ABSTRACT

This paper presents a novel approach to estimate the structural damage location and extent based on combining independent component analysis (ICA) extraction of time domain data and artificial neural networks (ANN). The advantage of using time history measurements is that it provides the original vibration information. However, the volume of data, measurement noise, and the lack of reliable feature extraction tools are the major obstacles. To circumvent them, the independent component analysis technique is applied to represent the 'measured' data with a linear combination of dominant statistical independent components and the mixing matrix $[A]$. Such a representation seems to capture the essential structure of the measured vibration data. The vibration features represented by the mixing matrix provide the relation between the measured vibration response and the independent components and are then employed to build the simplified neural network model for damage detection. To estimate the structural damage location and extent, a combination of neural network models is applied. The simplified neural network model is first used for damage detection and then the second NN model is used for damage location. The models in the third level are further used to assess the extent of the damage once the damage has been located by the second model. A truss structure, with simulated displacement data, was used to demonstrate the approach and the results show the effectiveness of the method.

NOMENCLATURE

$\{S(t)\}$ vector of independent components
$s_i(t)$ the $i$-th independent component
$[A]$ mixing matrix
$\{X\}$ vector of measured time domain data
$[C]$ covariance matrix of time domain data
$[\Sigma]$ diagonal matrix of singular values
$[\Psi]$ the associated singular vector matrix
$\{Y(t)\}$ a vector of time domain data
$[W]$ the filtering matrix
$kurt(y_i(t))$ the Kurtosis of the estimated signal $y_i(t)$

FRF Frequency Response Function
SVD Singular Value Decomposition

1. INTRODUCTION

Structural damage detection based on measured vibration data is becoming increasingly important not only for preventing catastrophic failures but also for uninterrupted operation and prolonged service life. Detailed surveys of damage detection methods are given in [1-3]. Generally speaking, damage detection techniques can be classified according to the type of measured data on which they are based. Modal parameters (natural frequencies and mode shapes), provide a substantial reduction in measured vibration data, and are often used for damage detection [4-7]. However, modal parameters are not always easy to interpret in terms of mathematical modelling of linear vibrating systems and they are also insensitivity to local mass and stiffness changes. Methods based on the measured frequency response functions (FRFs), such as the
FRF curvature method and the FRF quotient method, are given in [8,9]. Most damage detection algorithms dealing with FRFs use a validated FE model as a reference and attempt to find discrepancies between this reference model and the FRFs of some damaged specimen [10-12]. But, practical difficulties still remain in obtaining a reference model for comparison purposes. Alternatively, methods based on the raw time signals are given in [13,14]. An obvious advantage of using raw time domain data is that the original vibration information is captured without any signal processing degradation. For instance, Farrar et al. [15] applied an auto-regressive (AR) model for measured acceleration-time histories and selected the coefficients of the AR model as damage indicators. However, in spite of promising results, there are some major obstacles that remain unresolved, such as the difficulties in dealing with large volumes of data, inherent measurement noise, and lack of reliable feature extraction tools.

In recently years, techniques based on multivariate statistics and neural networks have been applied to structural damage detection [1,16]. Zang & Imregun [17-19] utilized a principal component analysis technique (PCA) to condense the FRF data. The basic idea is to compute the so-called principal components (PC) of a test matrix, the rows of which are the actual measured FRFs. The PC-compressed FRFs are represented by their projections onto the most significant principal components and have the additional benefit of containing less measurement noise. Furthermore they used both supervised and unsupervised ANNs to successfully detect the damage in the case of two representative structures: a railway wheel and space antenna.

Most recently, Zang et al. [20] used the independent component analysis (ICA) technique to extract the dynamic features from measured vibration time histories. The combination of independent component analysis and neural networks for structural damage detection is presented in [21]. This paper can be considered as an extension of the above methodology to further estimate the structural damage location and extent. The ‘measured’ time domain data represented with a linear combination of dominant statistical independent components and the mixing matrix [A] seem to capture the essential characteristics. The additional benefits are not only the ability to deal with the relatively high measurement noise, but also the availability of higher-order statistical data that can be used during damage detection process. The extracted dynamic features, represented by a mixing matrix, are then employed to build a simplified artificial neural network for damage detection. The combination of multiple levels of neural network models is applied not only to detect the structural damage also to estimate its location and extent. The methodology will be applied to a truss structure in order to assess its feasibility.

