AIAA JOURNAL
Vol. 41, No. 1, January 2003

# Fault Classification Using Pseudomodal Energies and Neural Networks

Tshilidzi Marwala*
*Dzinet Investment Holding, Sandton 2146, South Africa*

**A new fault identification method is introduced that uses pseudomodal energies to train neural networks. The proposed procedure is tested on a simulated cantilevered beam and a population of 20 cylindrical shells, and its performance is compared to that of the procedure that uses modal properties to train neural networks. Both the cantilevered beam and cylindrical shells are divided into three substructures, and faults are introduced into these substructures. The cylinder is excited using modal hammer, and acceleration is measured using an accelerometer. Each fault case is assigned a fault identity with the presence of fault represented by a 1, whereas the absence of fault is represented by a 0. Following this fault representation scheme, a fault located in substructure 1 would have an identity of [1 0 0], with two zeros indicating the absence of faults in substructures 2 and 3. The neural network used is a multilayer perceptron trained using scaled conjugate method. The statistical overlap factor and principal component analysis are used to reduce the size of the input data. For both examples the pseudomodal-energy-trained neural networks provide better classification of faults than the networks trained using the conventional modal properties.**

## Nomenclature

| | | |
|---|---|---|
| $a_q$ | = | lower frequency bounds for $q$th pseudomodal energy |
| $b_q$ | = | upper frequency bounds for $q$th pseudomodal energy |
| $[C]$ | = | damping matrix |
| d[•] | = | differential operator |
| $E$ | = | objective function |
| $\{F\}$ | = | force input vector |
| $f_{\text{inner}}$, $f_{\text{outer}}$ | = | hidden and output activation function |
| $j$ | = | $\sqrt{-1}$ |
| $[K]$ | = | stiffness matrix |
| $[M]$ | = | mass matrix |
| max | = | maximum |
| min | = | minimum |
| $w$ | = | weights |
| $\{X\}$, $\{X'\}$, $\{X''\}$ | = | displacement, velocity, and acceleration vectors |
| $x$ | = | input to the neural network |
| $y$ | = | output to the neural network |
| $\alpha$ | = | prior contribution to the error |
| $\zeta_i$ | = | $i$th damping ratio |
| $\mu$ | = | mean of the input data |
| $\sigma$ | = | standard deviation |
| $\{\phi\}_i$, $\{\bar{\phi}\}_i$ | = | $i$th mode shape and complex mode shape vector |
| $\omega$, $\omega_i$, $\bar{\omega}_i$ | = | frequency, natural frequency $i$, complex natural frequency |
| $\{0\}$ | = | null vector |
| $\|^*\|$ | = | absolute value of * |
| $\wedge$ | = | covariance matrix |

*Subscripts*

| | | |
|---|---|---|
| $i, j, n, k, l, q$ | = | indices |

*Superscripts*

| | | |
|---|---|---|
| $K$ | = | number of network output units |
| $M$ | = | number of hidden units |
| $N$ | = | number of (measured degrees of freedom, modes, training examples) |
| $P$ | = | number of vectors in the training set |
| $T$ | = | transpose |

Received 6 March 2002; revision received 24 July 2002; accepted for publication 24 July 2002. 

*Director, P.O. Box 787391; tmarwala@yahoo.com.

## Introduction

VIBRATION data have been used with various degrees of success to identify damage in structures. Three types of signals have been used to this end: modal domain, for example, the modal properties; frequency domain, for example, frequency response functions (FRFs); and time-frequency domain, for example, the wavelet transforms.[1] In this paper a parameter called pseudomodal energy, defined as the integral of the FRF over various frequency bandwidths, which was introduced in Ref. 2, is used in conjunction with neural networks for fault identification in structures. The performance of the pseudomodal energies is compared to that of the modal properties, which have been used widely before Ref. 1. The application of neural networks for fault identification has been conducted widely before Refs. 1 and 3.

The pseudomodal energies and neural networks are tested on classifying faults on a simulated cantilevered beam and a population of 20 cylinders. The neural networks trained using pseudomodal energies are compared to those trained using modal properties with regard to classifying faults in structures.

