Damage assessment of structures from changes in curvature damage factor using artificial neural network

Rashmi Ranjan Tripathy & Damodar Maity*
Department of Civil Engineering, Indian Institute of Technology Guwahati, 781 039, India

Received 17 February 2004; accepted 20 May 2004

This paper presents a neural network based approach to detect and assess the structural damage. The basic strategy applied in this study is to train a neural network to recognize the behaviour of the undamaged structure as well as the structure with various possible damaged states. Curvature damage factor (CDF) is used as a possible candidate for the damage identification by error back-propagation training algorithm (EBPTA). 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 cantilever beam and a plane frame. The results show the efficiency of the developed algorithm.

Damage to structure may be caused as a result of accidents, deterioration or severe natural event such as earthquake and storms. Sometimes the extent and location of damage can be determined through visual inspection. But visual inspection technique has a limited capability to detect damage, especially when damage lies inside the structure and is not visible. So an effective and reliable damage assessment methodology will be a valuable tool in timely determination of damage and deterioration state of structural member. The information produced by a damage assessment process can play a vital role in the development of economical repair and retrofit programs of existing civil engineering structures.

Conventional damage assessment methods proposed in the literature mostly follow more or less the same approach. First, a mathematical model for the structure is constructed. The mathematical model is then used to develop an understanding of 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 parameters from the response of the damaged structure is an inverse process; where causes must be discerned from effects.

In recent years a significant amount of effort has been devoted to damage assessment schemes based on analysis of measured responses of the structure before and after damage. Sanayei and Onipede\textsuperscript{1} used static displacement to find damage under applied loads. In a truss model damage is introduced by reduction of cross-sectional area. Pandey et al.\textsuperscript{2} utilized curvature damage factor to find damage of cantilever and simply supported beam structure. Wu et al.\textsuperscript{3} 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 and Shen\textsuperscript{4} used the change of its dynamic properties (eigenvalues and mode shapes) to find damage using a backward-propagation neural network. They carried out experiment on 3-DOF spring-mass-damper system. Barai and Pandey\textsuperscript{5} applied neural network based damage detection on bridge truss configuration after carrying out both static and dynamic analysis of structure. Wahab and Roeck\textsuperscript{6} showed that curvature damage factor shows more clear peaks at damage locations than curvature mode shape. Shih and Kao\textsuperscript{7} used the optimal weights of the approximating artificial neural networks for damage detection of linear structures. They presented SDOF and MDOF examples to demonstrate the feasibility.

It is observed from literature that the damage may be detected in a better way when curvature damage factor is considered as an input of the structural response. But this is not sufficient in case of multi-
damage large structure. However, artificial neural network (ANN) can effectively deal with qualitative, uncertain, and incomplete information, thereby making it highly promising for detecting structural damage, as most of the in situ measured data of large civil engineering structures such as buildings and bridges are not precise and often incomplete. 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 location and extent of damage. This capability is within the scope of the pattern matching capabilities of ANNs. The utilization of these capabilities in damage assessment is the basis of study described here. In the present study, ANNs 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 in a cantilever beam and plane frame by back-propagation neural network considering curvature damage factor as input parameter to the network. The approach here consists of three sub processes. First, by varying the model parameters of the structure, the corresponding responses for the system are obtained using the finite element analysis. Secondly, the 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 appropriate value of parameters for untrained patterns. The model parameter taken here is $EI$ value of the structural member and structural responses are curvature damage factor, which is obtained from displacement mode shape.

**Neural Network**

ANNs form a class of systems that are derived from biological neural networks. ANN is a framework consisting of many number of 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 will reflect the level of simulations by different input signals. The dynamic weights, which connect neurons of different layers, are continuously modified during the process of learning. The back-propagation algorithm uses this information to adjust the weights such that a "mean-square" error is minimized. Rumelhart et al.\textsuperscript{8} provided an excellent algorithm that allows the multi layer neural network to internally organize itself to be able to reconstruct the presented patterns. This method leads to the recent most popular neural network-learning scheme called the Error Back Propagation Training Algorithm. A typical architecture of a multi layer neural network is shown in Fig. 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 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 scale 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 monotonically increasing non-linear (or linear) function. In this study non-linear sigmoidal activation function is used. The input pattern is propagated forward, and calculated responses are obtained. The errors between desired outputs and the calculated outputs are then propagated backward through the network, providing vital information for weight adaptation. The back-propagated algorithm uses this information to adjust the weights such that a "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. In this paper a three-layer back-
propagation neural network is developed for damage detection purpose.

