Sādhanā Vol. 29, Part 3, June 2004, pp. 315–327. © Printed in India

Damage assessment in structure from changes in static parameter using neural networks

DAMODAR MAITY* and ASISH SAHA

Civil Engineering Department, Indian Institute of Technology – Guwahati, Guwahati 781 039, India
e-mail: damodar@iitg.ernet.in

MS received 6 May 2003; revised 29 March 2004

Abstract. Damage to structures may occur as a result of normal operations, accidents, deterioration or severe natural events such as earthquakes and storms. Most often the extent and location of damage may be determined through visual inspection. However, in some cases this may not be feasible. The basic strategy applied in this study is to train a neural network to recognize the behaviour of the undamaged structure as well as of the structure with various possible damaged states. When this trained network is subjected to the measured response, it should be able to detect any existing damage. This idea is applied on a simple cantilever beam. Strain and displacement are used as possible candidates for damage identification by a back-propagation neural network. The superiority of strain over displacement for identification of damage has been observed in this study.

Keywords. Back-propagation neural network; damage assessment; finite element method; mean square error.

1. Introduction

Structural systems in a wide range of aeronautical, mechanical and civil engineering fields are prone to damage and deterioration during their service life. So an effective and reliable damage assessment methodology will be a valuable tool in timely determination of damage and deterioration in structural members. The information obtained by a damage assessment process can play a vital role in the development of economical repair and retrofit programmes.

Most of the damage assessment methods proposed in the literature follow more or less the same approach. First, a mathematical model for the structure is constructed. This is then used to develop an understanding of the structural behaviour and to establish correlations between specific member damage conditions and changes in the structural response. These mathematical models are inherently direct process models, proceeding linearly from causes (damage location and extent) to effects (structural response). However, the identification of member damage from the response of the damaged structure is an inverse process; causes

*For correspondence

must be discerned from effects. Researchers have proposed damage assessment schemes based on analyses of measured responses of the structure before and after damage. Cawley & Adams (1979) used decrease in natural frequencies and increase in damping to detect cracks in fibre-reinforced plastics. Sanayei & Onipede (1991) used static displacement to determine damage under applied loads. In a truss model, damage is introduced by reduction of cross-sectional area. Pandey *et al* (1991) utilized curvature damage factor to determine damage in cantilevers and simply supported beam structures.

Recently, neural computing has been applied successfully in many fields. Most of these applications deal with problems of pattern recognition (Bishop 1998). Various researchers have started to experiment with neural networks (NNs) for damage identification purposes during the last decade (Worden 1996; Rytter & Kirkegaard 1997; Friswell & Mottershea 1999) as an alternative to the updating methods. NNs have some advantages that make them very attractive: the ability to treat damage mechanisms implicitly, and the capacity to generalize their responses and robustness in the presence of noise. The strategy of these approaches is to train an NN to recognize different damage scenarios from the measured response of the system. In these approaches, the selection of damage parameters, damage scenarios and the adjustment of the numerical model to the physical system are prerequisites for success. Wu *et al* (1992) used the pattern matching capability of a neural network to recognize the location and the extent of individual member damage from the measured frequency spectrum of the damaged structure. Tsou & Shen (1994) used the change of its dynamic properties (eigenvalues and mode shapes) to find damage using a backward-propagation neural network. They carried out their experiment on 3-DOF spring-mass-damper system. Barai & Pandey (1995a) applied neural network based damage detection on bridge truss configuration after carrying out for both static and dynamic analysis of structure. They have also studied (Barai & Pandey 1995b) the performance of the generalized delta rule in the context of multilayer perception simulation of damage identification in truss structures considering strain as structural input parameter. Issues related to the performance of the network with reference to hidden layers and neurons were examined and suggested to choose more than one hidden layer, provided the number of connections is not constant. Nakamura *et al* (1998) has proposed a nonparametric method for damage detection in a building damaged during the Hyogo-Ken Nambu Earthquake of 17 January 1995. Wahab & Roeck (1999) used curvature damage factor to determine damage. They showed that curvature damage factor shows more clear peaks at damage locations than curvature mode shape. Cerri & Vestroni (2000) addressed the problem of identifying structural damage affecting one zone of a beam using measured frequencies. The beam model has a zone in which the stiffness is lower than the undamaged value. Chinchalkar (2001) described a numerical method for determining the location of a crack in a beam of varying depth when the lowest three natural frequencies of the cracked beam are known. The author modelled the crack as a rotational spring and plotted graphs of spring stiffness versus crack location for each natural frequency. The point of intersection of the three curves gives the location of the crack. Morassi (2001) presented a method, which deals with the identification of a single crack in a vibrating rod based on the knowledge of the damage-induced shifts in a pair of natural frequencies. Yongyong *et al* (2001) presented a genetic algorithm-based shaft crack detection technique based on the finite element method. However, the authors suggested that the finite element model of the shaft crack needs to be studied deeply, so that the model simulates the real system more reasonably and higher accuracy of the detection can be expected. Zapico & Gonzalez (2003) have dealt with a methodology based on NNs intended to give overall information about the localization and the amount of damage in the steel-quake structure after seismic loading. The natural frequencies of the structure were used as inputs of the NNs.