## Pseudomodal Energies

Before the pseudomodal energy is introduced, the modal properties, which have been used extensively in fault identification in mechanical systems, are reviewed.[1] The modal properties are related to the physical properties of the structure. All elastic structures can be described in terms of their distributed mass, damping, and stiffness matrices in the time domain through the following expression:

$$[M]\{X''\} + [C]\{X'\} + [K]\{X\} = \{F\} \tag{1}$$

If Eq. (1) is transformed into the modal domain to form an eigenvalue equation for the $i$th mode, then[4]

$$\left(-\bar{\omega}_i^2[M] + j\bar{\omega}_i[C] + [K]\right)\{\bar{\phi}\}_i = \{0\} \tag{2}$$

The introduction of damage in structures changes the mass and stiffness matrices. From Eq. (2) it can be deduced that changes in the mass and stiffness matrices cause changes in the modal properties of the structure.

The pseudomodal energies are defined as the integrals of the real and imaginary components of the frequency response functions over various frequency ranges that bracket the natural frequencies. The FRFs can be expressed in receptance and inertance form.[4] On the one hand, receptance expression of the FRF is defined as the ratio of the Fourier-transformed displacement to the Fourier-transformed force. On the other hand, inertance expression of the FRF is defined as the ratio of the Fourier-transformed acceleration to the Fourier-transformed force. The pseudomodal energies can be expressed in terms of receptance and inertance forms in the same way as the FRFs are expressed in these forms. From the FRFs expressed as a function of modal properties, pseudomodal energies may be calculated as a function of the modal properties. This is done in order to infer the capabilities of pseudomodal energies to identify faults from those of modal properties. The most commonly used techniques to measure vibration data measure the acceleration response instead of the displacement response.[4] In such a situation it is better to calculate the inertance pseudomodal energies as opposed to the receptance pseudomodal energies. The inertance pseudomodal energy has been derived by integrating the inertance FRF written in terms of the modal properties by using the modal summation equation as follows[2]:

$$\mathrm{IME}_{kl}^{q} = \int_{a_q}^{b_q} \sum_{i=1}^{N} \frac{-\omega^2 \phi_k^i \phi_l^i}{-\omega^2 + 2\zeta_i \omega_i \omega j + \omega_i^2} \, \mathrm{d}\omega \tag{3}$$

In Eq. (3), $a_q$ and $b_q$ represent respectively the lower and the upper frequency bounds for the $q$th pseudomodal energy calculated from the FRF caused by excitation at $k$ and measurement at l. The lower and upper frequency bounds bracket the $q$th natural frequency. Assuming that damping is low, Eq. (3) becomes[2]

$$\mathrm{IME}_{kl}^{q} \approx \sum_{i=1}^{N} \left\{ \phi_k^i \phi_l^i (b_q - a_q) - \omega_i \phi_k^i \phi_l^i j \left[ \arctan\left( \frac{-\zeta_i \omega_i - j b_q}{\omega_i} \right) - \arctan\left( \frac{-\zeta_i \omega_i - j a_q}{\omega_i} \right) \right] \right\} \tag{4}$$

Equation (4) reveals that the inertance pseudomodal energy can be expressed as a function of the modal properties. The inertance pseudomodal energies can be calculated directly from the FRFs using any numerical integration scheme. Doing this avoids going through the process of modal extraction and using Eq. (4). The advantages of using the pseudomodal energies over the use of the modal properties are as follows: all of the modes in the structure are taken into account as opposed to using the modal properties, which are limited by the number of modes identified; and integrating the FRFs to obtain the pseudomodal energies smoothes out the zero-mean noise present in the FRFs.

## Neural Networks

In this paper neural networks are viewed as parameterized graphs that make probabilistic assumptions about data. Learning algorithms are viewed as methods for finding parameter values that look probable in the light of the data. Learning processes can occur by training the network through either supervised or unsupervised learning. Unsupervised learning is used when only the input data are available. To illustrate the unsupervised learning route in the context of structural dynamics, one can consider two kinds of failures in structures: failure caused by loosening of joints or cracks in the structure. If the responses of the two failures are inherently different, an unsupervised learning scheme can be employed to distinguish these types of failures as either belonging to class 1 or 2. Supervised learning is the case where the input ($x$) and the output ($y$) are both available and neural networks are used to approximate the functional mapping between the two. In this paper supervised learning is applied.