The backpropagation algorithm is generally employed for its “supervised” learning, to reproduce the output of given input-output training sets within a required error tolerance. The following training error is defined:

$$E = \sum_{p=1}^{n} E_p = \sum_{p=1}^{n} \sum_{k=1}^{m} \left( T_{pk} - O_{pk} \right)^2$$  \hspace{1cm} (1)

where $E_p$ is the square of the error for the $P_{th}$ training pattern. $T_{pk}$ is the teacher signal to the $k_{th}$ unit in the output layer for the $P_{th}$ training. $O_{pk}$ is the output signal from the $k_{th}$ unit in the output layer for the $P_{th}$ training pattern. In the training process, the connection weight $w_{ji}$ is modified repeatedly based on the steepest descent method in order to minimize the above square error:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}}.$$ \hspace{1cm} (2)

where $\eta$ is the learning rate constant. The training is sensitive to the choice of the various net learning parameters. The first parameter is the “learning rate” which essentially governs the step size, a concept familiar to the optimization community, and the learning rate constant can be updated according to the following rule:

$$\Delta \eta = \begin{cases} +a & \text{if } \Delta E > 0 \\ -b \eta & \text{if } \Delta E < 0 \\ 0 & \text{otherwise} \end{cases}.$$ \hspace{1cm} (3)

This learning rate approach is an adaptive learning constant through which the value of learning constant $\eta$ is updating in each iteration according to the change in error $\Delta E$. This scheme is implemented by giving a contribution from the previous time step to catch weight change:

$$\Delta w(n) = -\eta \nabla E(n) + \alpha \Delta w(n-1)$$ \hspace{1cm} (4)

where $\alpha e[0 \ 1]$ is a momentum parameter. The momentum term typically helps to speed up the convergence and to achieve an efficient and more reliable learning profile. The second parameter $\alpha$ is the “momentum coefficient” which forces the search to continue in the same direction so as to aid the numerical stability, and furthermore, to go over local minima encountered in the search. The above algorithm may be found in details in Suh et al.9.

### Problem Formulation

#### Cantilever beam

The cantilever beam is idealized here considering 2-dimensional plane stress formulation. The beam is modelled with 8-noded isoparametric element. The stiffness matrix and the mass matrix for the present case are obtained from the following equations:

$$[K] = b \int_A [B]^T [D][B] dx dy$$ \hspace{1cm} (5)

$$[M] = b \int_A [N]^T \rho[N] dx dy$$ \hspace{1cm} (6)

where $[K]$ and $[M]$ are the stiffness and mass matrix of the beam respectively, $b$ is the width of the beam, $[B]$ is the strain-displacement relationship matrix, $[D]$ is the constitutive matrix and $[N]$ is the shape function.

#### Plane frame

Here the finite element concepts are used to formulate the displacement method of analysis treating the ‘member’ of a framed structure as an ‘element’. A planar frame structure is modelled using two-dimensional beam elements having three degrees of freedom ($\delta x, \delta y, \theta$) per node. The corresponding element stiffness and mass matrices are

$$[K_e] = \frac{EI}{L^3}\begin{bmatrix} \beta l^2 & 0 & 0 & -\beta l^2 & 0 & 0 \\ 0 & 12 & 6L & 0 & -12 & 6L \\ 0 & 6L & 4l^2 & 0 & -6L & 2L^2 \\ -\beta l^2 & 0 & 0 & \beta L^2 & 0 & 0 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ 0 & 6L & 2L^2 & 0 & -6L & 4L^2 \end{bmatrix}$$ \hspace{1cm} (7)

$$[M_e] = \frac{\rho AL}{420}\begin{bmatrix} 140 & 0 & 0 & 70 & 0 & 0 \\ 0 & 156 & 22L & 0 & 54 & -13L \\ 0 & 22L & 4L^2 & 0 & 13L & -3L^2 \\ 70 & 0 & 0 & 140 & 0 & 0 \\ 0 & 54 & 13L & 0 & 156 & -22L \\ 0 & -13L & -3L^2 & 0 & -22L & 4L^2 \end{bmatrix}$$ \hspace{1cm} (8)
where $\beta = A/I_{zz}$, $L$ is the length of element $e$, and $A$ is the cross-sectional area. $E$ and $\rho$ are the modulus of elasticity and mass density, respectively, and $I_{zz}$ is the second moment of area about the local $z$-axis.