A robust damage assessment methodology must be capable of recognizing patterns in the observed response of the structure resulting from individual member damage, including the capability of determining the extent of damage. This capability is within the scope of the pattern matching capabilities of neural networks. The utilization of these capabilities of neural networks in damage assessment is the basis of the study described here. In the present study, neural networks are used to extract and store the knowledge of the patterns, which is the response of the undamaged and damaged structure. Thus the need for construction of mathematical models and comprehensive inverse search is avoided.

The objective of the present paper is to locate and assess the damage occurring at any position in a cantilever beam by back-propagation neural network considering displacement and strain as input parameter to the network. The approach here consists of three sub-processes. First, by varying the model parameters of the structure, their corresponding response to the system is calculated through the finite element method. Second, a neural network is iteratively trained using a number of training patterns. Here, structural responses are given as input to the neural network, while parameters to be identified are shown to the network as desired data. Finally some structural responses measured are given to the well-trained network, which immediately outputs the appropriate value of parameters for untrained patterns. The model parameter taken here is the EI value of the structural member and the structural responses are displacement and strain for a comparison of the performance of the damage assessment algorithm.

2. Theoretical formulation

2.1 *Finite element formulation of structure*

The cantilever beam is idealized here considering the 2-dimensional plane stress formulation. The beam is modelled with 8-noded isoparametric element. The stiffness matrix for the present case may be written as

$$[K] = b \iint_A [B]^T [D][B] \mathrm{d}x \mathrm{d}y, \tag{1}$$

where $[K]$ is assembling stiffness matrix of structure, b is the width of the beam, $[B]$ is the strain-displacement relationship matrix and $[D]$ is the constitutive matrix of the structure. The nodal displacement due to applied load may be calculated by the following equation

$$[F] = [K]\{d\}, \tag{2}$$

where $\{d\}$ is the nodal displacement and $\{F\}$ is the applied load in the node.

Strain may be calculated from the strain–displacement relationship

$$\{\varepsilon\} = [B]\{d\}. \tag{3}$$

2.2 *Brief introduction to neural networks*

An artificial neural network is a framework consisting of many neuron-like processing units. Each neuron is simulated by the sum of the incoming weighted signals and transmits the activated response to the other connected neuron units. Such a network represents an efficient and parallel computational entity and reflects the level of simulations by different input signals.

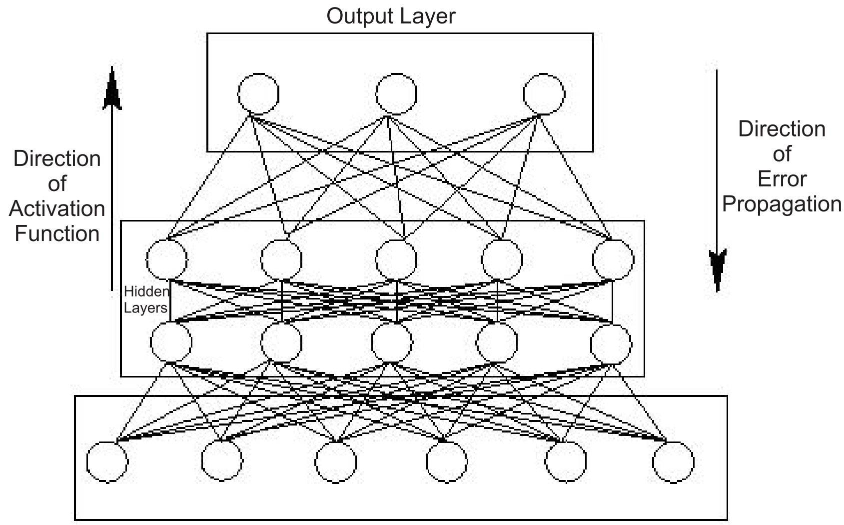

Figure 1. A three-layer neural network.

