Fault classification in structures with incomplete measured data using autoassociative neural networks and genetic algorithm

Tshilidzi Marwala[1] and S. Chakraverty[2,*]

[1]School of Electrical and Information Engineering, University of the Witwatersrand, Private Bag 3, Wits, 2050, South Africa
[2]B.P.P.P. Division, Central Building Research Institute, Roorkee 247 667, India

A method for fault classification in mechanical systems in the presence of missing data entries is introduced. The method is based on autoassociative neural networks where the network is trained to recall the input data through some nonlinear neural network mapping. From the trained network an error equation with missing inputs as design variables is constructed. Genetic algorithm is used to solve for the missing input values. The proposed method is tested on a fault classification problem in a population of cylindrical shells. It is found that the proposed method is able to estimate single-missing-entries to the accuracy of 93% and two-missing-entries to the accuracy of 91%. The estimated values were then used in the classification of faults and the fault classification accuracy of 94% was observed for single-missing-entry cases and 91% for two-missing-entry cases while the full database set is able to give classification accuracy of 96%.

Keywords: Fault classification, genetic algorithm, neural networks.

NEURAL networks have been used in conjunction with vibration data with varying degrees of success to identify faults in structures[1–7]. Neural networks approximate functions of arbitrary complexity using training data. Supervised neural networks are used to represent a mapping from an input vector onto an output vector, while unsupervised networks are used to classify the data without prior knowledge of the classes involved. The most common neural network architecture is the multilayer perceptron (MLP), which is trained using the back-propagation technique[8]. An alternative network is the radial basis function (RBF)[8]. Both the MLP and RBF have been using for fault identification in structures but it was generally observed that MLP performs better than RBF[9]. This is due to the fact that the RBF usually requires the implementation of the pseudo-inverse of a matrix for training, which is often singular while MLP uses conventional optimization methods, which are stable[8].

*For correspondence. (e-mail: sne_chak@yahoo.com)

Lopes *et al.*[10] successfully implemented impedance methods and neural networks for structural health monitoring. Marwala[11] used probabilistic committee of neural networks to classify faults in a population of nominally identical cylindrical shells. The probabilistic neural networks were trained using the hybrid Monte Carlo[12] and an accuracy of 95% was observed in classifying 8 classes fault cases. Chen *et al.*[13] successfully used neural networks and response only data for fault diagnosis in structures. Wu *et al.*[14] used an MLP neural network to identify damage in a model of a three-story building. Damage was modelled by reducing member stiffness by between 50% and 75%. The input to the neural network was the Fourier transform of the acceleration data, while the output was the level of damage in each member. The network was able to diagnose damage within 25% accuracy. Levin and Lieven[15] applied a RBF neural network and modal properties to identify errors in the finite element model of a cantilevered beam. The method was found to give good identification of faults even with a limited number of experimentally measured degrees of freedom and modes. Atalla and Inman[16] successfully trained a RBF neural network using frequency response functions in order to identify faults in structures. Marwala and Hunt[17] successfully applied multi-layer perceptron neural networks and finite element models to identify faults in cantilevered beam. Atalla and Inman[16] trained a RBF neural network using frequency response functions in order to identify faults in structures. Suresh *et al.*[18] used modular neural network approach to identify crack location in a cantilever beam while Reddy and Ganguli[19] used Radial Basis Function neural networks in a helicopter rotor blade. Pawar and Ganguli[20] used genetic fuzzy system for damage detection in beams and helicopter rotor blades.

When these neural networks are applied in real life situation, one of the main problems encountered is the issue of sensor failure. If one of the sensors fails then the neural network is unable to make a decision because it only works with a complete input set. What is normally done is to use the average value of that sensor calculated over some defined period in the past and hope that the next time around

the sensor will be available. In the literature there is no method proposed thus far that takes account of the absence of entries of inputs to the neural networks for fault classification in structures. It must be noted, however, that the issue of estimating missing data has been implemented in other areas in mechanical systems such as validating the gas-path sensor data[21] and therefore this paper contributes to the field of structural mechanics a procedure that has been implemented in gas dynamics. In this paper we propose a method of estimating missing entries in the database that is based on autoassociative models[22] combined with genetic algorithm for data estimation and subsequently fault identification in structural mechanics. The proposed method is tested on a classification of faults in a population of nominally cylindrical shells.