2. METHODOLOGY

Vibration-based structural damage detection can be considered as a kind of pattern recognition paradigm. It consists of data acquisition, signal processing, feature extraction and data reduction, and identification analysis. Rytter [22] described the structural damage state by 4 levels: existence, location, extent and prediction. The focus of this paper is to detect the damage location and to assess the extent based on feature extraction for time histories from different sensors and a detection technique using multiple neural networks. Figure 1 shows the framework of the damage detection and assessment process.

2.1 Feature Extraction Using the ICA Technique

In recent years, feature extraction methods such as blind signal separation (BSS) via independent component analysis (ICA) have been applied to speech enhancement, telecommunications and medical signal processing [23]. ICA techniques provide statistical signal processing tools for finding a linear co-ordinate system in multivariate data and hence they are well-suited to feature extraction, noise reduction, density estimation and regression [24,25]. A summary of ICA techniques for extracting the dynamic features from measured vibration time histories in [20] will be given here for the sake of completeness.

The zero-mean matrix $X_{N \times T} = \{x_{ik}\}$ ($i = 1,2,\ldots,N; k = 1,2,\ldots,T$) has $N$ rows of measured time histories $\{x'_i\}$ (the mean of the data $\overline{x}_i$ has been subtracted) from $N$ sensors, each with $T$ time points. Each observation $x'_i(\cdot)$ can be considered to be a linear combination of $M$ statistically independent sources, which are the individual elements of the vector $\{S(\cdot)\} = \{s_1(\cdot), s_2(\cdot), \ldots, s_M(\cdot)\}^T$. The sources $s_i(\cdot)$ are called the independent components and have unit variance. The linear relationship between $\{X\}$ and $\{S\}$ is given by

$$X(t)_{N \times 1} = [A]_{N \times M} \{S(t)\}_{M \times 1} = \sum_{i=1}^{M} a_i s_i(t) \tag{1}$$

where $[A]$ is an unknown mixing matrix. The main issue can then be defined as the estimation of the mixing matrix $[A]$ and the realization of the source vector $\{S(t)\}$ using only the measured data vector $\{X\}$.

The ICA algorithms normally find the independent components of a data set by minimizing or maximizing some objective function [26]. Here we will use the fixed-point algorithm in [27,28] because of its suitability for handling raw time-domain data and good convergence properties.

The first step is to perform a prewhitening of the measured data vector $\{X\}$ by linearly transforming to a vector $\{\hat{X}\}$ whose elements are mutually uncorrelated and all have unit
A singular value decomposition (SVD) of the covariance matrix \( C = \{X(t)\}^T \{X(t)\}^T \) yields,
\[
C = [\Psi][\Sigma][\Psi]^T
\]
(2)
where \( [\Sigma] = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n) \) is a diagonal matrix of singular values and \([\Psi]\) is the associated singular vector matrix. Then, the vector \( \{\hat{X}(t)\} \) can be expressed as,
\[
\{\hat{X}(t)\} = [\Sigma]^{-\frac{1}{2}} [\Psi]^T \{X(t)\}.
\]
(3)
A further advantage of using an SVD-based technique is the possibility of noise reduction by discarding singular values smaller than a given threshold.