The dynamic weights which connect neurons of different layers are continuously modified during the process of learning. Rumelhart *et al* (1986) provided an excellent algorithm that allows the multilayer neural network to internally organize itself so to be able to reconstruct the presented patterns. This method leads to the recent very popular neural network learning scheme called the back-propagation algorithm. A typical architecture of a multilayer neural network is shown in figure 1. The input layer receives input patterns; it usually does not have processing units in this layer but simply transmits the signal to the next layer. The hidden layer or layers, residing between input layer and the output layer, consist of a certain number of processing units. Each node in the preceding layer is fully connected to all processing units, and the connections are called the weights that represent different weighting scales to the input signals. The processing unit sums up the weighted signals and activates a response transmitting to the next layer. The activation function may be a monotonically increasing nonlinear (or linear) function. In this study, a nonlinear sigmoidal activation function is used. The input pattern is propagated forward, and calculated responses are obtained. The difference between the desired outputs and the calculated outputs is then propagated backward through the network, providing vital information for weight adaptation. The back-propagation algorithm uses this information to adjust the weights such that the "mean-square" error measure is minimized. This supervised learning algorithm, using gradient descent optimization scheme, helps the network converge to a minimum in the weight space and completes the learning process.

Preparing a good set of training pattern is very important for the neural network learning process. The data generated for the training must be properly sampled and should contain all range of data within the domain. However, it is difficult to get the generalized data set for training and also higher oscillation in the data may make it difficult to reach global minima. On the other hand, the network with more noisy data, if trained properly can predict this in a better way. The number of hidden layers and the number of nodes in the hidden layer is rather problem-dependent and is an open topic for research in getting the optimal number. There is no convincing evidence found that the network with one hidden layer performs badly compared to networks with two or more hidden layers. But by increasing the number of hidden layers the CPU time increases considerably. Therefore single hidden layer network has been used. In this paper, a three-layer back-propagation neural network (figure 1) is developed for damage detection.

3. Numerical examples and results

The computer codes developed in the present study with formulation outlined are applied on a simple cantilever beam structure. The calculated static displacement and strain at some nodal points are used for training the neural network. Element damage is defined as a reduction of EI value of the element. In the present study damage at single location and multiple locations in structure are determined.

A cantilever beam with rectangular cross-section subjected to an end moment, which is replaced by a couple of forces is considered in the present analysis. The dimensions of the beam are shown in figure 2. Displacements at node numbers 5, 13, 21, 29 and 37 are calculated and used as neural network input. The values of input and output are both normalized between 0·1 and 0·9 using the following equation

$$y = 0{\cdot}1 + 0{\cdot}8(x - x_{\min})/(x_{\max} - x_{\min}), \tag{4}$$

where y is the normalized value of input/output, x is the original value of input/output, and $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of a particular data set respectively.
Single element damage: In this category, damage in the element is simulated by reducing EI values from 5 to 95% at 5% intervals. Mean square error (MSE) for training samples are taken as 0·001 after considering convergence and accuracy of training.
Multiple element damage: For this case more than one element in the structural member is considered to be damaged. Two elements, i.e., element numbers 5 and 8 are considered damaged in the present study. By varying EI values from 10 to 90% with 10% interval total 81 number of samples are generated in which 56 samples are used for training and 25 samples for testing in the present analysis. The MSE chosen for this case is 0·005.

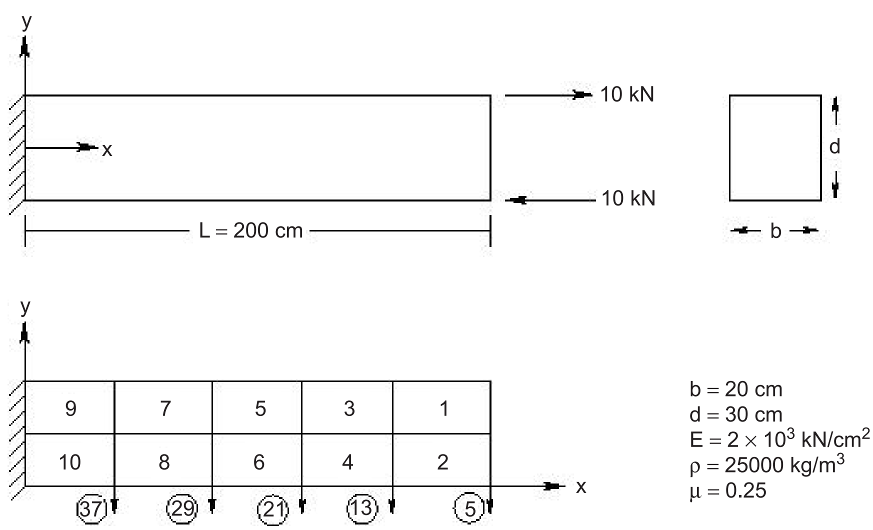

Figure 2. Cantilever beam problem.