Mathematical background

Neural networks

In this study we use neural networks to construct the autoassociative neural networks, which are networks with inputs and output being the same[22]. There are several types of neural network architectures and in this paper we focus on the MLP and the MLP architecture contains a hyperbolic tangent basis function in the hidden units and linear basis functions in the output units[8]. A schematic illustration of the MLP is shown in Figure 1. The relationship between the output y and input x can be written as follows[8].

$$y_k = \sum_{j=0}^{M} w_{kj}^{(2)} \tanh\left(\sum_{i=0}^{d} w_{ji}^{(1)} x_i \right). \quad (1)$$

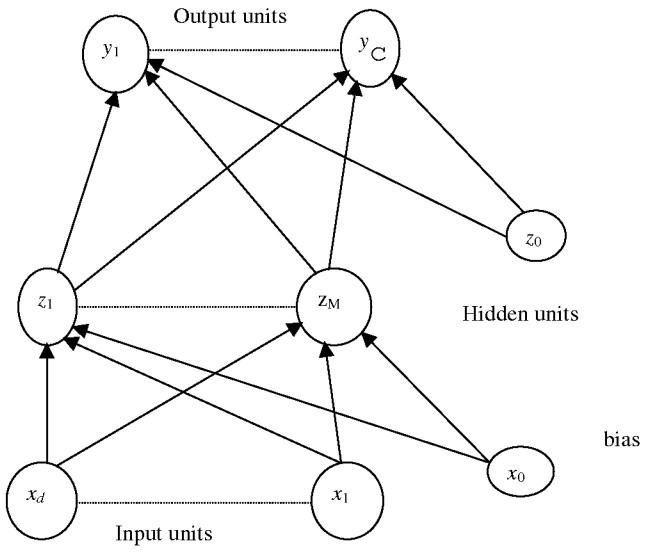

Figure 1. Feed-forward network having two layers of adaptive weights.

Here, $w_{ji}^{(1)}$ and $w_{kj}^{(2)}$ indicate weights in the first and second layer, respectively, going from input i to hidden unit j, M is the number of hidden units and d is the number of output units.

The model in Figure 1 is able to take into account the intrinsic dimensionality of the data. Models of this form can approximate any continuous function to arbitrary accuracy if the number of hidden units M is sufficiently large. Training the neural network identifies the weights in eq. (1). A cost function must be chosen to identify the weights in eq. (1). A cost function is a mathematical representation of the overall objective of the problem. In this paper, the main objective is to construct the cost function that identifies a set of neural network weights given the measured data. If the training set $D = \{x_k, t_k\}_{k=1}^{N}$ is used and assuming that the targets y are sampled independently given the inputs x_k and the weight parameters, w_{kj}, the cost function, E, may be written as follows using the sum-of-squares of errors cost function[8]:

$$E = \sum_{n=1}^{N} \sum_{k=1}^{K} \{t_{nk} - y_{nk}\}^2 , \quad (2)$$

where t is the target data, N is the number of training examples and K is the number of outputs.

Before network training is performed, the network architecture needs to be constructed by choosing the number of hidden units, M. If M is too small, the neural network will be insufficiently flexible and will give poor generalization of the data because of high bias. However, if M is too large, the neural network will be unnecessarily flexible and will give poor generalization due to a phenomenon known as over-fitting caused by high variance. In this study to minimize the equation, the scaled conjugate gradient method is used[23] in conjunction with back-propagation[8]. The scaled conjugate gradient method is an optimization procedure that is based on conjugate gradient method but uses optimized mathematical expressions to reduce the computational intensity of conjugate gradient method. It must, however, be noted that there are no material differences in accuracy of the results between the scaled conjugate gradient, conjugate gradient and other gradient-based optimization methods. The only difference between these methods is the computational efficiency and the scaled conjugate gradient method was chosen because of its computational efficiency.