**Calculation of curvature damage factor**

After calculating stiffness and mass matrix, the eigen value problem for free vibration case may be written as

$$(K-\lambda_j M)v_j = 0 \quad \ldots (9)$$

From the displacement mode shape, the curvature mode shape may be obtained by central difference approximation as

$$v_i^c = (v_{i+1} - 2v_i + v_{i-1})/h^2 \quad \ldots (10)$$

where $h$ is the length of the element. Curvature damage factor is defined as

$$CDF = 1/N \sum_{i=1}^{N} |v_{oi}^\prime\prime - v_{di}^\prime\prime| \quad \ldots (11)$$

$N$ is the total number of modes considered, $v_{oi}^\prime\prime$ is the double derivative of curvature mode shape of the intact structure at $i^{th}$ node and $v_{di}^\prime\prime$ is that of the damaged structure at $i^{th}$ node.

**Numerical Examples and Results**

In the present study formulation outlined before with the computer codes developed are applied on different types of structural system such as on a cantilever beam and a plane frame. The calculated curvature damage factors at some nodal points are used for training purpose of the 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 found out.

**Example 1: Validation of the neural network performance**

The code developed for single hidden layer back-propagation neural network training is compared with the results obtained by Tsou and Shen\(^4\) on a 3-DOF spring-damper-mass system. The 3-DOF spring-mass-damper system is shown in Fig. 2. Results obtained by Tsou and Shen\(^4\) and proposed network are shown in Table 1 for comparison where $\Delta K_1$, $\Delta K_2$ and $\Delta K_3$ are the changes in spring stiffness. Comparison shown in Table 1 validates the present algorithm.

**Example 2: Damage on a cantilever beam**

A cantilever beam with rectangular cross-section is considered in the present analysis. The dimensions of the beam are shown in Fig. 3. Curvature damage factors at node number 5, 13, 21, 29 and 37 are calculated and used as neural network input. The values of input and output both are normalized between 0.1 and 0.9 using the following equation

$$y = 0.1 + 0.8 \left( \frac{x - x_{\min}}{x_{\max} - x_{\min}} \right) \quad \ldots (11)$$

here, $y$ is the normalized value of input/output, $x$ is the original value of input/output, $x_{\max}$ and $x_{\min}$ are maximum and minimum values of a particular data set respectively.

![Fig. 2—A 3-DOF spring-mass-damper system](image)

**Table 1—Comparison of neural network performance**

<table>
<thead>
<tr>
<th>Damage (%)</th>
<th>$\Delta K_1$</th>
<th>$\Delta K_2$</th>
<th>$\Delta K_3$</th>
<th>$\Delta K_1$</th>
<th>$\Delta K_2$</th>
<th>$\Delta K_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>24.81</td>
<td>24.80</td>
<td>24.76</td>
<td>25.45</td>
<td>23.55</td>
<td>24.60</td>
</tr>
<tr>
<td>45</td>
<td>45.17</td>
<td>45.17</td>
<td>45.26</td>
<td>44.81</td>
<td>45.25</td>
<td>46.30</td>
</tr>
<tr>
<td>65</td>
<td>64.75</td>
<td>64.77</td>
<td>64.75</td>
<td>65.80</td>
<td>66.98</td>
<td>66.01</td>
</tr>
<tr>
<td>95</td>
<td>93.21</td>
<td>93.37</td>
<td>93.26</td>
<td>90.93</td>
<td>90.34</td>
<td>96.06</td>
</tr>
</tbody>
</table>
Case - I: Single element damage