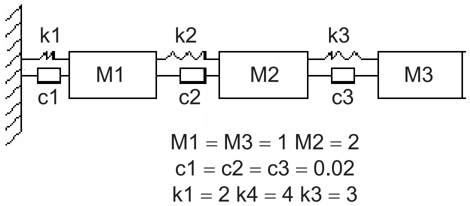

Figure 3. A 3-DOF spring-mass-damper system.

3.1 *Validation of the proposed algorithm*

The code developed for single hidden layer back-propagation neural network training is compared with the results obtained by Tsou & Shen (1994) on a 3-DOF spring–damper–mass system. The 3-DOF spring-mass-damper system is shown in figure 3. Results obtained by Tsou & Shen (1994) and proposed network are shown in table 1 for comparison, where ΔK_1, ΔK_2 and ΔK_3 are change in spring stiffness. The comparison shown in table 1 validates the present algorithm.

3.2 *Displacement as neural network input*

After training for different values of nodes on hidden layer, learning rate (η) and momentum coefficient (α), some optimum values for training and testing error with number of iterations for both the cases are given in table 2. The variation of training and testing error with number of iterations for single and multiple damage cases are shown in figures 4 and 5. It is observed from figure 4, which represents the single damage case, that the difference of training and testing error becomes negligible after 3000 iterations, where the performance is well enough. Also, the errors do not reduce significantly after this number (3000) of iterations. It is interesting to note that the error gets reduced significantly after a few iterations (500 in this case) as the weights are randomly initialized. However, it is observed that the performance at that point is very poor. For the multiple damage case, the results plotted in figure 5 show that after 2000 iterations, the errors do not reduce significantly. The comparison of desired output to neural network output is shown in the form of bar charts from figures 6 to 8. Performance of the network for the single element damage case is shown in figure 6. It is important to note that the largest discrepancies in testing pattern occur at 5% and 95% damage cases. This is because these testing patterns are beyond the training representative range which is from 10 to 90% in these cases. However, the estimated results are still in good range. Figure 7 represents the performance of the network for the multiple damage case with damage in element number 5,

Table 1. Comparison of neural network performance.

Damage (%)	Spring damage by Tsou & Shen (1994)			Spring damage by proposed network		
	ΔK_1	ΔK_2	ΔK_3	ΔK_1	ΔK_2	ΔK_3
20	19·59	19·62	19·57	20·48	18·86	19·35
25	24·81	24·80	24·76	25·45	23·55	24·60
45	45·17	45·17	45·26	44·81	45·25	46·30
65	64·75	64·77	64·75	65·80	66·98	66·01
95	93·21	93·37	93·26	90·93	90·34	96·06

Table 2. Training and testing error for different nodes, with different values of η and α (displacement as neural network input).

Problem type	No. of nodes on hidden layer	η	α	No. of iterations	Mean square training error	Mean square testing error
Single	3	0·6	0·4	8843	0·0010018	0·0025624
element	3	0·5	0·7	4371	0·0010014	0·0023924
damage	4	0·6	0·4	9024	0·0010013	0·0030669
	4	0·5	0·7	6869	0·0010020	0·0027212
Multiple	4	0·3	0·4	4090	0·0050241	0·0082753
element	4	0·5	0·5	2479	0·0050241	0·0100939
damage	5	0·3	0·4	3987	0·0050242	0·0070288
	5	0·5	0·5	4251	0·0050243	0·0085932

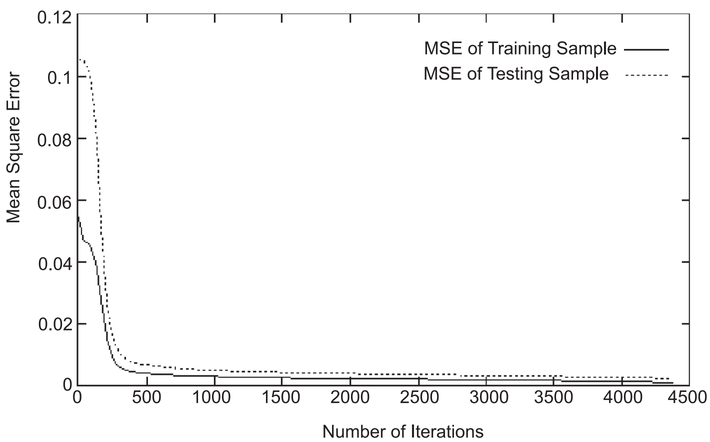

Figure 4. Variation of training and testing error with number of iterations (single damage case).