The second step is to employ the fixed-point algorithm using Kurtosis. Assume that by using a separating matrix \([W]\), the measured data vector \( \{X(t)\} \) can be linearly transformed to a vector \( \{Y(t)\} \), such that all elements \( y_i(t) \) are both mutually uncorrelated and have unit variance. Thus the statistical expectation of the product \( \{Y(t)\}^T \{Y(t)\} \) is given by \( E[\{Y(t)\}^T \{Y(t)\}] = [I] \). In this case, one can write
\[
\{Y(t)\} = [W]\{X(t)\}
\]
(4)
Using equation (1), (3), the ICA problem can be expressed in the modified form,
\[
\{\hat{S}(t)\} = \{Y(t)\} = [W]\{\hat{X}(t)\}
\]
(5)
If we consider only one source signal at a time, the problem of estimating the filter matrix \([W]\) can be somewhat simplified. From equation (5),
\[
\hat{s}_i(t) = y_i(t) = [w_i]^T \{\hat{X}(t)\}
\]
(6)
where \( \{w_i\} \) denotes the \( i \)-th row of \([W]\). Therefore, using the deflation approach, it is possible to estimate \([W]\) on a row-by-row basis, a sequential procedure that implies that each independent component is also estimated separately.

The Kurtosis of the estimated signal \( y_i(t) \) is defined as
\[
kurt(y_i(t)) = E\left[y_i^4(t)\right] - 3\left(E\left[y_i^2(t)\right]\right)^2
\]
(7)
from equation (6), equation (7) can be simplified to,
\[
kurt\left([w_i]^T \{\hat{X}(t)\}\right) = E\left([w_i]^T \{\hat{X}(t)\}\right)^4 - 3\|w_i\|^4
\]
(8)
Using the constraint \( \|w_i\| = 1 \), the \( i \)-th filtering vector \( \{w_i\} \) can be obtained by minimising the kurtosis of \( y_i(t) \) using a gradient descent type algorithm. To estimate \( M \) independent components, the algorithm must be run \( M \) times. To ensure a different independent component is estimated each time, a simple orthogonalising projection must be used inside the loop. This is possible because the rows of the filtering matrix \([W]\) are orthonormal to each other by virtue of having undergone pre-whitening. Thus the independent components can be estimated one by one by projecting the current solution vector \( \{w_i\} \) onto the space orthogonal to the previously found rows of the filtering matrix \([W]\).

Following the estimation of the filtering matrix \([W]\) and the vector of independent components \( \{S\} \), estimation of the mixing matrix \([A]\) can be somewhat simplified by using a vector \( \{\hat{X}\} \) transformed from a preliminary pre-whitening of the response data \( \{X\} \). Re-writing equation (5) gives
\[
\{\hat{S}(t)\} = \{Y(t)\} = [W]\{\hat{X}(t)\}
\]
(9)
\[
= [W][\Sigma]^{-\frac{1}{2}}[\Psi]^T \{X(t)\}
\]
thus, the mixing matrix \([A]\) can be estimated as,
\[
\]
(10)
Therefore, using the ICA algorithm, the time responses from different sensors can be transformed into a linear mixture of higher-order statistically independent components and the original time histories from spatial sensors can be reconstructed by
\[
\hat{x}_i(t) = \sum_{j=1}^{M} a_{ij}s_j(t) + v_i(t);
\]
(11)
\((i = 1, 2, \ldots, N; \; j = 1, 2, \ldots, M)\)

where \( \hat{x}_i(t) \) denotes the reconstructed time history from the \( i \)-th spatial sensor. The matrix \([A]\) represents the relationship between the measured responses (inputs) and the independent components (outputs). In other words, \([A]\) may be viewed as a transformation matrix between the time
Damage cases denoted DS1, DS2, ..., DS9 were then introduced to the truss structure by reducing the stiffness of the 9 axial members by 50% respectively, thus obtaining 81 damaged specimen time histories. Including the healthy state of the truss, there is a total of 90 time histories. Figure 3 shows a time history comparison of damaged (dotted line) and healthy (solid line) specimens at node 5 in the horizontal direction.

2.2 Neural Networks for Damage Detection

Artificial Neural Networks provide a general, non-linear parameterised mapping between a set of inputs and a set of outputs. Once trained on available sample data, they can recognise patterns and hence they are ideally suited to signature analysis. Although many types of ANNs are used in practice, a popular neural network model called a multiplayer perceptron (MLP) or a back-propagation (BP) type network will be used here [29]. The network consists of an input layer, hidden layers, and an output layer. Each layer is composed of variable nodes. The extracted features from ICA techniques are used to define the data for the input nodes, while the output nodes show the state of the structure. The number of nodes in the hidden layers is selected to make the network more efficient and to interpret the data more accurately. The relationship between the input and output can be non-linear or linear, and its characteristics are determined by the weights assigned to the connections between the nodes in the two adjacent layers. Changing the weight will change the input–to-output behaviour of the network.