Autoassociative networks and missing data

Autoassociative networks are models where the network is trained to recall the inputs[17]. This means that whenever an input is presented to the network the output is the predicted input. These networks have been used in a number of applications including novelty detection, feature selection and data compression[24–28]. It must be noted, however,

that on applying autoassociative neural networks for data compression the network has fewer nodes in the hidden layer. However, it must be noted that for missing data estimation it is absolutely crucial that the network must be as accurate as possible and that this accuracy is not necessarily realized through few hidden nodes as is the case when these networks are used for data compression. It is therefore crucial that some process of identifying the optimal architecture must be used. Using eq. (1) autoassociative memory networks may be formulated by setting input x to be equal to output y. Equation (1) may thus be re-written in simplified form as:

$$\{y\} = f(\{w\},\{x\}) . \tag{3}$$

Here $\{y\}$ is the output vector, $\{x\}$ is the input vector, f is a function and $\{w\}$ is the mapping weight vector. Given the fact that $\{x\} = \{y\}$, eq. (3) may thus be re-written as follows:

$$\{x\} = f(\{w\},\{x\}) . \tag{4}$$

For a perfectly mapped system, eq. (4) holds, however, for a realistic mapping there will be some error, and thus eq. (4) may be re-written as:

$$\{e\} = \{x\} - f(\{w\},\{x\}) . \tag{5}$$

The sum of squares of both the left hand side and the right hand side of eq. (5) will give:

$$E = \sum_{i=1}^{c} (\{x\} - f(\{w\},\{x\}))^2 , \tag{6}$$

where c is the size of the input vector. For a situation when not all the inputs are known, then the input data may be divided into known x_{kw} and unknown components x_u and thus eq. (6) may be written as follows:

$$E = \sum_{i=1}^{c} \left(\begin{Bmatrix} x_u \\ x_{kw} \end{Bmatrix} - f\left(\{w\}, \begin{Bmatrix} x_u \\ x_{kw} \end{Bmatrix} \right) \right)^2 . \tag{7}$$

From eq. (7), the unknown component x_u data is estimated from the known component x_{kw} by minimizing the error in eq. (7). It is absolutely important that a global minimum error be achieved because a local minimum error results with the incorrect estimation of the unknown component x_u. In this study a global optimum method, genetic algorithm, is used to find the global optimum solution[29]. The next section thus explains the genetic algorithm.

Genetic algorithms

Genetic algorithm (GA) was inspired by Darwin's theory of natural evolution[29,30]. Genetic algorithm is a simulation of natural evolution where the law of the survival of the fittest is applied to a population of individuals. This natural optimization method is used for optimization, and in this paper, to minimize the error in eq. (7), genetic algorithm is implemented by generating a population and creating a new population by performing the following procedures: (1) crossover; (2) mutation; and (3) reproduction[29]. The crossover operator mixes genetic information in the population by cutting pairs of chromosomes at random points along their length and exchanging over the cut sections. This has a potential of joining successful operators together. Arithmetic crossover technique[29,30] is used in this paper. Arithmetic crossover is implemented by taking parents $p1$ and $p2$ and a random number a sampled uniformly between 0 and 1, and create children $c1$ and $c2$ to be $c1 = p1.a + p2.(1-a)$ and $c2 = p1.(1-a) + p2.a$, respectively. The mutation operator picks the chromosomes at random and inverts it. This has a potential of introducing to the population new information. In this paper, non-uniform mutation[30] is used. Non-uniform mutation changes one of the parameters of the parent based on a non-uniform probability distribution. It is implemented by picking a variable or 'gene' of the chromosomes at random and increasing or decreasing it towards its upper or lower bound, respectively. This has a potential of introducing new information to the population. It changes one of the parameters of the parent based on a non-uniform probability distribution. A Gaussian distribution is used and starts with a higher variance, and narrows to a point distribution as the current generation approaches the maximum generation. Reproduction takes successful chromosomes and reproduces them in accordance to their fitness functions. In this paper normalized geometric selection method is used[30]. Normalized geometric selection method is a ranking selection function based on the normalized geometric distribution and it limits the probability of selecting the fittest individual especially at the early stages of the run process. With this method, the probability of selecting the ith individual in a population of size L is:

$$P = \frac{q(1-q)^{r-1}}{1-(1-q)^L} . \tag{8}$$

Here q is the probability of selecting the best individual and r is the rank of the individual where 1 is the best. In Goldberg's study of genetic algorithms in function optimization, a series of parametric studies across a five-function suite of problems suggested that good genetic algorithm performance requires the choice of a high crossover rate, a low mutation rate (inversely proportional to the population size), and a moderate population size[30].