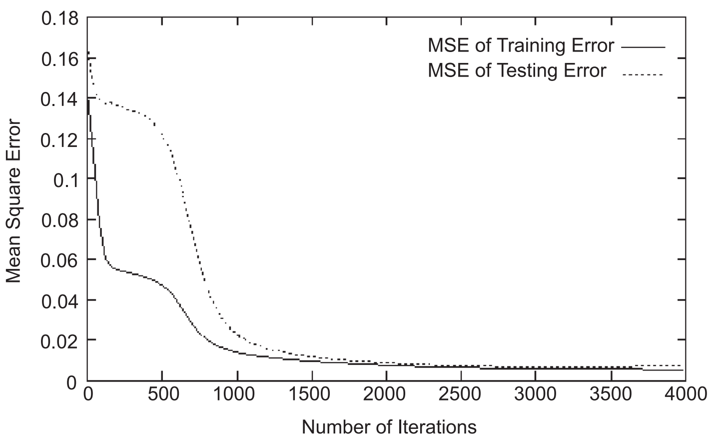

Figure 5. Variation of training and testing error with number of iterations (multiple damage case).

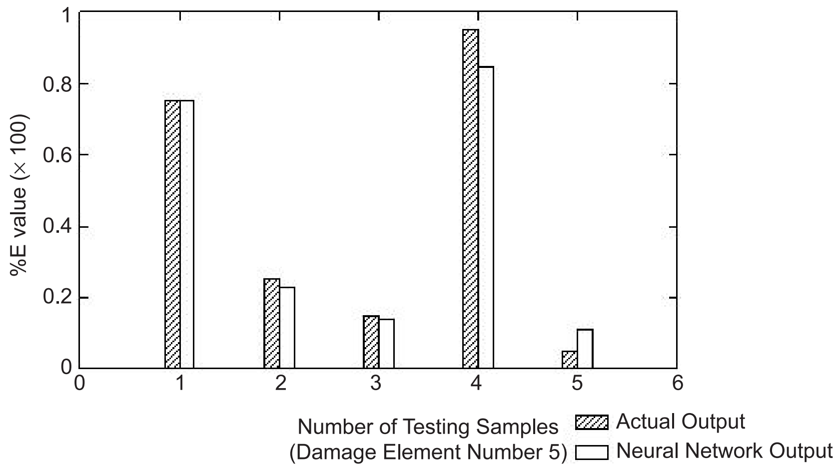

Figure 6. Comparison of actual output and neural network output (single damage case).

whereas, figure 8 represents the same with damage in element number 8. It is understood from bar chart that the results obtained from neural network training and the actual values are quite close which indicates that the performance of the proposed network is quite satisfactory.

3.3 *Strain as neural network input*

Strains are calculated at the same nodal points to that of displacement. Thus in this case also, neural networks inputs have five nodes consisting of strain values. The input and output values are normalized from 0·1 to 0·9 using (4) as mentioned earlier. Two categories of problems, i.e., single element and multiple element damage problems, are solved. The MSE chosen for both the cases are the same as those considered in the previous case. After training for different values of nodes on hidden layer, learning rate (η) and momentum coefficient (α),

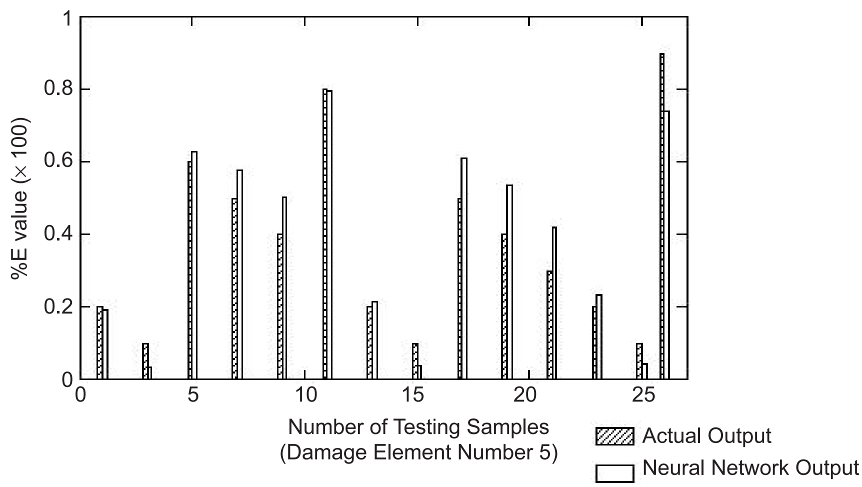

Figure 7. Comparison of actual output and neural network output (multiple damage case).