In this category, damage in a single element, i.e., in element number 8 is simulated by reducing $EI$ value from 5 to 95% with 5% interval. Here 19 numbers of samples are generated in which 11 samples are used for training and 8 for testing. MSE (mean square error) for training samples are taken as 0.001 after considering convergence and accuracy of training by changing the number of nodes on hidden layers, learning rate and momentum coefficient. Different values of the mean square testing error are obtained and are given in tabular form. It is clear from Table 2 that for hidden layer of 6 nodes with learning rate constant 0.5 and moment coefficient 0.5 gives the best result after 8389 number of iterations. It is observed from Fig. 4, which represents the single element damage case, that the difference of training and testing error becomes almost constant after 2000 iterations. Also, the errors do not reduce significantly after this number of iterations. The actual output and neural network output are shown in Fig. 5, which shows a good agreement of the results. The maximum percentage of testing error in this case is 13.33.

Case - II: Damage in two elements

For this case more than one element in the structural member is considered damaged. Two elements, i.e., element numbers 8 and 10 are considered damaged. By varying $EI$ value from 10 to 90% with 10% interval, total 81 number of samples are generated in which 55 samples are used for training and 26 for testing in the present analysis. The MSE chosen for this case is also 0.001. By changing the number of node on hidden layer, learning rate and momentum coefficient; different values of mean square testing error are obtained. It is observed from
Table 2 that for hidden layer of 6 nodes with learning rate 0.4 and moment coefficient 0.5 gives the best result after 48543 number of iterations. From Fig. 6 it is clear that the difference between training and testing error becomes negligible after 20,000 number of iterations. The comparison between the actual output and neural network output is shown in Fig. 7 for element number 8. The maximum percentage of testing error in this case is 17.6.

Case - III: Damage in three elements

In this case three elements, i.e., element numbers 6, 7 and 10 are considered damaged. By varying $EI$ value from 10 to 90% with 20% interval, total 125 number of samples are generated in which 90 samples are used for training and 35 for testing in the present analysis. The MSE chosen for this case is also 0.001. By changing the number of nodes on hidden layer, learning rate and momentum coefficient; different values of mean square testing error are obtained. It is observed from Table 2 that for hidden layer of 6 nodes with learning rate 0.4 and moment coefficient 0.5 gives the best result after 48984 number of iteration. From Fig. 8 it is clear that the difference between training and testing error becomes negligible after 10,000 number of iterations. It is observed from the results from Table 2 that for multi damage case the number of iterations required is more than that of single damage case. The comparison between the actual output and neural network output is shown in
Fig. 9 for element number 6. The maximum percentage of testing error in this case is 20.0.

**Example 3: Damage on a plane frame**

**Case - I: Single element damage**

A two-dimensional frame is considered in the present analysis as shown in Fig. 10. In this category, damage in one element, i.e., in element no. 9 is simulated by reducing $EI$ value from 5 to 95% with 5% interval. Thus 81 number of samples are generated, in which 60 samples are used for training and 21 for testing. MSE for training samples is taken as 0.001 after considering convergence and accuracy of training. By changing the number of nodes on hidden layer, learning rate and momentum coefficient; different values of mean square testing error are obtained. Table 3 shows that for hidden layer of 3 nodes with learning rate 0.4 and moment coefficient 0.5 gives the best result after 816 number of iterations. The training and testing error with the number of iterations are shown in Fig. 11. It is clear from Fig. 11 that after 500 number of iterations, the difference between the two errors becomes negligible. The actual output and neural network output are shown in Fig. 12, which shows a good agreement of the results. The maximum percentage of testing error in this case is 20.0.

**Case - II: Damage in two elements**

For the multi element damage case, more than one element in the structural member is considered damaged. Two elements, i.e., element numbers 1 and

