathbf{w}\|^2 \right] d\mathbf{w}$.

Since the likelihood function represents the situation of an ideal model, Equation 4 can be used as a target for choosing optimal weights. The weight vector **w**, corresponding to the maximum of the posterior distribution can be found by minimising the negative logarithm of Equation 4 with respect to the weights. This simply means that a new criterion for picking weight values (network training) given hyperparameters values α and β has been arrived at.

Once a network has been trained, one can evaluate the probability distribution of network predictions, ***y****, from inputs, ***x****, with the following integration:

$$p(y^*|\mathbf{x}^*, D) = \int p(y^*|\mathbf{x}^*, \mathbf{w}) p(\mathbf{w}|D) d\mathbf{w} \quad \textbf{(5)}$$

assuming a likelihood for a new dataset in the form of Equation 3. This is an integration over weight space and is therefore of the same dimension as the number of weights. In practice, the calculation is unfortunately not tractable, which means that approximations must be employed. A common approach is to approximate Equation

5 with a spherical Gaussian distribution around a mode of the posterior (see [2, 3] for more details). Once $p(y^*|\mathbf{x}^*, D)$ has been approximated, its expected value can be used as the network prediction, its variance will provide error bars (confidence intervals) that take into account the noise on the target data as well as the width of the posterior distribution of the weights.

The choice of hyperparameters in the likelihood and prior affects the prediction capability of the final trained network. If, for example, the α parameter in Equation 2 has been set too large, the function implemented by the network might not be flexible enough to describe the data. If the hyperparameter β in Equation 2 is set too large, the network may be modelling the noise, if it is set too small, the model may not be capturing the whole picture. This means that hyperparameters need to be dealt with in a rigorous manner.

The Bayesian way to deal with the uncertainty that comes from choosing specific hyperparameters is to remove their influence from any of the calculations through marginalisation (integrating the hyperparameters out). Through marginalisation one can avoid the problem of choosing specific hyperparameters. Unfortunately, the integrals required for this are usually intractable given any reasonable choice of priors for the hyperparameters. Alternatively, the problem of selecting hyperparameters can be viewed as another optimisation problem, this is achieved through the *evidence procedure* [3]. The evidence procedure makes the assumption that $p(\alpha, \beta|D)$ is sharply peaked around the most probable values of α and β, which means that finding these most probable values leaves the analytical expression for $p(\mathbf{w}|D)$ unchanged from Equation 5. The procedure is, therefore, to find the hyperparameters that optimise $p(\mathbf{w}|D)$ and then fix them for all further calculations that include $p(\mathbf{w}|D)$.

One distinct advantage of the evidence procedure is that one can gain information from the optimised hyperparameters using *automatic relevance determination (ARD)* [3]. If one sets a separate α hyperparameter for each input to the network, after optimisation, it will represent the inverse variance of all of the weights on the units that fan out from that input. This can be used to gauge the importance of each input to the prediction being made; if an optimised hyperparameter is small, this means that large weights are allowed and that the corresponding input is important.

The implementation of the neural networks in this work relies on the Netlab toolbox for Matlab [3], which is freely available for download from Aston University's website [4].

PRELIMINARY RESULTS

Each of the preliminary models developed has been tested using drop test data. Figure 2 shows a typical telescopic port main landing gear (MLG) structure that has undergone drop testing using a drop test rig. Drop tests are performed to verify the dynamic compression damping and energy absorption characteristics of the landing gear shock absorber [5]. In a drop test, the landing gear is mounted in a fixture that geometrically represents the aircraft landing gear attachment structure. The landing gear is dropped from various heights onto a ground reaction platform. The drop

height is set to achieve the required vertical descent velocity. The correct proportion of landing weight is supported by the moving carriage and wing lift is simulated by upward acting jacks. Prior to the drop, the wheels are spun to simulate the aircraft landing speeds. Landing attitude is varied by angling the landing gear in the fixture or angling the ground reaction platform. Loads are measured on the landing gear using strain gauges and the ground-to-tyre loads are measured using loads cells in the ground reaction platform.