An ANN analysis consists of two stages, namely training and testing. During the training stage, an input-to-output mapping is determined iteratively using the available sample data. The actual output error, propagated from the current input set, is compared with the target output and the required compensation is transmitted backwards to adjust the node weights so that the error can be reduced at the next iteration. The learning stage is terminated once a pre-set error threshold is reached and the node weights are frozen at this point. During the testing stage, data from specimens with unknown properties are provided as input and the corresponding output is calculated using the fixed node weights.

3 Application of a Truss Structure

3.1 Preliminaries

The truss structure of Figure 2 was used to investigate the combined ICA/ANN damage detection technique. All 9 elements have elastic modulus of 200GPa and a cross-sectional area of $2.5 \times 10^{-3}$ m$^2$. No damping is considered. The horizontal and vertical displacements of Node 1 and the horizontal displacement of Node 6 were constrained. An impact load of 200 N was applied at Node 5 and the 9 resulting displacement time histories at the nodes were computed.

In advancing the simulation towards more realistic data, the 90 time history data were contaminated with $p=\{0\%$, $5\%$, $10\%$, $15\%$, $20\%$, $25\%$, $30\%\}$ noise at each time point with a uniform distribution on the interval $[0,1]$. Therefore, 630 noise-corrupted copies were obtained.

3.2 ICA for Feature Extraction

The 630 time histories, 63x1 corresponding to the healthy state and 63x9 corresponding to the damaged states from DS1 to DS9, were used to form a 630x301 matrix, where 301 is the number of points in each time history. Clearly, such a large data set cannot be used with ANNs and some form reduction, or feature extraction, is necessary. The whitening approach using principal component analysis was employed and the first ten dominant principal components containing 97.84% of the original information were selected for further independent component analysis. Then, the ten independent
component functions and the features based on the mixing matrix \([A]\) (630x10) were obtained.

### 3.3 Neural Network for Damage Detection

After ICA feature extraction, the original time history specimens are transformed to a linear combination of the mixing matrix \([A]\) and the higher order independent components. The mixing matrix \([A]\), which represents the features of the measured data, can then be used as the input data to the neural network model for damage identification. Therefore, the available 630 specimens were first divided into a training group of 540 and a testing group of 90. The training and verification group consisted of 54 'specimens' for each damage state (9 elements in total), again with added noise. The testing group consisted of 90 specimens containing 9 of each state respectively. The output of network consists of 2 nodes, representing the healthy and damage states shown in Table 1.

| Table 1: Output nodes definition for damage detection |
|-----------|-----------|-----------|
| State     | Node1     | Node2     |
| Healthy (HS) | 1         | 0         |
| Damaged (DS1, DS2,..., DS9) | 0         | 1         |

One hidden layer with 5 nodes is chosen for the BP network. Thus, a simple three-layer BP network with 9 input nodes, a 5 node hidden layer, and 2 output nodes was built for further training and testing. During the training and verification period, the learning and momentum rates for network were set up at 0.6 and 0.3 respectively in the early stage but fell to 0.01 and 0.001 later. The network converged smoothly to RMS errors of 1.38% for training and 0.53% for verification in 3000 iterations. This indicates that the neural network is stable and well trained.

After successful training and verification of the network, a set of 90 noise-corrupted data, 9 from the healthy specimen and 81 from the damaged specimen of 9 different elements were selected for testing. All data in each set were fed sequentially into the network and are classified successfully.