---

**Table 3**—The training and testing error for different nodes, with different $\eta$ and $\alpha$ value (plane frame)

<table>
<thead>
<tr>
<th>Problem type</th>
<th>No. of nodes on hidden layer</th>
<th>$\eta$</th>
<th>$\alpha$</th>
<th>No. of iterations</th>
<th>Mean square training error</th>
<th>Mean square testing error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single element damage</td>
<td>3</td>
<td>0.4</td>
<td>0.5</td>
<td>816</td>
<td>0.001</td>
<td>0.00540</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.5</td>
<td>643</td>
<td>0.001</td>
<td>0.00556</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.4</td>
<td>0.5</td>
<td>763</td>
<td>0.001</td>
<td>0.00662</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.5</td>
<td>0.5</td>
<td>619</td>
<td>0.001</td>
<td>0.00671</td>
</tr>
<tr>
<td>Two element damage</td>
<td>5</td>
<td>0.4</td>
<td>0.5</td>
<td>37994</td>
<td>0.001</td>
<td>0.00531</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.5</td>
<td>0.5</td>
<td>33680</td>
<td>0.001</td>
<td>0.00511</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.4</td>
<td>0.5</td>
<td>28312</td>
<td>0.001</td>
<td>0.00819</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.5</td>
<td>0.5</td>
<td>30346</td>
<td>0.001</td>
<td>0.00501</td>
</tr>
<tr>
<td>Three element damage</td>
<td>5</td>
<td>0.4</td>
<td>0.5</td>
<td>47542</td>
<td>0.001</td>
<td>0.00146</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.5</td>
<td>0.5</td>
<td>49584</td>
<td>0.001</td>
<td>0.00187</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.4</td>
<td>0.5</td>
<td>65038</td>
<td>0.001</td>
<td>0.00156</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.5</td>
<td>0.5</td>
<td>73021</td>
<td>0.001</td>
<td>0.00161</td>
</tr>
</tbody>
</table>
9 are considered damaged. By varying $EI$ value from 10 to 90% with 10% interval, total 81 number of samples are generated in which 60 samples are used for training and 21 for testing in the present analysis. The MSE chosen for this case is 0.001. By changing the number of nodes on hidden layer, learning rate and momentum coefficient; different values of mean square testing error are obtained. It is seen from Table 3 that for hidden layer of 6 nodes with learning rate 0.5 and moment coefficient 0.5 gives the best result after 30346 number of iterations. Fig. 13 shows the variation of training and testing error. It is clear that for multi element damage case the number of iterations is more as compared to the single element damage for the case of plane frame. The comparison between the actual output and neural network output is shown in Fig. 14 for member number 1. The maximum percentage of testing error in this case is 9.4.

**Case - III: Damage in three elements**

In this case three elements, i.e., element number 1, 6 and 9 are considered damaged. By varying $EI$ value from 10 to 90% with 20% interval total 125 number of sample are generated in which 90 samples are used for training and 35 for testing in the present analysis. The MSE chosen for this case is also 0.001. By changing the number of nodes on hidden layer, learning rate and momentum coefficient; different values of mean square testing error is obtained. It is observed from Table 3 that for hidden layer of 5 nodes with learning rate 0.4 and moment coefficient 0.5 gives the best result after 47542 number of iterations. From Fig. 15 it is clear that the difference between training and testing error becomes negligible, after 10,000 number of iterations. It is observed from the results from Table 2 that for multi damage case the number of iterations required is more than that of single damage case. The comparison between the

---

**Fig. 11**—Variation of training and testing error with number of iterations (Single element damage)

**Fig. 12**—Comparison of actual output and neural network output (Single element damage)

**Fig. 13**—Variation of training and testing error with number of iterations (Two-element damage)

**Fig. 14**—Comparison of actual output and neural network output (Two-element damage)
The actual output and neural network output is shown in Fig. 16 for element number 1. The maximum percentage of testing error in this case is 23.3.

Conclusions
The primary objective of the present investigation is to find out the location and amount of damage by neural network based technique with the help of curvature damage factor as network input. Keeping this in view, a computer code is developed in which structural response due to damage is obtained. The response data are fed into the network to find out the damage. It is observed that neural network can successfully identify and calculate the amount of damage for both single and multiple element damage cases. The main advantage of the neural network is that the response measurement is required only at limited number of points, thus making the technique more practical. It is clearly observed from the results that selection of network architecture is of paramount importance in the accuracy of the method. Some particular value of $\alpha$ and $\eta$, and certain number of nodes on hidden layer network, trains the network better as it gives less MSE for the testing sample. From the above examples one may conclude that as the number of iterations increases, the difference between training and testing error is reduced and it becomes very small after a certain number of iterations. Thus the trained network may be used effectively for damage detection with certain number of nodes on hidden layer and with specific values of learning rate constant and momentum coefficient as mentioned.

References