There are several types of neural-network procedures, some of which will be considered later, for example, multilayer perceptron (MLP) and radial basis function (RBF).[5] In this paper the MLP is used because it provides a distributed representation with respect to the input space as a result of cross coupling between hidden units, whereas the RBF provides only local representation. In this paper

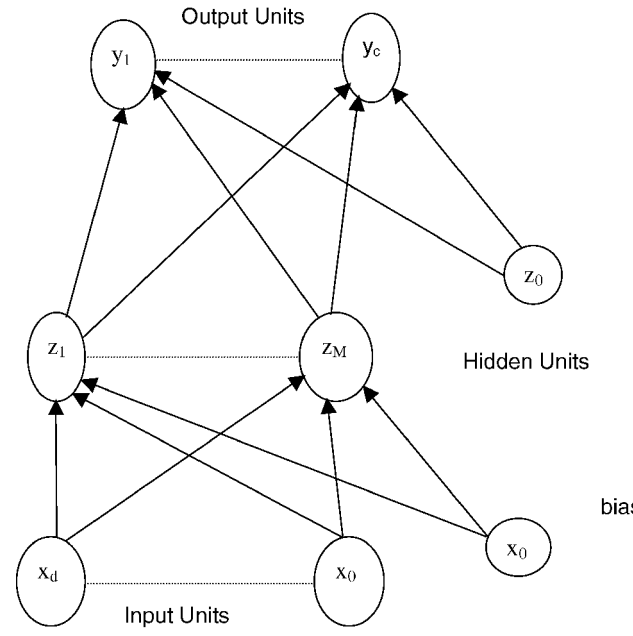


**Fig. 1 Feedforward network having two layers of adaptive weights.**

the MLP architecture contains a hyperbolic tangent basis function in the hidden units and logistic basis functions in the output units.[5] A schematic illustration of the MLP is shown in Fig. 1. This network architecture contains hidden units and output units and has one hidden layer. The bias parameters in the first layer are shown as weights from an extra input having a fixed value of $x_0 = 1$. The bias parameters in the second layer are shown as weights from an extra hidden unit, with the activation fixed at $z_0 = 1$. The model in Fig. 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 and this is an intrinsic property of the MLP networks. The MLP can be expanded by considering several layers, but it has been demonstrated by the Universal Approximation Theorem[6] that for all situations a two-layered architecture is adequate for the MLP. As a result of this theorem, in this paper a two-layered network shown in Fig. 1 is chosen. The relationship between the output $y$ and $x$ can be written as follows[5]:

$$y_k = f_{\text{outer}} \left\{ \sum_{j=1}^{M} w_{kj}^{(2)} f_{\text{inner}} \left[ \sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right] + w_{k0}^{(2)} \right\} \tag{5}$$

Here, $w_{ji}^{(1)}$ and $w_{ji}^{(2)}$ indicate weights in the first and second layer, respectively, going from input $i$ to hidden unit $j$, whereas $w_{j0}^{(1)}$ indicates the bias for the hidden unit $j$. In this paper the function $f_{\text{outer}}(\bullet)$ is logistic, whereas $f_{\text{inner}}$ is a hyperbolic tangent function. The logistic function is defined as follows:

$$f_{\text{outer}}(\nu) = 1/(1 + e^{-\nu}) \tag{6}$$

The logistic activation function maps the interval $(-\infty, \infty)$ onto a $(0,1)$ interval and can be approximated by a linear function provided the magnitude of $\nu$ is small. The hyperbolic tangent function is

$$f_{\text{inner}}(\nu) = \tanh(\nu) \tag{7}$$

Training the neural network identifies the weights in Eq. (5). In this paper the maximum-likelihood approach is used to identify a set of weights that maximizes the ability of a network to predict the output whenever presented with the input data. An optimization procedure is used to identify the weights and biases of the neural networks in Eq. (5). A cost function must be chosen in order to use the optimization technique. A cost function is a mathematical representation of the overall objective of the problem. In this paper the main objective, which is used to construct a cost function, is to identify a set of neural-network weights given vibration data and identity of faults. If the training set $D = \{x_k, y_k\}_{k=1}^{N}$ 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 $E$ can be written as follows using the cross-entropy cost function[5]:

$$E = -\sum_{n=1}^{N}\sum_{k=1}^{K}[t_{nk}\ln(y_{nk}) + (1-t_{nk})\ln(1-y_{nk})] + \frac{\alpha}{2}\sum_{j=1}^{W} w_j^2 \tag{8}$$

The cross-entropy function is chosen because it has been found to be more suited to classification problems than the sum-of-square of error cost function.[5] In Eq. (8), $n$ is the index for the training pattern, and $k$ is the index for the output units. The second term in Eq. (8) is the regularization parameter, and it penalizes weights of large magnitudes.[5] This regularization parameter is called the weight decay, and its coefficient $\alpha$ determines the relative contribution of the regularization term on the training error. This regularization parameter ensures that the mapping function is smooth. Including the regularization parameter has been found to give significant improvements in network generalization.[7] If $\alpha$ is too high, then the regularization parameter oversmoothes the network weights, thereby giving inaccurate results. If $\alpha$ is too small, then the effect of the regularization parameter is negligible; unless other measures that control the complexity of the model, such as the early stopping method,[5] are implemented, then the trained network becomes too complex and thus performs poorly on the validation set.