Missing entry methodology

As described earlier, the missing entry methodology here combines the autoassociative neural networks and opti-

mization method, viz. the genetic algorithm. The proposed method is shown in Figure 2. It is implemented by determining the number of missing entries and calling it *N*. Then the missing entry objective function with *N* variables is constructed using eq. (7). Genetic algorithm is then used to minimize the missing entry objective function and optimum solution is the estimated values of the missing variables. The missing entry objective function in eq. (7) can also be solved using gradient-based approach since the gradient of the error function can easily be calculated using back-propagation[8]. However, the gradient-based methods are not global methods and therefore are not used in this study. They can, however, be used to fine-tune the solution given by the genetic algorithm. However, preliminary investigation in this study indicated that fine-tuning the genetic algorithm solution does not offer any advantages with regards to the estimation of the missing entries nor the accuracy of fault classification.

Dynamics

In this study, modal properties, i.e. natural frequencies and mode shapes are used for fault classification. For this reason these parameters are described in this section. Modal properties are related to the physical properties of the structure. All elastic structures may be described in terms of their distributed mass, damping and stiffness matrices in the time domain through the following expression[31]:

$$[M]\{X''\}+[C]\{X'\}+[K]\{X\}=\{F\}, \tag{9}$$

where $[M]$, $[C]$ and $[K]$ are the mass, damping and stiffness matrices respectively, and $\{X\}$, $\{X'\}$ and $\{X''\}$ are the displacement, velocity and acceleration vectors, respectively, while $\{F\}$ is the applied force vector. If eq. (9) is transformed into the modal domain to form an eigenvalue equation for the ith mode, then[31]:

$$(-\varpi_i^2[M]+ j\varpi_i[C]+[K])\{\bar{\phi}\}_i=\{0\}, \tag{10}$$

where $j=\sqrt{-1}$, ϖ_i is the ith complex eigenvalue, with its imaginary part corresponding to the natural frequency ω_i, $\{0\}$ is the null vector, and $\{\bar{\phi}\}_i$ is the ith complex mode shape vector with the real part corresponding to the normalized mode shape $\{\phi\}_i$. From eq. (10) it may be deduced that the changes in the mass and stiffness matrices cause changes in the modal properties of the structure. Therefore, the modal properties can be identified through the identification of the correct mass and stiffness matrices.

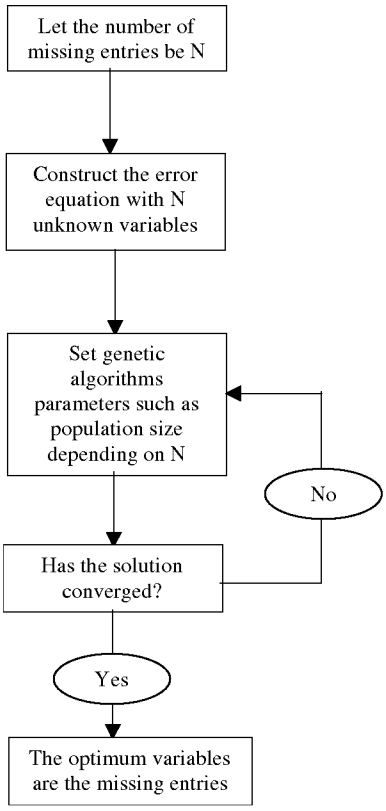

Figure 2. A schematic diagram indicating the implementation of the missing data estimator.

Example: cylindrical structure

In this section the procedure proposed is experimentally validated. The experiment is performed on a population of cylinders, which are supported by inserting a sponge rested on a *bubble-wrap*, to simulate a 'free-free' environment (see Figure 3) and the details of this may be found in ref. 9. The sponge is inserted inside the cylinders to control boundary conditions. This will be further discussed below. Conventionally, a 'free–free' environment is achieved by suspending a structure usually with light elastic bands. A 'free–free' environment is implemented so that rigid body modes, which do not exhibit bending or flexing, can be identified. These modes occur at frequency of 0 Hz and they can be used to calculate the mass and inertia properties. In the present study, we are not interested in the rigid body modes. Here, a 'free–free' environment is approximated using a bubble-wrap. Testing the cylinders suspended is approximately the same as testing it while resting on a bubble-wrap, because the frequency of cylinder-on-wrap is below 100 Hz. The first natural frequency of cylinders being analysed is over 300 Hz and this value is several orders of magnitudes above the natural frequency of a cylinder on a bubble-wrap. Therefore the cylinder on the wrap is effectively decoupled from the ground. It should be noted that the bubble-wrap adds some damping to the structure but the damping added is found to be small enough for the modes to be easily identified. When the damping ratios were estimated, it was observed that the structure was lightly damped and therefore damping did not play significant role in this paper. The

