Flight test flutter prediction using neural networks

W.J. Crowther, J.E. Cooper, School of Engineering, University of Manchester, M13 9PL
2 June, 2000

Abstract

Flight flutter testing is a crucial part in the certification of a prototype aircraft. To aid the clearance process, a number of different methods have been proposed to determine the speed at which flutter occurs based upon data obtained during the flight envelope expansion. However, the most commonly used approach is simply to extrapolate the estimated damping ratios. In this paper, a method is proposed for the prediction of damping ratios during flight test using a neural network trained on model data. The proposed method is compared with a simple statistical extrapolation approach and the effects of noise investigated. For noise-free data, the neural network method shows improved accuracy compared to the statistical method. With noisy data, the accuracy of the statistical method is unacceptably poor, however the accuracy of the neural network method remains good as long the network is trained with noisy data.

Keywords

Flutter, neural networks, flight test

Nomenclature

f	mode frequency
m	training vector index
n	test point index
$\mathbf{t}$	neural network target output vector
$\mathbf{x}$	neural network input vector
$\mathbf{y}$	neural network output vector
V	aircraft speed, kts
ζ	mode damping ratio

Abbreviations

kts	knots (1 knot = 0.514 m/s)
MLP	Multi Layer Perceptron

1 INTRODUCTION

Flutter is a violent unstable oscillation that occurs due to the interaction of inertial, aerodynamic and elastic forces. Flight Flutter Testing [1] is a mandatory part of an aircraft's certification process that must be undertaken in order to demonstrate that the aircraft is flutter free throughout the desired flight envelope. The flutter testing procedure [2,3] can be difficult due to the conflicting requirements of completing the tests safely in as short a time as possible.

The aim of the flutter test procedure is to clear the flight envelope shown in figure 1 at various constant height and constant Mach number flight conditions. The procedure is made up of three stages:

1. The aircraft is excited in some manner and the response measured

2. The modal parameters are estimated using system identification methods [4]
3. The decision is made to proceed to the next flight test point

Arguably, it is the last element that is the most difficult. In essence, the flutter test engineer needs to ensure that an adequate degree of stability will be maintained at the next test point, and this is usually determined by considering the modal parameters estimated from the measured test data. Any deficiencies in the first two elements of the test will be passed through to the prediction stage causing greater uncertainty. It must also be remembered that the stability of the system can change abruptly with only a small increase in flight speed. If a computational aeroelastic model is available, it is generally only used for guidance regarding the characteristics of the flutter behaviour, and it is rare to update the model as the envelope is expanded.

Within Industry, the most common approach to stage three of the flight flutter test procedure is to simply extrapolate the damping trends by hand. There is somewhat of an art to this procedure, particularly when there are closely coupled modes. It is often useful to include information about the mode shapes, see [5] for an automated procedure to do this, in order to reduce problems with the mode tracking.

A number of methods have been proposed that attempt to automate the flutter speed prediction from test data process. The Flutter Margin method [6] is based on the premise that a more fundamental stability criterion (the Routh Stability criterion) should be used rather than simply tracking the damping ratios. Originally, the method was formulated for binary systems and the user has to know a-priori which of the modes are going to form the flutter mechanism. A recent extension [7] has produced a formulation for ternary flutter.

The envelope function [8] was developed based on the assumption that the impulse response function contained information about the overall stability of the system. A shape function based upon the centroid of the impulse response envelope is plotted and then extrapolated to predict the flutter speed. Similarly, a time domain ARMA method [9,10] using the Jury Stability criterion has been tried that also considers the overall stability of the system. This method was developed for the response to unmeasured turbulent excitation although the method can be extended to the measured input case.

Nissim and Gilyard [11], extending a method introduced by Gaukroger et al. [12], adopted a different approach by attempting to identify the entire aeroelastic system. If data from two different flight test points is measured, then it is possible to identify all of the matrices in the conventional aeroelastic model. Once the model is known, an eigenvalue solution of the system equations can be performed at increasing speeds until the flutter speed is found. Although this approach is the most ideal from a mathematical viewpoint, there are problems in its implementation with large order systems. Also, recent work [13] has developed an approach for determining the flutter speed based upon worst case robust stability theory.

In this paper, an approach is introduced to determine the stability at the next flight test point using a neural network. The network is trained using modal parameter estimates from a computational aeroelastic model. During the flight test, previous frequency and damping values are used to help predict damping values at future test points. The method is intended to be used during flight flutter testing as an additional tool to aid the safe expansion of the flight envelope. This is distinct from other recent neural network approaches to flutter prediction e.g. [14], which are used at the design rather than flight test stage to aid prediction of flutter speeds for a range of model parameters.

2 Aeroelastic modelling

2.1 Description of model

A numerical model of a four engined, high wing transport aircraft was used in this work [8]. The model describes the dynamics of the aircraft through structural and aerodynamic modal inertia, damping and stiffness matrices. The structural components were derived from a condensed finite element model whereas the aerodynamic terms were obtained from a panel method. The symmetric and anti-symmetric motions of the structure are mutually independent and can be considered separately. For this case symmetric motion was considered.

At any particular flight condition, the eigenvalues of the system matrix obtained from the first order form of the system equations can be used to estimate the frequencies and damping ratios. The corresponding eigenvectors were used to compute the mode shapes at 17 positions on the model. This process was repeated at a number of different flight conditions and the frequencies and damping ratios plotted against speed. Care has to be taken with tracking the frequency and damping values in order to ensure that the correct curves are drawn, particularly when the values cross over each other. Various schemes were used here, including the use of mode shapes and/or the position of the roots on the Argand Plane. Manual adjustment of the plots was still required at a few test points to ensure that the frequency and damping curves corresponded to the correct mode.

2.2 Simulation results

Damping and frequency data for six structural modes of the baseline model for the speed range 250 to 500 kts at 20000 feet altitude are shown in figure 2. These six modes contain the critical flutter mechanism, occurring at around 390kts, that consists of the interaction of modes 9 and 11, involving wing bending and engine heave motions. Although mode 10 has very similar frequency and damping characteristics to mode11 throughout the speed range, its mode shape behaviour is very different and is not involved in the flutter. In order to aid clarity of presentation of the results, it was decided to omit mode 10 from the dataset. This subset of modes is used to illustrate the neural network methodology proposed in this paper, whereas in practice more modes would be considered. Modes with frequencies below and above those considered did not contribute to the flutter mechanism and were also omitted.