Before minimization of the cost function 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 because of a phenomenon known as overfitting caused by high variance.

The weights $w_i$ and biases (with subscripts 0 in Fig. 1) in the hidden layers are varied using optimization methods until the cost function is minimized. Gradient descent methods are implemented, and the gradient of the cost function is calculated using the back-propagation method.[5] Both the conjugate gradient[8] and the scaled conjugate gradient methods[9] were implemented at the preliminary stage of this research. It was decided to pursue the scaled conjugate gradient method because it was found to be computationally efficient and yet retains the essential advantages of the conjugate gradient technique. The reason behind higher computational efficiency of the scaled conjugate gradient method over the conjugate gradient method is not the subject of this paper but can be obtained in Ref. 6.

## Input to the Neural Networks

In many experimental examples there are more pseudomodal energies and modal properties than needed for fault identification, and all of these could not be possibly used for neural-network training. These data must therefore be reduced, and the reason for this reduction is discussed in this section. In the statistical community there is a phenomenon called the curse of dimensionality,[10] which refers to the difficulties associated with the feasibility of density estimation in many dimensions. It is therefore a good practice to reduce the dimension of the data, hopefully without the loss of essential information. This section deals with the techniques implemented in this paper to reduce the input space. The techniques implemented in this paper reduce the dimension of the input data by removing the parts of the data that do not contribute significantly to the dynamics of the system being analyzed or those that are too sensitive to irrelevant parameters such as slight changes in temperatures. To achieve this, we implement the statistical overlap factor (SOF) to select those data that show high sensitivity to faults. The SOF is defined as follows[2]:

$$\mathrm{SOF} = \left|\frac{\bar{\mu}_1 - \bar{\mu}_2}{(\sigma_1 + \sigma_2)/2}\right| \tag{9}$$

Here $\bar{\mu}_1$ and $\bar{\mu}_2$ are the means of distributions, $\sigma_1$ and $\sigma_2$ are their respective standard deviations, and $|\bullet|$ stands for the absolute value of $\bullet$. Here we calculate the SOFs between undamaged and damaged structure and select those input values that show higher SOFs. There is a possibility that the data selected using the SOF might be correlated. The principal component analysis (PCA) is used to reduce the data already reduced using the SOF into uncorrelated input space.

The PCA orthogonalizes the components of the input vector so that they are uncorrelated with each other. When implementing the PCA for data reduction, correlations and interactions among variables in the data are summarized in terms of a small number of underlying factors. The PCA has been successfully used to reduce the dimension of the data in the past.[5]

The variant of the PCA implemented in this paper finds the directions in which the data points have the most variance. These directions are called principal directions. The data are then projected onto these principal directions without the loss of significant information of the data. Here a brief outline of the implementation of the PCA adopted in this paper is described. The first step in the implementation of the principal component analysis is to construct a covariance matrix defined as follows[11]:

$$\wedge = \sum_{p=1}^{P}(x^p - \mu)(x^p - \mu)^T \tag{10}$$

Here $T$ is the transpose. The second step is to calculate the eigenvalues and eigenvectors of the covariance matrix and arrange them from the biggest eigenvalue to the smallest. The first $N$ biggest eigenvalues are chosen. The data are then projected onto the eigenvectors corresponding to $N$ most dominant eigenvalues.

Because the pseudomodal energies and modal properties extracted are to be used for neural-network training, it is important to normalize them so that those inputs that have higher magnitudes do not dominate the training. Here the scaling technique that is implemented ensures that all parameters fall within the interval [0,1]. To achieve this, the following scaling method is used:

$$x_m^{\mathrm{new}} = \frac{x_m^{\mathrm{old}} - \min\left(x_m^{\mathrm{old}}\right)}{\max\left(x_m^{\mathrm{old}}\right) - \min\left(x_m^{\mathrm{old}}\right)} \tag{11}$$

where $x_m$ is a row of the input parameters.