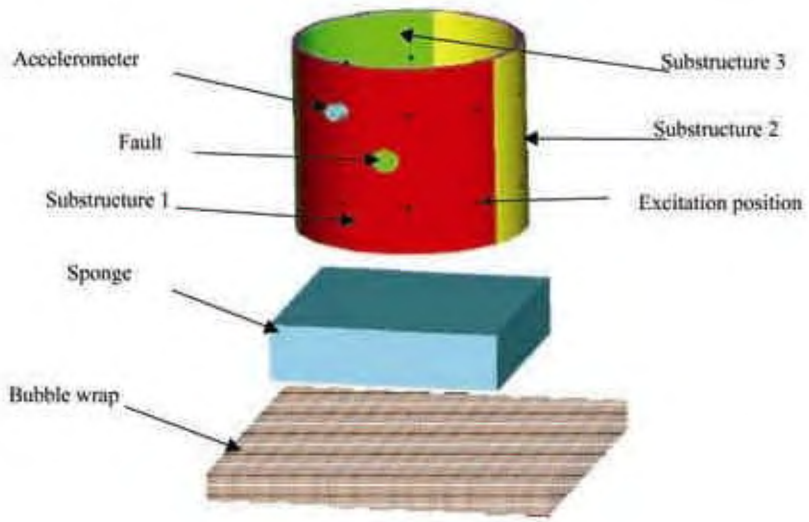

Figure 3. Illustration of a cylindrical shell showing the positions of the impulse, accelerometer, substructures, fault position and supporting sponge.

Table 1. Number of different types of fault-cases generated

Fault	[0 0 0]	[1 0 0]	[0 1 0]	[0 0 1]	[1 1 0]	[1 0 1]	[0 1 1]	[1 1 1]
Number	60	24	24	24	24	24	24	60

impulse hammer test is performed on each of the 20 steel seam-welded cylindrical shells (1.75 ± 0.02 mm thickness, 101.86 ± 0.29 mm diameter and of height 101.50 ± 0.20 mm). The impulse is applied at 19 different locations as indicated in Figures 3–9 on the upper half of the cylinder and 10 on the lower half of the cylinder. The sponge is inserted inside the cylinder to control boundary conditions and by rotating it every time a measurement is taken. The bubble wrap simulates the free–free environment. The top impulse positions are located 25 mm from the top edge and the bottom impulse positions are located 25 mm from the bottom edge of the cylinder. The angle between two adjacent impulse positions is 36°.

Problems encountered during impulse testing include difficulty of exciting the structure at an exact position especially for an ensemble of structures and in a repeatable direction. Each cylinder is divided into three equal substructures and holes of 10–15 mm in diameter are introduced at the centers of the substructures to simulate faults. For one cylinder the first type of fault is a zero-fault scenario. This type of fault is given the identity [0 0 0], indicating that there are no faults in any of the three substructures. The second type of fault is a one-fault scenario, where a hole may be located in any of the three substructures. Three possible one-fault scenarios are [1 0 0], [0 1 0], and [0 0 1] indicating one hole in substructures 1, 2 or 3 respectively. The third type of fault is a two-fault scenario, where a hole is located in two of the three substructures. Three possible two-fault scenarios are [1 1 0], [1 0 1], and [0 1 1]. The final type of fault is a three-fault scenario, where a hole is located in all three substructures, and the identity of this fault is [1 1 1]. There are 8 different types of fault-cases considered (including [0 0 0]).

Because the zero-fault scenarios and the three-fault scenarios are over-represented, twelve cylinders are picked at random and additional one- and two-fault cases are measured after increasing the magnitude of the holes. This is done before the next fault case is introduced to the cylinders. The reason why zero-fault and three-fault scenarios are over-represented is because all cylinders tested give these fault-cases, whereas not all cylinders tested give all 3 one-fault and 3 two-fault cases. Only a few fault-cases are selected because of the limited computational storage space available. For each fault-case, acceleration and impulse measurements are taken. The types of faults that are introduced (i.e. drilled holes) do not influence damping.

Each cylinder is measured three times under different directions by changing the orientation of a rectangular sponge inserted inside the cylinder. The number of sets of measurements taken for undamaged population is 60 (20 cylinders × 3 for different directions). All the possible fault types and their respective number of occurrences are listed in Table 1. In Table 1 it should be noted that the numbers of one- and two-fault cases are each 72. This is because as mentioned above, increasing the sizes of holes in the substructures and taking vibration measurements generated additional one- and two-fault cases.