3 Problem formulation

For the present aircraft model, values of mode damping and frequency are available at speed intervals of 15kts. As a basis for the present work, it was decided that the proposed methods should predict the damping for any given mode at a point three speed intervals ahead of the current speed. The prediction should be based on an input vector comprised of information available at the present and previous four test points, figure 3. The generalised form of the problem is thus expressed as

$$\varsigma_{n+3}^{m} = g\left(V_n, \varsigma_n^m, f_n^m, V_{n-1}, \varsigma_{n-1}^m, f_{n-1}^m, \cdots, V_{n-4}, \varsigma_{n-4}^m, f_{n-4}^m, V_{n+3}\right) \qquad (1)$$

Where ς_{n+3}^{m} is the target damping value to be predicted, *n* is the index of the current test point, *m* is a given mode and *g*( ) is an unknown function to be determined.
For convenience, equispaced test points are used in the present study. However, this is not a requirement of the method.

Note that the choice of predictor structure given by equation 1 is strongly problem dependent. In the present case, a three step ahead prediction based on four previous values was chosen on the basis that a flight test engineer could reasonably expect to obtain useful results from this information using manual graphical extrapolation techniques.

4 Statistical approach

A simple statistical approach was developed to act as a control against which the neural network method could be compared. The statistical method was based on prediction of damping values by extrapolation of a polynomial fitted to known damping data. Note that this approach does not use any information contained within the mode frequency data. The problem may be thus expressed now as

$$\varsigma_{n+3}^{m} = h\left(V_n, \varsigma_n^m, V_{n-1}, \varsigma_{n-1}^m, \cdots, V_{n-4}, \varsigma_{n-4}^m, V_{n+3}\right) \qquad (2)$$

where $h(\ )$ is the adopted statistical algorithm.

Curve fitting and subsequent extrapolation was implemented using the Matlab *polyfit* and *polyval* functions. Evaluation of extrapolation errors for different order polynomials showed that the best overall performance was achieved using a second order fit.

5 Neural Network approach

5.1 Background

Neural networks are now routinely used in for a variety of roles in engineering, commerce and banking. Applications include series prediction, pattern recognition, fault diagnosis and modelling [16,17, 18]. As a general rule, neural networks find useful roles in problems that are based on finding relationships in noisy and/or nonlinear data, with the further prerequisite that there should be abundant data available for neural network training [19, 20, 21].

A neural network is a collection of interconnected processing nodes that provides a mapping between an input space and an output space. The nature of this mapping is related to the type of processing element used, the way in which processing elements are connected together and the strength, or weight, of the connection between each element. The process of training a neural network is that of obtaining a set of connection weights for a given network structure that provides the best mapping between input and output training data sets [22, 23, 24]. For the present work, this weight set is optimal[20] when the error function $E = \sum_m \left(\mathbf{y}^m - \mathbf{t}^m\right)^2$ is minimised for all training vector pairs *m*, where $\mathbf{y}^m$ is the actual output of the network in response to an input $\mathbf{x}^m$ and $\mathbf{t}^m$ is the target (ideal) output of the network for input $\mathbf{x}^m$.

5.2 Neural network architecture

The neural network architecture used for the present work is the Multi-Layer Perceptron (MLP) trained using error back propagation. With three layers and nonlinear activation functions used for the second two layers, this type of network can approximate any continuous bounded function to an arbitrary accuracy, given sufficient training data [25, 26]. Note that similar results can be obtained using three layer radial basis function networks. A MLP network architecture was chosen for reasons of convenience.

A schematic of the chosen neural network implementation is shown in figure 4. The number of neurons in the input layer of the network is dictated by the number of elements in the input vector, whereas the number of neurons in the output layer is always equal to one. For input vectors containing damping and frequency data, 11 input neurons were required: 5 damping values, 5 frequency values and the current speed. Since the data points were equispaced in speed, the other speed values were redundant. The number of neurons in the hidden layer is analogous to the order of fit used in statistical regression and is therefore problem dependent. A suitable number of hidden layer neurons was found by the method recommended in reference 21. The number of hidden neurons was initially set to one and the network was trained until the stopping condition was reached (see section 5.3). The number of hidden neurons was then incremented by one, the network weights randomised and the network retrained. As the number of hidden neurons increases, the reduction in the trained network error becomes vanishingly small at the expense of increased training time. The final choice of number of hidden neurons is a balance between elapsed training time and network accuracy.

5.3 Training strategy

In line with neural network best practice[21], data available for network training was divided into three independent datasets: a training set, a test set and a validation set. The network was trained using the training data set. As training progressed, the network error was periodically evaluated using the test set. Initially the test set error decreases in line with the training set error. However, after a certain number of iterations, overfitting of the training data occurs, causing the test set error to increase. Training was automatically terminated at this point. Finally, the performance of the trained network was evaluated using the validation dataset.

5.4 Neural network implementation

Neural networks were trained and evaluated using the commercial software package Neuframe[27] running on a 400MHz Pentium PC. Typical training times were of the order of 4 hours.

5.5 Dataset design .

Given the problem formulation in equation 1, the task is to generate a set of neural network training vectors that provide examples of the correct mapping between the input data and the target damping value. Given that the aeroelastic model will only be an approximation of the real aircraft, it is necessary to provide training data from a range of models all with slightly different parameters such that the parameter space of the real aircraft is adequately covered. To achieve this, elements of the model mass matrix were perturbed with +/-10% random noise and complete damping and frequency data sets obtained for 9 different models. Eight of these data sets comprised the training data and the remaining one was used as the independent test case (i.e. a case not used to train the network).

The training data set is illustrated in figure 5. Note that the damping data for mode 9 shows two differing characteristics. The straight-line part is in fact mode 10 but the eigenvalue sorting algorithm has failed to determine this. Similarly, mode 12 shows a number of very different damping characteristics that are partly due to incorrect eigenvalue sorting. It should be noted that correct eigenvalue sorting is an inherently hard problem for complex dynamic models. For this reason, it was decided that the proposed neural network method should be tested with data that included incorrectly sorted eigenvalues.

To improve the numerical conditioning of the neural network method, it was necessary to impose artificial upper and lower bounds on the damping and frequency data. Damping values were clipped at +0.1 and -0.04, and frequency data clipped at +12, -0. The data shown in figure 5 has been clipped.

Since real aircraft damping and frequency data will be noisy, a second version of the training set was created with random noise added of amplitude +/-5% of the clipping range.