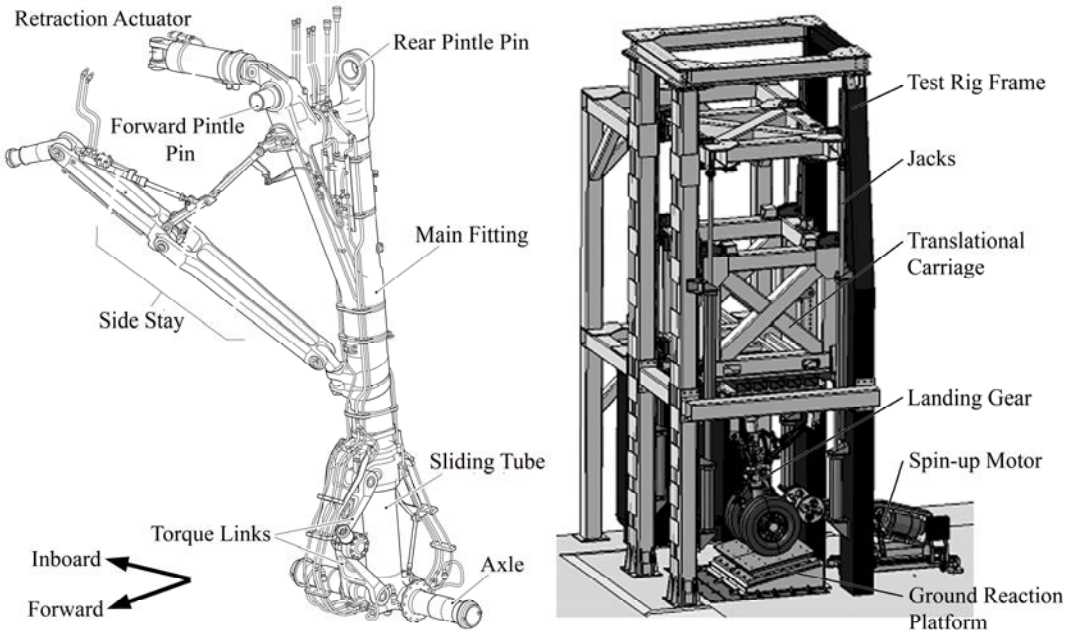

Figure 2. Main Landing Gear Structure and Drop Test Rig, after [6].

A number of MLP neural networks have been developed using landing gear drop test data. The inputs from the drop test data include shock absorber travel, tyre closure, shock absorber pressure, wheel speed, drop carriage accelerations, landing gear accelerations, while the initial output target to be predicted is side stay load. The inputs and outputs have been normalised to ensure that no variable is considered more important by the neural network simply because of the magnitude of its measurements.

After the MLP networks have been trained, a testing data set (which does not overlap the data used for network training) is used to judge the prediction performance of the networks on unseen data. Figure 3 shows the predictions of a side stay load from a trained classic MLP network for a test set of input data. The actual measured side stay load is also plotted for comparison. The results of this implementation are very encouraging; the MSE of this prediction is a very low 8.53%. In a similar trial with a Bayesian MLP, the prediction capabilities proved to be just as good, and in this case the confidence intervals are calculated for each prediction. Figure 4 shows a closer view of the prediction of a trained Bayesian MLP on the same test set of data shown in Figure 3. A zoomed view has been selected to get a clearer view of the confidence intervals, which are at a 3σ level.

The preliminary neural network models have also been trained for the prediction of vertical, drag and side ground-to-tyre loads. These ground-to-tyre loads are measured through the ground reaction platform during the drop tests. These preliminary results are very encouraging, and demonstrate that the preliminary models developed are fit for purpose. The true test of course will come later if flight test data is considered.

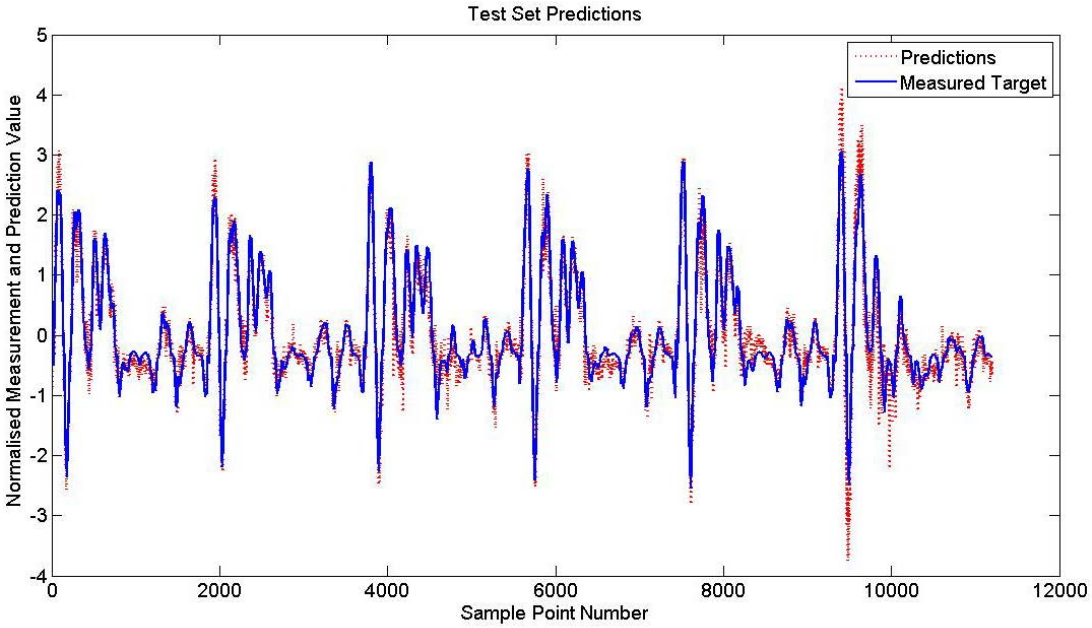

Figure 3. Side Stay Load Prediction from Preliminary Neural Network.