The impulse and response data are processed using the Fast Fourier Transform to convert the time domain impulse history and response data into the frequency domain. The data in the frequency domain are used to calculate the FRFs. The sample FRF results from an ensemble of 20 undamaged cylinders are shown in Figure 4. This figure indicates that the measurements are generally repeatable at low frequencies and are not repeatable at high frequencies. Axisymmetric structures such as cylinders have re-

peated modes due to their symmetry[32]. In this paper, the presence of an accelerometer and the imperfection of cylinders destroy the axisymmetry of the structures. Therefore the problem of repeated natural frequencies is neatly avoided, thereby making the process of modal analysis easier to perform[33]. The problem of uncertainty of high frequencies is avoided by only using frequencies under 4000 Hz.

Testing the proposed procedure

From the data measured in the previous section, 10 parameters were selected. The autoassociative network with 10 inputs and 10 outputs was constructed and several numbers of hidden units were used as shown in Figure 5. As shown on this figure, it was found that 10 hidden units was the optimal network that gives the best prediction of the input data. It must be noted, however, that it is generally assumed that the best autoassociative network is the one that has the lowest possible number of hidden units[25]. However, in this study, i.e. the case of missing data estimation, this factor must not be taken for granted and it is recommended that a separate study, like the one conducted here, should be used to determine the optimal autoassociative network. This is because for missing data estimation it was found that the success of the procedure is determined by how accurate the networks are and the accuracy does not necessarily only occur when the size of hidden nodes is small. As indicated before, the autoassociative network was trained using scaled conjugate method[25].

The first experiment consisted of cases where one of the inputs to the neural network was assumed to be unknown and then estimated using genetic algorithm method. On implementing genetic algorithm, the arithmetic cross-over, non-uniform mutation and normalized geometric selection as described before were used. On implementing arithmetic cross-over several parameters needed to be chosen and these were bounds and the probability of cross-over. The bounds were determined from the maximums and minimums of historical values of the particular data point while the probability of cross-over was chosen to be 0.75 as suggested in ref. 29. On implementing mutation the parameters that needed to be chosen were the bounds, and these were chosen as was done for cross-over, and the probability of mutation, that was chosen to be 0.0333 as recommended by Goldberg[30]. Genetic algorithm had a population of 20 and was run for 25 generations.

The proposed method for the case of one missing data per input set, estimated the missing value to the accuracy of 93%. When the proposed method was tested for the case with two missing data per input set, the accuracy of the

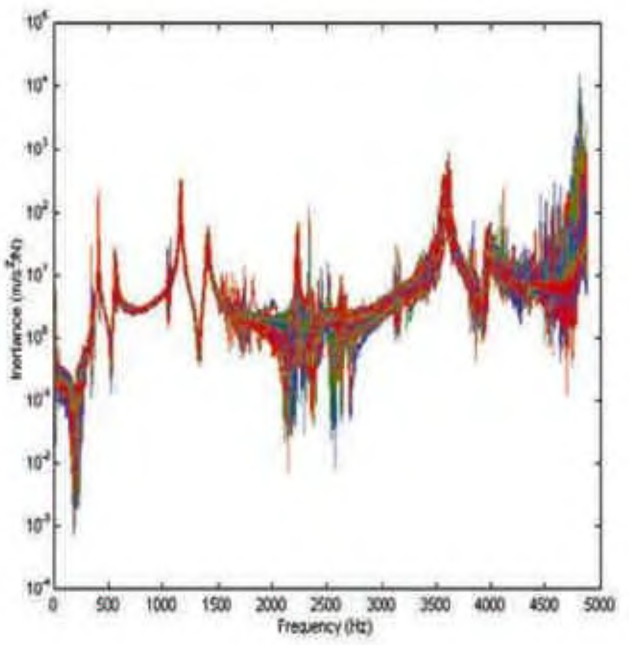

Figure 4. Measured frequency response functions from a population of cylinders.