6 Results and discussion

6.1 Noise-free data

A comparison between actual damping values and damping values predicted based on the statistical regression method for the modes of interest is shown in figure 6. The polynomial extrapolation works well for cases where the curvature of the damping-speed data is constant, e.g. modes 6 and 11. However, for modes where there are large changes in curvature of the damping-speed data, the regression-based prediction is relatively poor. In particular, for mode 9 (the first flutter mode), the flutter speed is over-predicted by 25kts (6.4% of true flutter speed). This margin of error would compromise safety in flight test. Note that increasing the order of fit of the polynomial does not improve accuracy.

As a direct comparison with the statistical method, a neural network was trained and evaluated using just damping data (i.e. the frequency data was omitted), figure 7. These results are slightly better than the statistical results of figure 8, however the neural network predicted damping still tends to lag behind the true value (mode 9). Further network training or use of increased hidden neurons fails to improve these results indicating that there is insufficient discrimination in the training data.

With the inclusion of the frequency data in the neural network training and evaluation, the prediction accuracy is much improved, as would be expected, figure 8. In particular, notice that the predicted flutter speed for mode 9 is within approximately 5% of the true flutter speed. The use of both damping and frequency information improves the method because a more complete model of the system dynamics is available to the neural network during training. This result also suggests that the statistical method could be improved by using multivariable regression.

The validation cases chosen for the present study do not exhibit any eigen value sorting errors. Note that the inclusion of validation cases with eigen value sorting errors does not affect the accuracy of the statistical approach. However, with the neural network approach, the use of validation cases that are not well represented in the training data reduces the accuracy achieved.

6.2 Noisy data

Prediction results for the statistical method operating on noisy data are shown in figure 9. The accuracy of the results is greatly reduced compared to the equivalent noise-free results. Clearly, the regression-based statistical method employed is very noise-sensitive and not appropriate in the present case.

To investigate the noise-sensitivity of the neural network approach, a network trained with noise-free data was queried with noisy data, figure 10. The results shown here are clearly very poor, with accuracy much worse than the statistical method operating on the same data. This highlights the need to train neural networks with data that matches the data expected during flight test as closely as possible, including the noise.

With a neural network trained using noisy data in the first instance, the accuracy of the results of querying with noisy data is improved greatly, figure 11. Whilst there is increased scatter in the results, the damping trends are followed faithfully for all modes apart from 12, and the predicted flutter speed for mode 9 is of the same accuracy as that achieved with noise-free data.

7 Conclusions

- A neural network approach to extrapolation of aeroelastic mode damping values with increasing airspeed has been demonstrated and compared with a regression method
- The regression method works well for noise-free data when there is little change in curvature of the damping data with speed. However, where the change in curvature is high, for example around the mode flutter speed, the regression method shows unacceptable inaccuracy.
- For the neural network method, with training based on damping data only, the prediction results are only slightly better than the results from the regression method.
- The use of damping *and* frequency data to train the neural network leads to a significant increase in prediction accuracy. This is because the neural network is now predicting on the basis of a more complete dynamic model of the system.
- With 5% noise added to the data, the regression method shows an unacceptable degradation in accuracy
- The accuracy of the neural network method decreases only slightly with the addition of noise to the data. However, the network must be trained with noisy data initially
- The results are encouraging and the approach will be applied to real flight test data.

Acknowledgements

The authors gratefully acknowledge the help of Greg Dimitriadis in the development of the aeroelastic model used in this work.

References

1. Kehoe, M.W. ‘A Historical Overview of Flight Flutter Testing’ In AGARD CP-566 Advanced Aeroservoelastic Testing and Data Analysis Paper 1. 1995.
2. Wright, J.R. ‘Flight Flutter Testing’ Lecture Series on Flutter of Winged Aircraft. Von Karman Institute. 1991.
3. Johnson, J. ‘Flutter Testing of Modern Aircraft’ AIAA Student Journal Spring pp 6-11. 1989.
4. Cooper, J.E. ‘Parameter Estimation Methods for Flight Flutter Testing’ In AGARD CP-566. Advanced Aeroservoelastic Testing and Data Analysis. Paper 10. 1995.
5. Desforges, M.J., Cooper, J.E. & Wright, J.R. ‘Mode Tracking During Flutter Testing using the Modal Assurance Criterion’ Proc I.Mech.E Part G J.Aerospace Engineering v210 pp 27-37. 1996.
6. Zimmermann, N.H. & Weissenberger, J.T. ‘Prediction of flutter onset speed based on flight testing at subcritical speeds’ J. Aircraft v1 n4 pp 190 – 202 1964.
7. Price, S.J. & Lee, B.H.K. ‘Evaluation and Extension of the Flutter Margin methods for Flight Flutter Prediction’ J. Aircraft v30 n3 pp 395- 402 1993.
8. Cooper, J.E., Emmett, P.R., Wright, J.R. & Schofield, M.J. ‘Envelope Function – A tool for analyzing Flutter Data’ J. Aircraft v30 n5 pp 785 – 790 1993.
9. Matsuzaki, Y. & Ando, Y. ‘Estimation of Flutter Boundary from Random Responses due to turbulence at subcritical speeds’ J. Aircraft v18 n10 pp862-868. 1981.
10. Matsuzaki, Y. & Torii, H. ‘Response Characteristics of a Two Dimensional Wing Subjected to Turbulence near the Flutter Boundary’ J.Sound & Vibration v136 n2 pp 187 – 199 1990.
11. Nissim, E. & Gilyard, G.B. ‘ Method for Experimental Determination of Flutter Speed by Parameter Identification’ AIAA-89_1324-CP 1989.
12. Gaukroger, D.R., Skingle, C.W. & Heron, K.H. ‘An Application of System Identification to Flutter Testing’ J. Sound & Vibration v72 n2 pp 141-150. 1980.
13. Lind, R.C. & Brenner, M.J. ‘A Worst Case Approach for On-Line Flutter Prediction’ Int. Forum on Aeroelasticity & Structural Dynamics, Rome 1997 pp 79-86.
14. Lecce, L., Pucci, M. & Pecora, M. ‘Flutter Speed Prediction using the Artificial Neural Network Approach’ Int. Forum on Aeroelasticity & Structural Dynamics, Manchester, Paper 13 1995.
15. Hancock, G.J. , Wright, J.R. and Simpson, A. ‘On the Teaching of the Principles of Wing Flexure – Torsion Flutter’ Aeronautical Journal, pp 285 – 305 1985.
16. Crowther, W.J., Edge, K.A., et al, 'Fault diagnosis of a hydraulic actuator circuit using neural networks - An output vector space classification approach', *Proc. Instn. Mech. Engrs*, Vol 212 Part I, pp 57-68, February 1997.
17. Azoff, E. Michael, Eitan Michael. - Neural network time series forecasting of financial markets / E. Michael Azo. - Chichester : Wiley, 1994. - (Wiley finance editions). - 0471943568
18. Tarassenko, L. A guide to neural computing applications. *Arnold*, 1998.
19. Gurney, K., An introduction to neural networks, *UCL Press London* 1997. ISBN 1-85728-503-4
20. Bishop, C., Neural networks for pattern recognition, *Oxford University Press*, ISBN 0-19-853849-9, 1996.