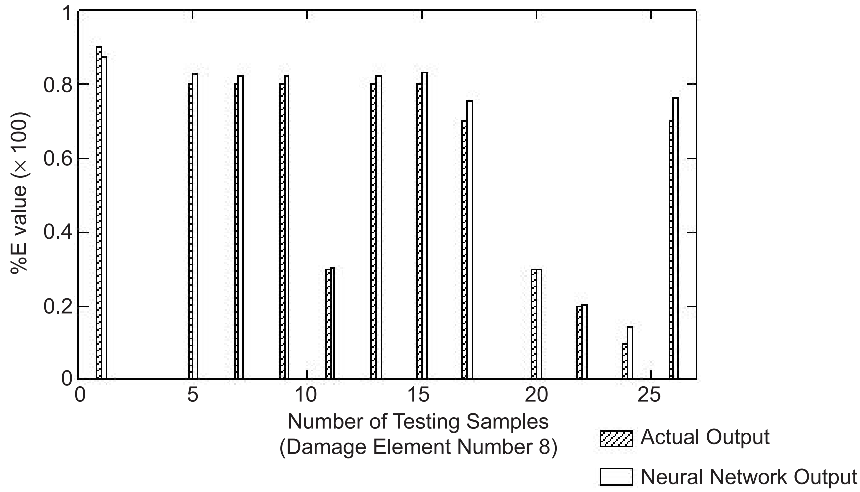

Figure 8. Comparison of actual output and neural network output (multiple damage case).

some optimum values for training and testing error with number of iterations for both the cases are given in table 3. The results show that the desired error level (0·001) for single damage case may be reached after 2459 iterations. The variation of testing and training error with number of iterations for single and multiple damage cases are shown in figures 9 and 10. It is observed from figure 9, which represents the single damage case, that the difference of training and testing error becomes negligible after 1500 iterations. It is interesting to note that the number of iterations and of testing errors become less when strain is taken as neural network input compared to displacement input. Similarly, from figures 5 and 10, for the multiple damage case, performance of the network is better when strain is considered as input parameter. Comparisons of desired output to neural network output are shown in figures 11 to 13. It is observed from the bar chart that the results obtained from neural network training and the actual values are quite close which reflects that the performance of the proposed network is quite satisfactory.

Table 3. Training and testing error for different nodes, with different values of η and α (strain as neural network input).

Problem type	No. of nodes on hidden layer	η	α	No. of iterations	Mean square training error	Mean square testing error
Single element damage	3	0·6	0·4	2839	0·0010021	0·0020982
	3	0·5	0·7	2459	0·0010022	0·0019259
	4	0·6	0·4	3144	0·0010022	0·0020913
	4	0·5	0·7	2606	0·0010021	0·0019288
Multiple element damage	4	0·6	0·4	507	0·0050233	0·0069615
	4	0·5	0·5	490	0·0050203	0·0065825
	5	0·6	0·4	658	0·0050239	0·0065120
	5	0·5	0·5	745	0·0050226	0·0063706

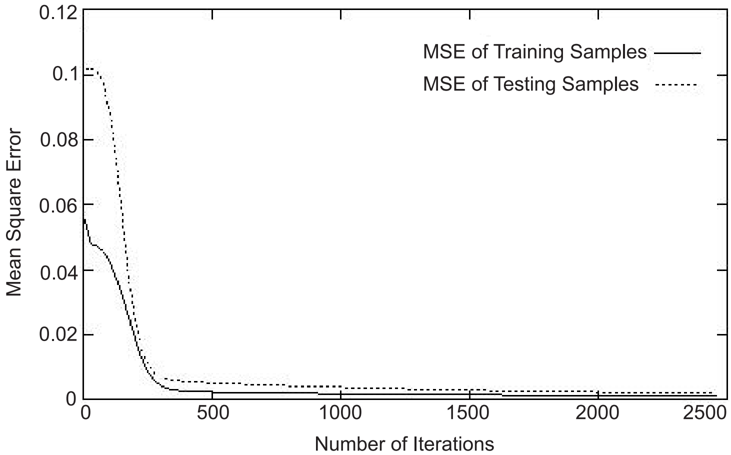

Figure 9. Variation of training and testing error with number of iterations (single damage case).