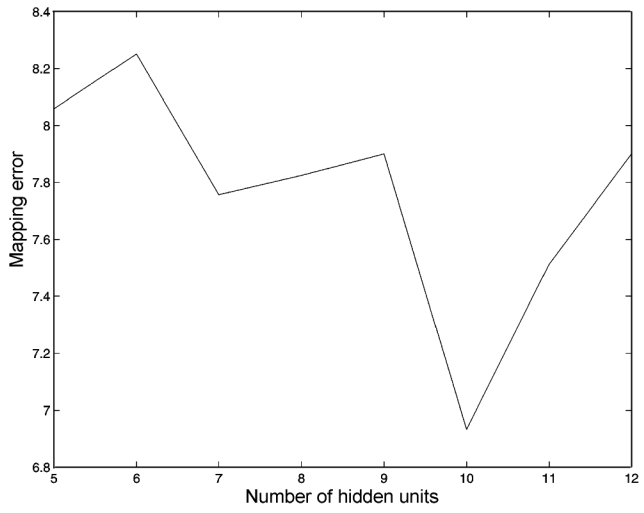

Figure 5. Measured and estimated missing value.

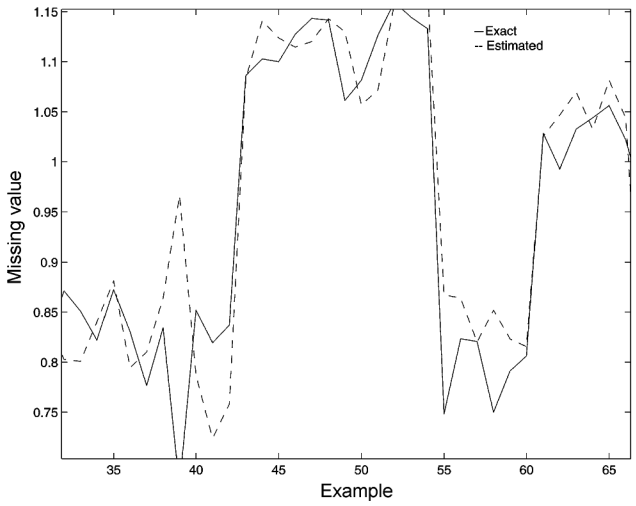

Figure 6. Prediction error versus the number of hidden nodes.

estimated values is 91%. The estimated values together with the accurate values are also indicated in Figure 6. This figure illustrates that the proposed missing data estimator gives the results that are consistent and accurate. In fact the data in this figure shows the correlation between the estimated data and the correct data to be 0.9.

In many cases the estimated values are intended for a particular reason, and in this paper they are intended to fulfil the goal of fault classification in a population of cylinders. The estimated values were, therefore, used for the classification of faults in a population of cylindrical shells and the fault classification accuracy of 94% was observed for a one-missing-entry case and 91% for the two-missing-entry case. When the complete database was used the fault classification accuracy of 96% was achieved.

The sources of errors in the experiment are measurement errors, modal analysis and neural network training. To minimize these errors reliable instruments were used for measuring data, reliable software were used for signal processing and modal analysis and standard procedures were used for training, generalization and testing of neural networks. The impact of these errors on the quality of results, was to a degree that it did not compromise the quality of the results.

Conclusion

In this study, a method based on autoassociative neural networks and genetic algorithms is proposed to estimate missing entries in data. This procedure was tested on a population of cylindrical shells. The proposed method is able to estimate single-missing-entries to the accuracy of 93% and two-missing-entries to the accuracy of 91%. Furthermore, fault classification accuracy of 94% was observed for single-missing-entry cases and 91% for two-missing-entry cases while the full database set is able to give classification accuracy of 96%.