### 3.4 Neural Network for Damage Location

As the health and damage specimens can be detected with the above simple neural network, the estimation of damage locations will be studied. Another three-layer ANN model was built with 10 input nodes, an 11 node hidden layer, and 10 output nodes. The output nodes represent the healthy and damage location states are defined in Table 2. The same available 630 specimens were used for training and testing of the network. During the training and testing, the actual output of most of specimens follows the target output closely, except that eighteen of the 540 training samples (3.3%) and three of the 90 testing samples (3.3%) failed to be estimated correctly by the network. All such failed samples were traced to the DS4 output of node 2 at the horizontal direction and DS5 output of node 2 and 3 at the vertical directions. It is likely that these time domain 'measurements' are not very sensitive to the damage even through all of the data were used successfully for damage detection in the previously network. The values of the 10-node output for all of the failed samples were less than 0.1 in the neural network, and so there is clearly no mistake in detecting undamaged locations but the identification did fail.

### 3.5 Neural Network for Damage Assessment

After the structural damage was located, the next step is to estimate the damage extent at the identified locations. Model updating of the structural FE model may be used to assess the damage extent accurately at the known locations. Here, we will use the neural network method to estimate the damage extent. Assume that damage case DS4 was introduced to the truss structure by reducing the stiffness of the 4th axial member. The different damage extents were simulated by reducing the stiffness by 10%, 20%, 30%, 50%, 75%, and 90% respectively. Together with the healthy state of the truss, there are 63 time histories. The 63 time history data were then contaminated by uniformly distributed noise from 0% to 30%, with the increase step of 5% respectively. Thus, 441 samples were obtained.

Repeatably using the ICA technique, the first ten independent components which contains 98.145% of the original information were selected and the corresponding features based on the mixing matrix \([A]\) (441X10) were obtained. The neural network model was built with a 10-node input layer, one 7-node hidden layer, and a 3-node output layer. The output definition is listed in Table 3. A stiffness reduction of less than 50% is defined as light damage, while a stiffness reduction of over 50% is taken to be heavy damage.

| Table 3: Output definition for damage assessment |
|-----------|-----------|-----------|
| State of DS4  | Node1 | Node2 | Node3 |
| Healthy       | 1    | 0    | 0    |
| Light damage (<50%) | 0    | 1    | 0    |
| Heavy damage (≥50%) | 0    | 0    | 1    |

![Figure 4: Network output for Node 1, 2 and 3: Training specimen (Target output: solid line, actual output: asterisk)
The network was trained with 378 specimens and tested with 63. The training output from the network is plotted in Figure 4, where the target and actual outputs are represented by solid and asterisk respectively. It shows that all the actual output closely matched the target output, though some very tiny fluctuations are present in the healthy and light damage states in nodes 1 and 2. The testing output of the network is plotted in Figure 5, compared with the target output. It can be seen that all testing specimens are classified correctly.

Figure 5: Testing output for Node1, 2 and 3. (Target: solid line, actual output: asterisk)

4. CONCLUDING REMARKS

The results indicate that the combination of time domain data processed by independent component analysis and the use of neural networks provides a suitable methodology for damage detection and assessment. Once the features in the data can be extracted, the simplified neural network models can be built for damage identification.

The independent component analysis is a powerful tool for decomposing the signals into uncorrelated independent components. Such a route has the advantage of not only reducing the size of the ‘measured’ data and the noise contaminating in the data, but also provides higher order statistics that could be helpful for damage identification.

Multiple neural networks can be utilized for structural damage detection and assessment. The quality of measured data from sensors has the important influence on the detection, especially on the identification of damage locations. Measurements which are insensitive to the damage locations have difficulty in identifying the damage locations in the neural network model.

The technique has the capability to cope with the situation that only the vibration response can be measured. Such a feature has potential for on-line industrial applications.

REFERENCES


[22] Rytter T. Vibration based inspection of civil engineering structure, PhD dissertation, Department of building technology and structure engineering, Aalborg University, Denmark, 1993.


<table>
<thead>
<tr>
<th>State</th>
<th>Node1</th>
<th>Node2</th>
<th>Node3</th>
<th>Node4</th>
<th>Node5</th>
<th>Node6</th>
<th>Node7</th>
<th>Node8</th>
<th>Node9</th>
<th>Node10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthy (HS)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS1)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS2)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS4)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS6)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS7)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS8)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Damaged (DS9)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Output nodes definition for damage location
Figure 1: Framework of damage detection and assessment process