21. Masters, T. Practical neural network recipes in C++, *Academic Press*, ISBN 0-12-479040-2, 1993.
22. Minsky, M. and Papert, S. Perceptrons - An introduction to computational geometry, *MIT Press*, London, 1969, ISBN 0-262-630222
23. Rumelhart, D., Hinton, G. and Williams, J., Learning internal representations by error propagation, In Parallel Distributed Processing, Explorations in the Microstructure of Cognition. Vol. 1 foundations, *MIT Press*, London, 1987, ISBN 0-262-18120-7.
24. Rumelhart, D.E., Widrow, B., Lehr, M.A. The basic ideas in neural networks, *Communications of the ACM*, Vol. 37, No. 3, pp. 86-92, Mar 1994.
25. Blum, E. and Li, L. Approximation theory and feed-forward networks, *Neural Networks*, 4:4, pp. 511-515.
26. Hornik, K., Stinchcombe, M., and White, H. ‘Multilayer feedforward networks are universal approximators’, Neural Networks, 2:5, p. 359-366, 1989.
27. http://www.neusciences.com/

List of figures

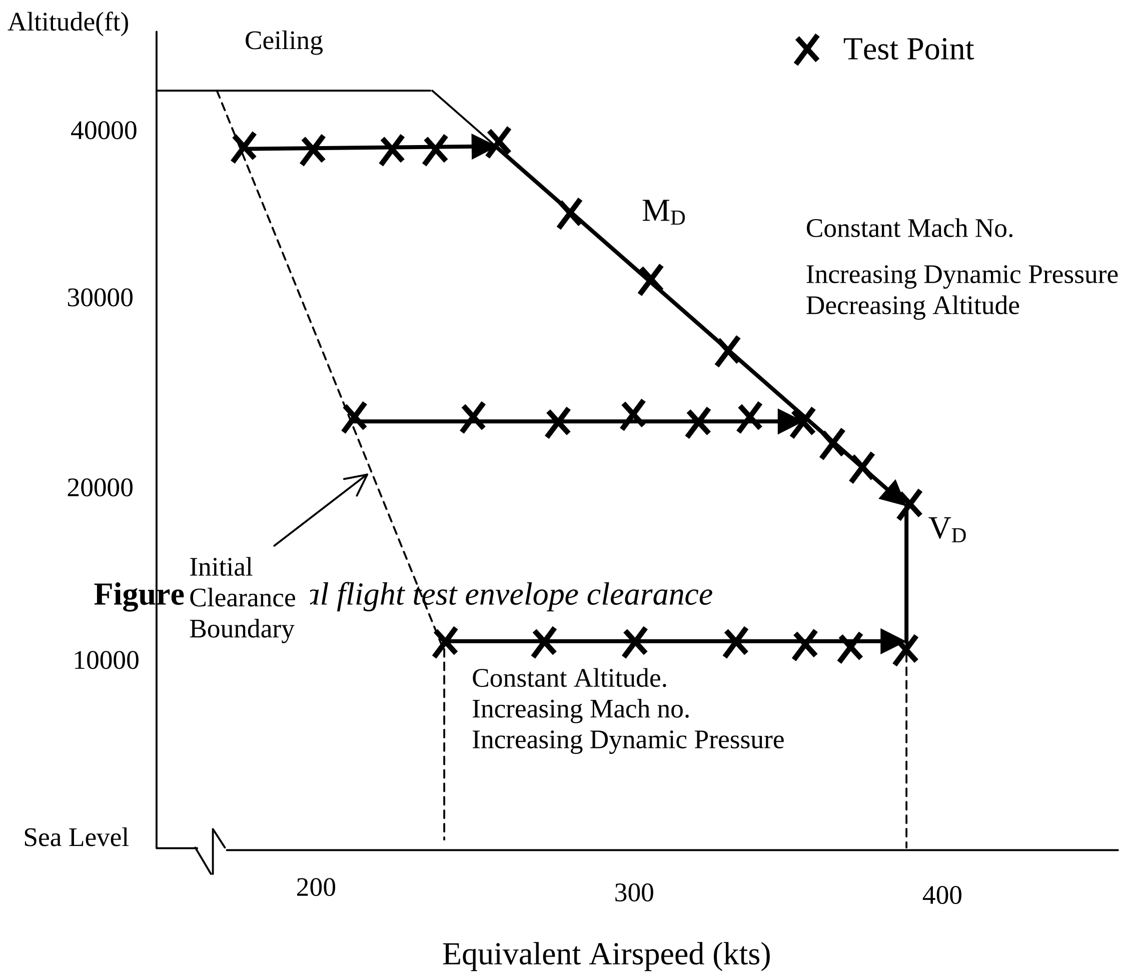

Figure *al flight test envelope clearance*

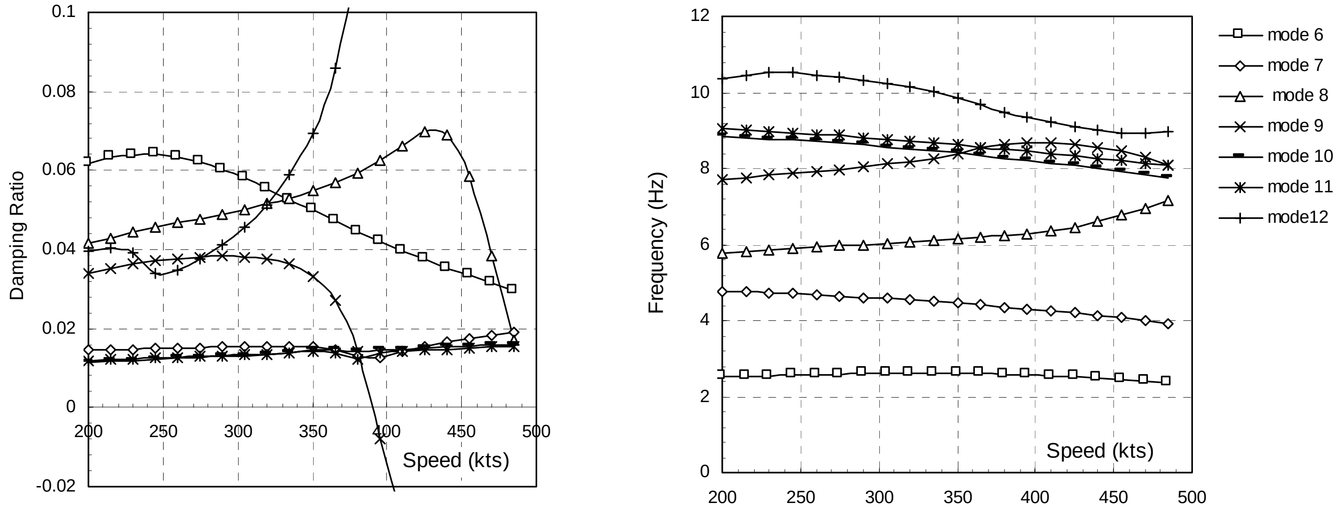

Figure 2 *Baseline model damping and frequency data, modes 6 to 12*

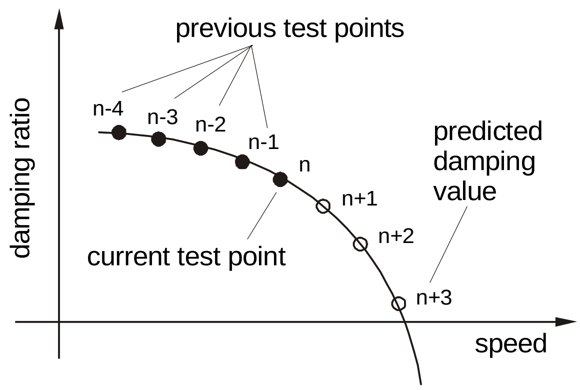

Figure 3 *Prediction of damping trends using existing flight test information.*

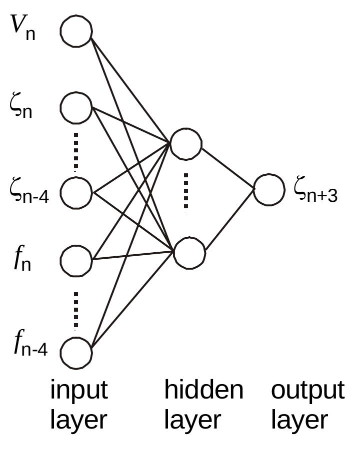

Figure 4 *MLP neural network implementation for predicting damping trends.*

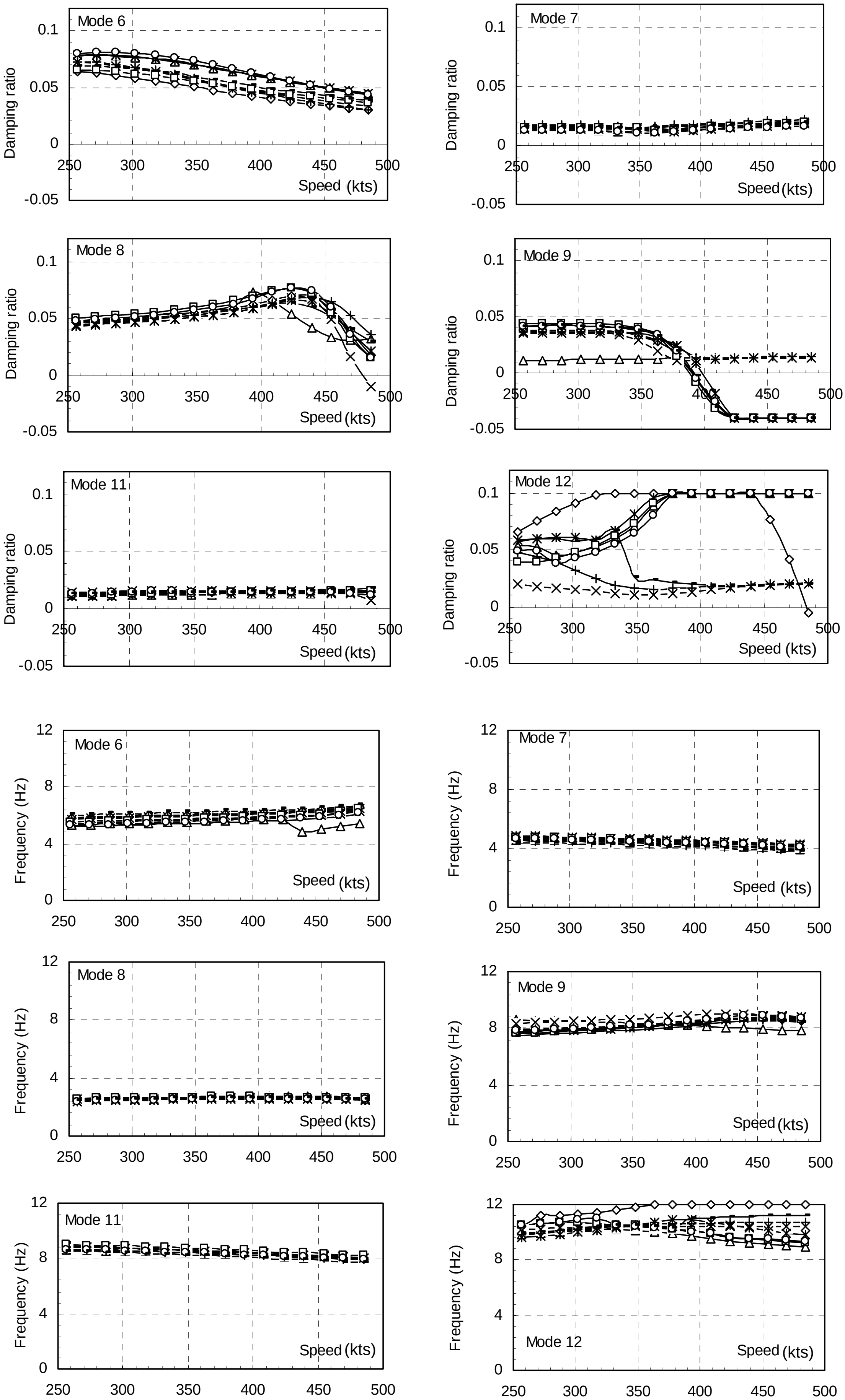

Figure 5 *Damping and frequency data used for neural network training. Training cases generated by +/- 10% random variation in the model mass matrix.*