1. Zang, C. and Imregun, M., Combined neural network and reduced FRF techniques for slight damage detection using measured response data. *Arch. Appl. Mech.*, 2001, **71**, 525–536.
2. Waszczyszyn, Z. and Ziemianski, L., Neural networks in mechanics of structures and materials – new results and prospects of applications. *Comp. Struct.*, 2001, **79**, 2261–2276.
3. Fang, X., Luo, H. and Tang, J., Structural damage detection using neural network with learning rate improvement. *Comp. Struct.*, 2005, in press.
4. Marwala, T., Fault classification using pseudo modal energies and probabilistic neural networks. *J. Eng. Mech.*, 2004, **130**, 1346–1355.
5. Marwala, T., Fault classification using pseudo modal energies and neural networks. *Am. Inst. Aero. Astron. J.*, 2003, **41**, 82–89.
6. Szewczyk, Z. P. and Hajela, P., Damage detection in structures based on feature sensitive neural networks. *ASCE J. Comput. Civil Eng.*, 1994, **8**, 163–179.
7. Doebling, S. W., Farrar, C. R. and Prime, M. B., Summary review of vibration-based damage identification methods. *Shock Vibration Digest*, 1998, **30**, 91–105.
8. Bishop, C. M., *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.
9. Marwala, T., Fault identification using neural networks and vibration data. Ph D thesis, University of Cambridge, UK, 2000.
10. Lopes, V., Park, G., Cudney, H. H. and Inman, D. J., Impedance-based structural health monitoring with artificial neural networks. *J. Intelligent Mater., Syst. Struct.*, 2000, **11**, 206–216.
11. Marwala, T., Probabilistic fault identification using a committee of neural networks and vibration data. *J. Aircraft*, 2001, **38**, 138–146.
12. Neal, R. M., Probabilistic inference using Markov chain Monte Carlo methods, University of Toronto Technical Report CRG-TR-93-1, Toronto, Canada, 1993.
13. Chen, Q., Chan, Y. W. and Worden, K., Structural fault diagnosis and isolation using neural networks based on response only data. *Comp. Struct.*, 2003, **81**, 2165–2172.
14. Wu, X., Ghaboussi, J. and Garret, J. H., Use of neural networks in detection of structural damage. *Comp. Struct.*, 1992, **42**, 649–659.
15. Levin, R. I. and Lieven, N. A. J., Dynamic finite element updating using neural networks. *J. Sound Vibr.*, 1998, **210**, 593–608.
16. Atalla, M. J. and Inman, D. J., On model updating using neural networks. *Mech. Syst. Signal Process.*, 1998, **12**, 135–161.
17. Marwala, T. and Hunt, H. E. M., Fault identification using finite element models and neural networks. *Mech. Syst. Signal Process.*, 1999, **13**, 475–490.
18. Suresh, S., Omkar, S. N., Ganguli, R. and Mani, V., Identification of crack location and depth in a centilever beam using a modular neural network approach. *Smart Mater. Struct.*, 2004, **13**, 907–916.
19. Reddy, R. R. K. and Ganguli, R., Structural damage detection in a helicopter rotor using radial basis function neural networks. *Smart Struct. Mater.*, 2003, **12**, 232–241.
20. Pawar, P. P. and Ganguli, R., Genetic fuzzy system for damage detection in beams and helicopter rotor blades. *Comp. Method. Appl. Mech. Eng.*, 2003, **192**, 2031.
21. Lu, P. J. and Hsu, T. C., Application of autoassociative neural network on gas-path sensor data validation. *J. Propul. Power*, 2002, **18**, 879–888.
22. Kramer, M. A., Autoassociative neural networks. *Comp. Chem. Eng.*, 1992, **16**, 313–328.
23. Møller, M., A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 1993, **6**, 525–533.
24. Hines, J. W., Uhrig, R. E. and Wrest, D. J., Use of autoassociative neural networks for signal validation. *J. Intelligent Robotic Syst.*, 1998, **21**, 143–154.
25. Kramer, M. A., Nonlinear principal component analysis using autoassociative neural Networks. *AIChE J.*, 1991, **37**, 233–234.
26. Upadhyaya, B. R. and Eryurek, E., Application of neural networks for sensor validation and plant monitoring. *Nuclear Technol.*, 1992, **97**, 170–176.
27. Jensen, C. A., El-Sharkawi, M. A. and Marks II R. J., Power system security assessment using neural networks: feature selection using Fisher discrimination. *IEEE Trans. Energy Conversion*, 2001, **16**, 757–763.
28. Reed, R. D. and Marks II R. J., *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, MIT Press, Cambridge, MA, 1999.
29. Holland, J., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
30. Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
31. Ewins, D. J., *Modal Testing: Theory and Practice*, Research Studies Press, Letchworth, UK, 1995.
32. Royston, T. J., Spohnholtz, T. and Ellington, W. A., Use of non-degeneracy in nominally axisymmetric structures for fault detection with application to cylindrical geometries. *J. Sound Vibration*, 2000, **230**, 791–808.
33. Maia, N. M. M. and Silva, J. M. M., *Theoretical and Experimental Modal Analysis*, Research Studies Press, Letchworth, UK, 1997.

ACKNOWLEDGEMENTS. T.M. thanks National Research Foundation, South Africa. S.C. thanks Department of Science and Technology, India and Director C.B.R.I. for funding under the Indo-South African Collaborative Project.

Received 8 April 2005; revised accepted 16 November 2005