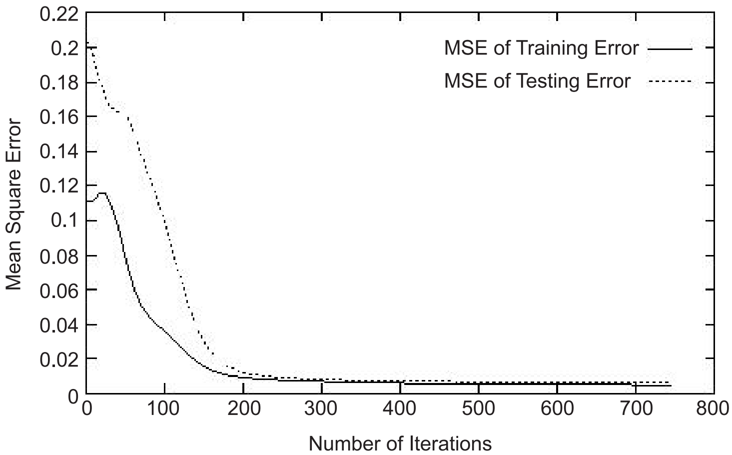

Figure 10. Variation of training and testing error with number of iterations (multiple element damage).

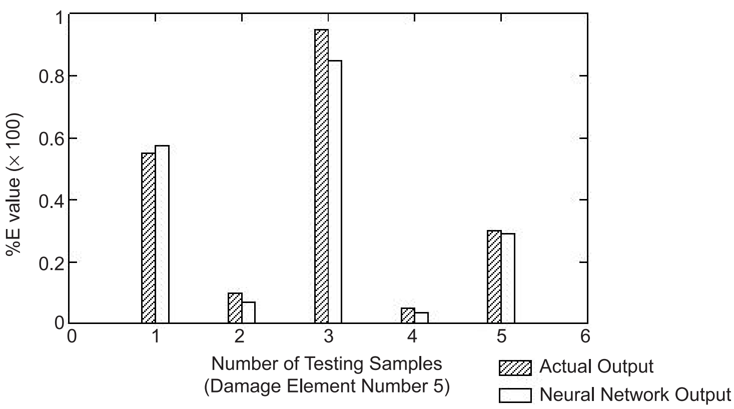

Figure 11. Comparison of actual output and neural network output (single damage case).

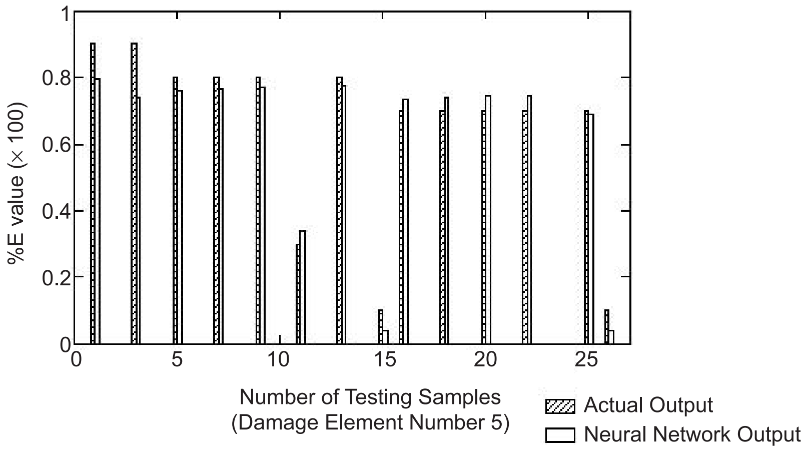

Figure 12. Comparison of actual output and neural network output (multiple damage case).

4. Conclusion

The primary objective of the present investigation is to determine the location and amount of damage of a beam member by a neural network based technique. With the above view, a computer code is developed in which structural response due to damage is carried out. The response data are fed into the network to determine the damage. It is observed that neural networks can successfully identify and calculate the amount of damage for both single and multiple element damage cases. The main advantage of a neural network is that response measurement is required only at a limited number of points. This makes the technique more practical oriented. It is clearly observed from the result that selection of network architecture is of paramount importance in the accuracy of method. The network trains better by producing less MSE of testing sample for some particular values of α and β, and a certain number of