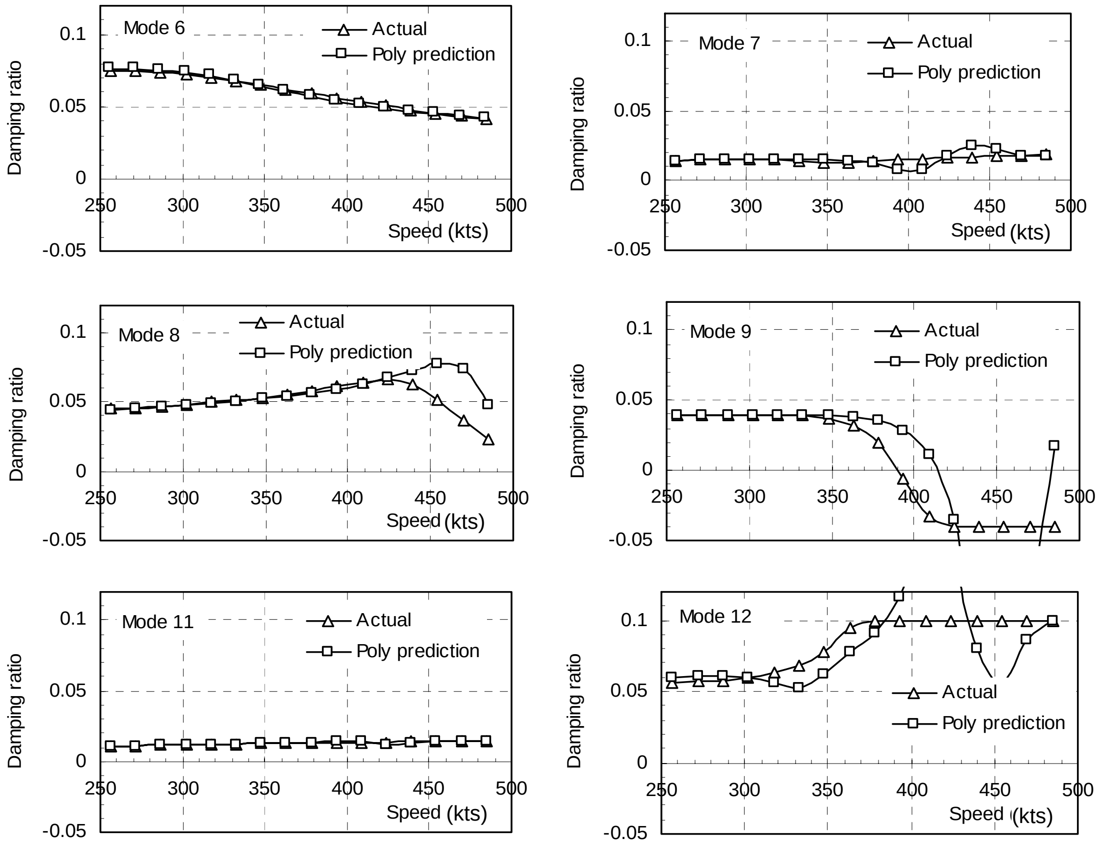

Figure 6 *Three-step-ahead statistical prediction of mode damping ratio using a second order polynomial extrapolation, noise-free data*

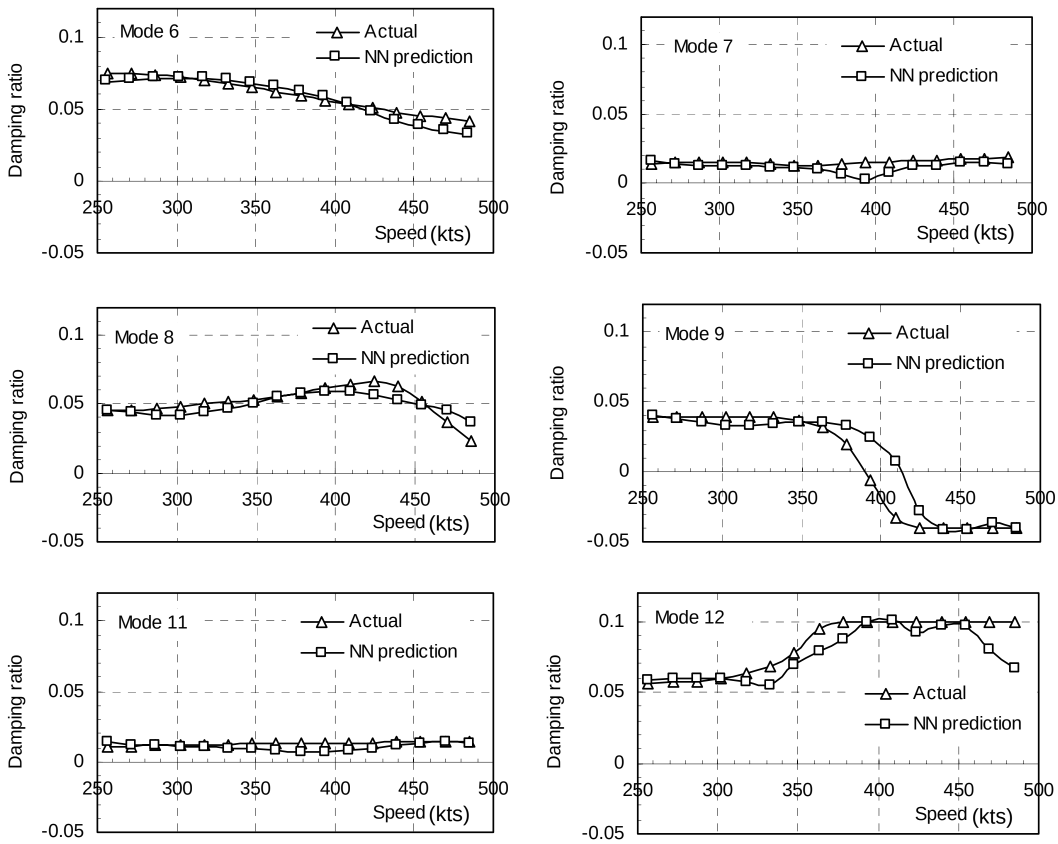

Figure 7 *Three-step-ahead prediction of mode damping ratio using neural network trained using damping data only(noise-free)*