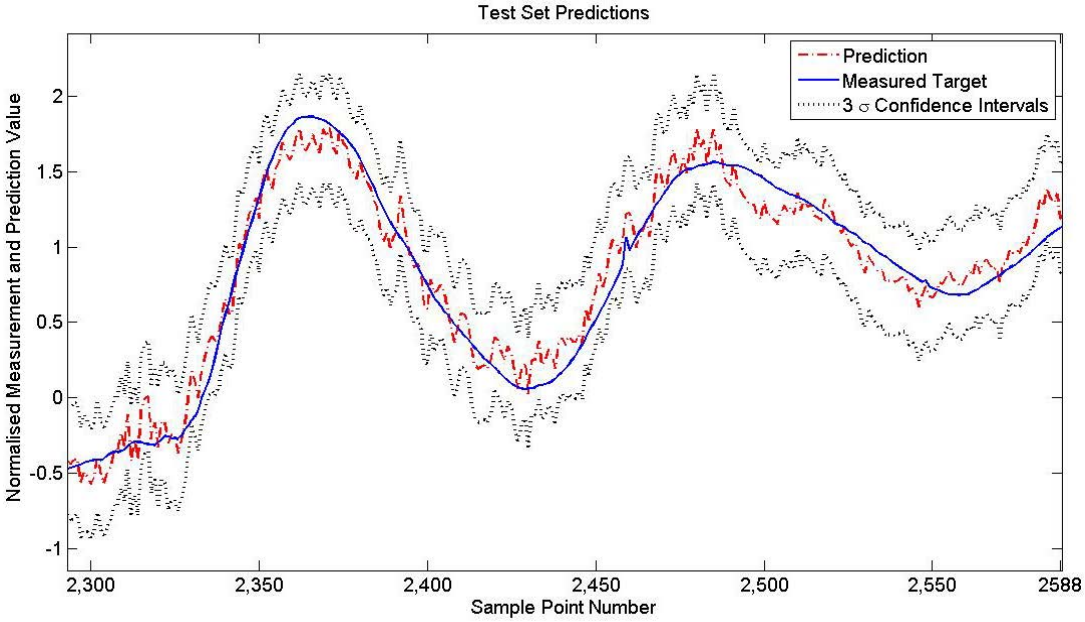

Figure 4. Zoomed View of Side Stay Load Prediction from Bayesian Neural Network.

CONCLUSIONS AND FUTURE PLANS FOR MODEL DEVELOPMENT AND TESTING

In conclusion, the MLP models developed at this preliminary stage have performed well for the prediction of the side stay load in the drop test data. The implementation of a Bayesian MLP has also provided a means of assessing confidence on any predictions made. In the next stages of this work, the model will be developed further and tested more rigorously using drop test data from different test set ups, simulated landing data and flight test data. Other mathematical modelling approaches will also be used in the development of high fidelity models including Gaussian Process regression, which, like the Bayesian MLP, will account for the uncertainty in the model predictions. Finally, dynamic modelling will be explored. While the current models provide a static map between input and target variables, the aim is to develop the capability for any dynamic behaviour to be accounted for by the models.

ACKNOWLEDGEMENTS

This work has been carried out through the Monitoring of Aircraft Component Health (MACH) project funded by Messier-Bugatti-Dowty and the U.K. Technology Strategy Board.

REFERENCES

[1] Bishop, C.M., *Neural Networks for Pattern Recognition*. 1995: Oxford university press.

[2] Bishop, C.M., *Pattern Recognition and Machine Learning*. 2006: Springer New York.

[3] Nabney, I., *NETLAB: Algorithms for Pattern Recognition*. 2002: Springer Verlag.

[4] Nabney, I., *Netlab Neural Network Software*, Aston University, Editor, http://www1.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/downloads.

[5] Messier-Bugatti-Dowty Design Guide, *Drop, Free Extension and Gas Spring Verification Tests and Correlation. DSG-4208*. 2011.

[6] *Messier-Bugatti-Dowty Qualification Test Report, QTR00015-1*. 2009.