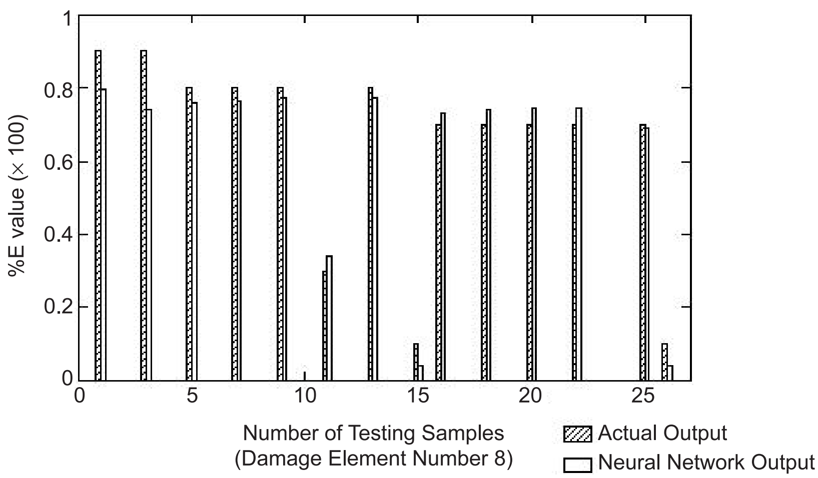

Figure 13. Comparison of actual output and neural network output (multiple damage case).

nodes on hidden layers. The performance of the network is poor in some testing patterns because of the lack of generalization of the training patterns. The network topology and the learning parameters are also problem dependent and lot of uncertainty is associated with it. However, it is an open area of research to get optimum value of the parameters and topology. It is observed from the output results that the network performance improves when strain is used as input pattern instead of displacement. In almost all the cases strain as input gives less training and testing error than displacement.

References

Barai S V, Pandey P C 1995a Multilayer Perceptron in Damage Detection of Bridge Structures. *Comput. Struct.* 54: 597–608

Barai S V, Pandey P C 1995b Performance of generalized delta rule in artificial neural network for damage detection. *Eng. Appl. Artif. Intell.* 211–221

Bishop C M 1998 *Neural networks for pattern recognition* (Oxford: Clarendon)

Cawley P, Adams R D 1979 The location of defects in structures from measurement of natural frequency. *J. Strain Anal.* 14: 49–57

Cerri M N, Vestroni F 2000 Detection of damage in beams subjected to diffused cracking. *J. Sound Vibr.* 234: 259–276

Chinchalkar S 2001 Determination of crack location in beams using natural frequencies. *J. Sound Vibr.* 247: 417–429

Friswell M I, Mottershea J E 1999 Finite element updating in structural dynamics. (Dordrecht: Kluwer Academic)

Morassi A 2001 Identification of a crack in a rod based on changes in a pair of natural frequencies. *J. Sound Vibr.* 242: 577–596

Nakamura M, Masari S F, Chassiakos A G, Caughey T K 1998 A method for non-parametric damage detection through the use of neural network. *Earthquake Eng. Struct. Dyn.* 27: 997–1010

Pandey A K, Biswas M 1994 Damage detection in structures using changes in flexibility. *J. Sound Vibr.* 169: 3–17

Pandey A K, Biswas M, Samman M M 1991 Damage detection from changes in curvature mode shapes. *J. Sound Vibr.* 145: 321–332

Rumelhart D E, Hinton G E, Williams R J 1986 Learning internal representations by error backpropagation. *Parallel Distrib. Process. Explor. Microstruct. Cogn.* 1: 318–362

Rytter A, Kirkegaard P H 1997 Vibration based inspection using neural networks. *Proc. DAMAS'97*, University of Sheffield, UK, pp. 97–108

Sanayei M, Onipede O 1991 Damage assessment of structures using static test data. *AIAA J.* 29: 1174–1179

Suh M W, Shim M B, Kim M Y 2000 Crack identification using hybrid neuro-genetic technique. *J. Sound Vibr.* 238: 617–635

Szewczyk Z P, Hajela P 1994 Damage detection in structures based on feature sensitive neural networks. *ASCE J. Comput. Civil Eng.* 8: 163–178

Tsou P, Shen M H 1994 Structural damage detection and identification using neural networks. *AIAA J.* 32: 176–183

Wahab M M A, Roeck G D 1999 Damage detection in bridges using modal curvatures: Application to a real damage scenario. *J. Sound Vibr.* 226: 217–235

Worden K 1996 Multi-layer perceptron (MLP), Version 3·4, A users guide. Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, UK

Wu X, Ghaboussi J, Garrett J H 1992 Use of neural networks in detection of structural damage. *Comput. Struct.* 42: 649–659

Yongyong H, Dan G, Fulei C 2001 Using genetic algorithm and finite element methods to detect shaft crack for rotor-bearing system. *Math. Comput. Simul.* 57: 95–108

Yun C, Bahng E Y 2000 Substructural identification using neural network. *Comput. Struct.* 77: 41–52

Zapico J L, Gonzalez M P 2003 Damage assessment using neural networks. *Mech. Syst. Signal Process.* 17: 119–125