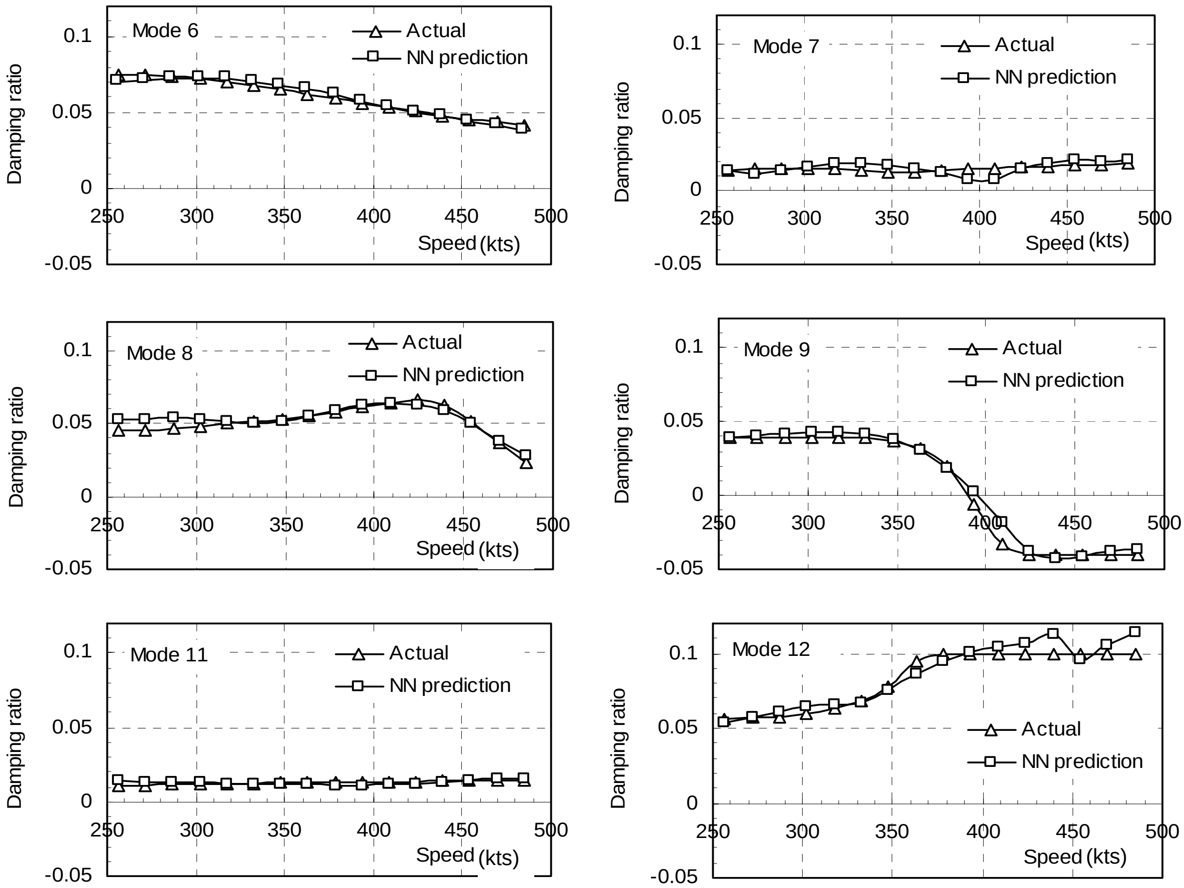

Figure 8 *Three-step-ahead prediction of mode damping ratio using neural network trained using damping and frequency data (noise-free)*

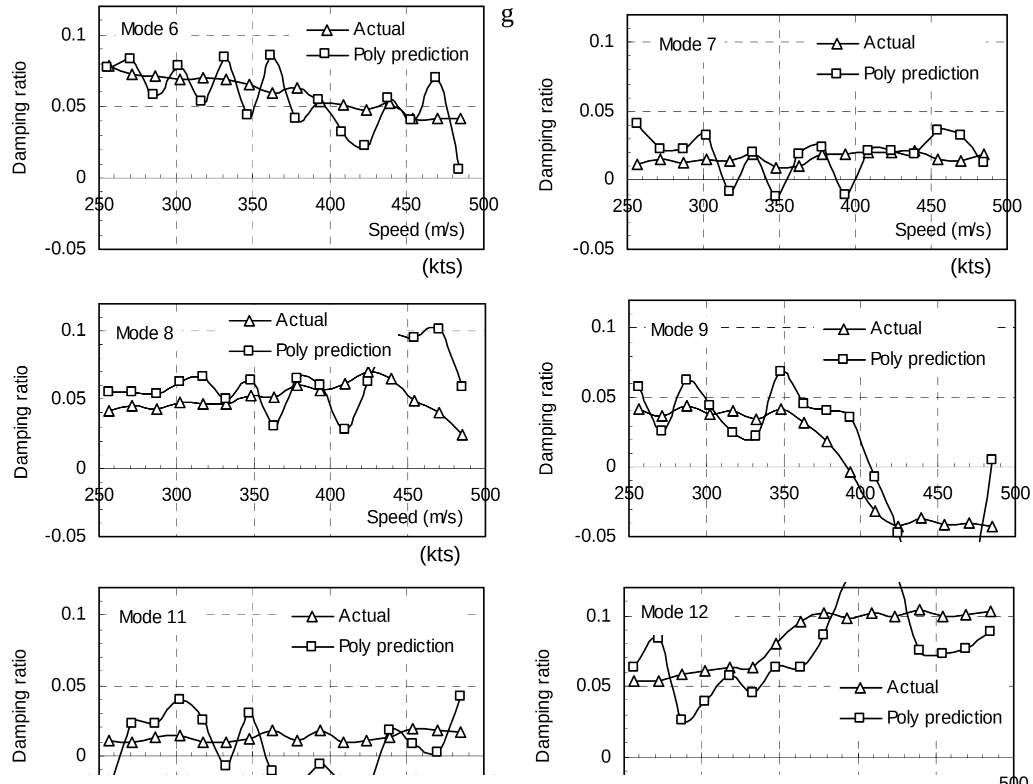

Figure 9 *Three-step-ahead prediction of mode damping ratio using a second order polynomial extrapolation, data.with 5% of full scale*

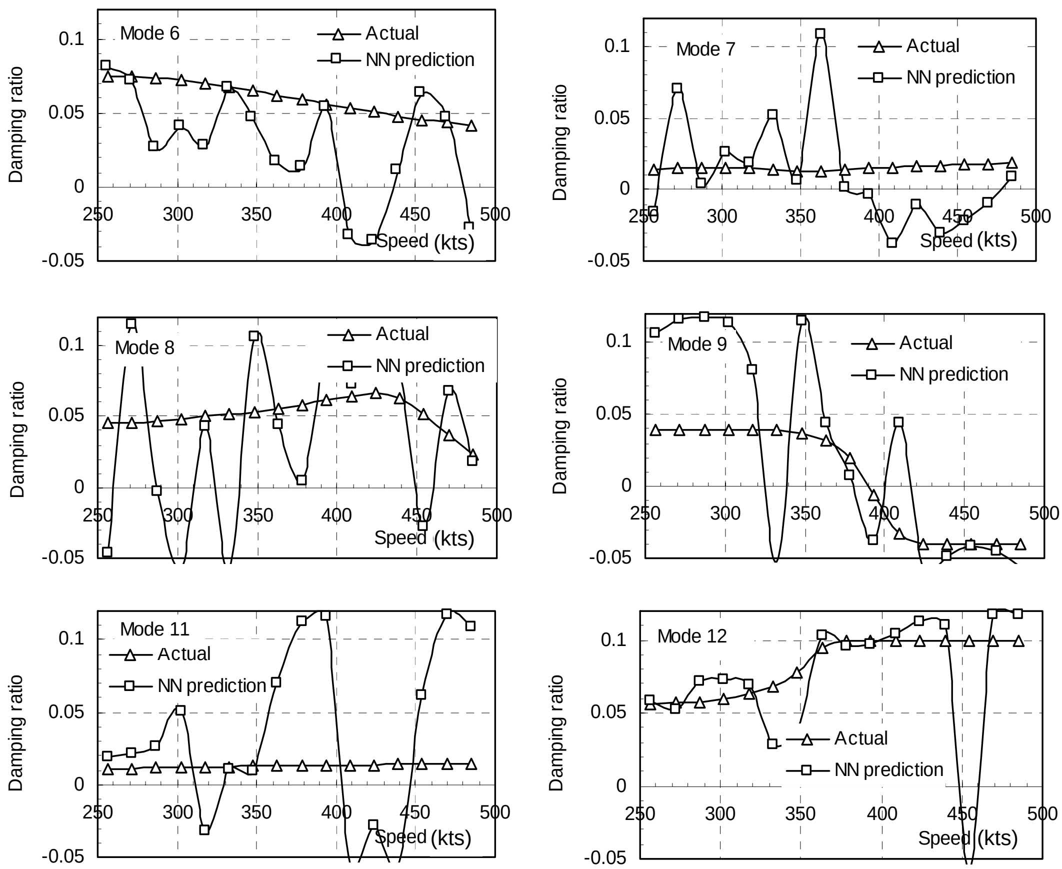

Figure 10 *Three-step-ahead prediction of mode damping ratio using a neural network trained with noise-free data but queried with noisy data*

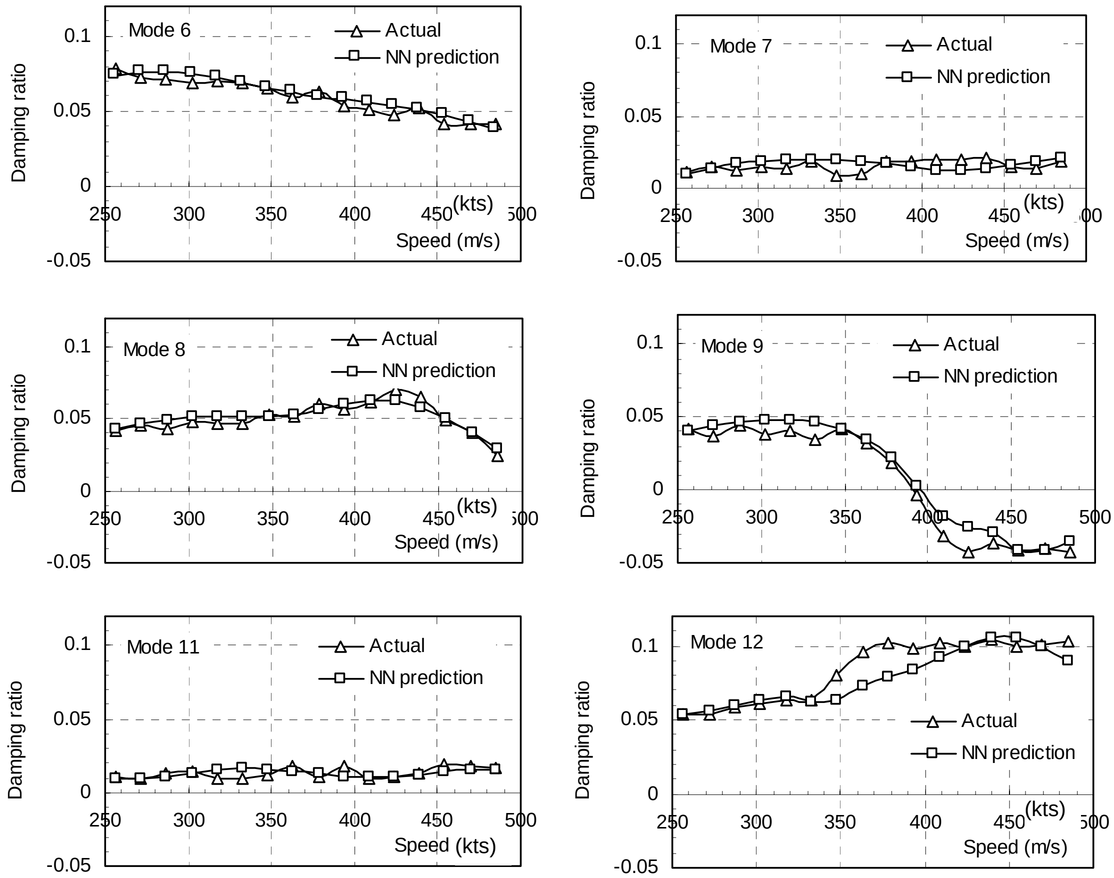

Figure 11 *Three-step-ahead prediction of mode damping ratio using a neural network trained with noisy data and queried with noisy data*