## Example 1: Simulated Beam

The pseudomodal energies and neural networks are applied to identify faults in a cantilevered beam illustrated in Fig. 2. The results are compared to the results obtained when the modal properties are used. The beam is made of aluminum with the following dimensions: length of the beam is 1.0 m, width is 50 mm, and thickness is 6 mm. The beam is free on one end and clamped on the other end and is restricted to translation in the $y$ axis and rotation about the $z$ axis. It is partitioned into three substructures and modelled with 50 elements (51 nodes) using the Structural Dynamics Toolbox[12] in MATLAB®. Node 51 is located at the free end of the beam, and node 1 is located at the clamped point of the beam. Because the beam is restricted to translation in the $y$ axis and rotation about the $z$ axis, the beam has 50 active nodes (because node 1 is clamped).

Using the Structural Dynamics Toolbox, the mass and stiffness matrices of size 100 by 100 are assembled. Here 50 active nodes, each with two degrees of freedom corresponding to translation in the $y$ plane and rotation about the $z$ axis, give 100 degrees of freedom and thus the mass and stiffness matrices of size 100 by 100. From the mass and stiffness matrices 100 modal properties are calculated. From the calculated modal properties inertance FRFs are calculated using the modal summation equation [an expression inside the integral in Eq. (3)] by assuming that the beam is lightly damped and fixing the damping ratios to 0.001. The FRFs calculated correspond to excitation at node 51, as shown in Fig. 2, and acceleration measurements in the direction shown in the same figure for all 50 active degrees of freedom. From the 50 FRFs calculated nine FRFs that correspond to nodes 3, 7, 13, 17, 23, 27, 33, 43, and 51 are selected.

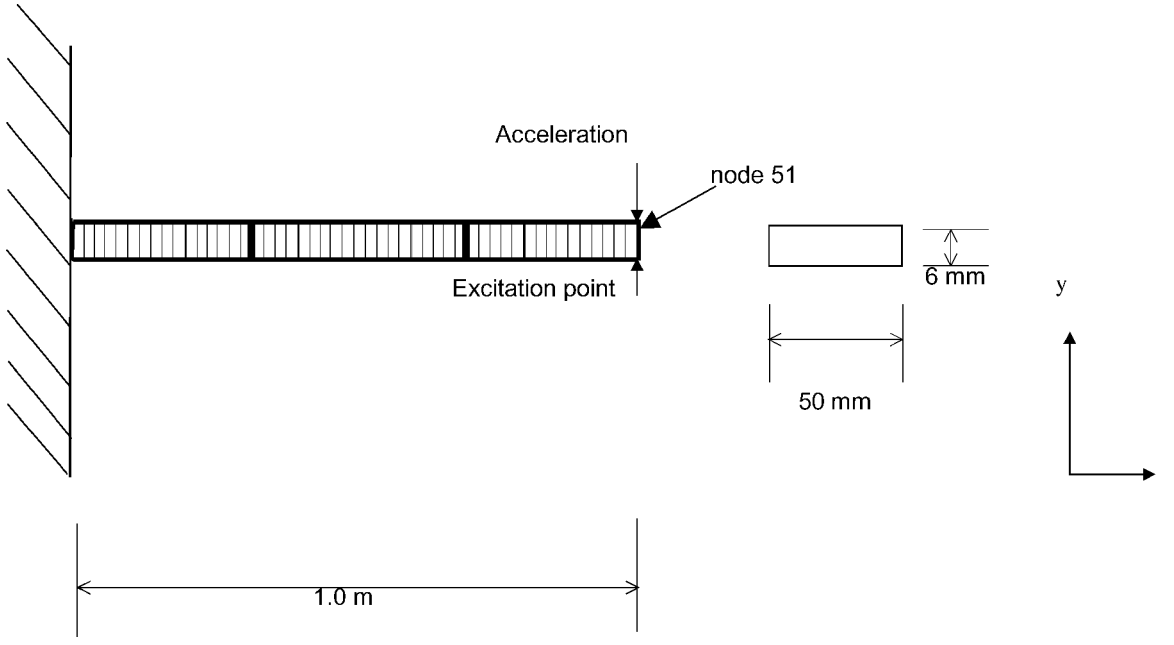


**Fig. 2 Cantilevered beam modelled with 50 elements and divided into three substructures.**

The FRFs that are selected exclude those corresponding to rotational degrees of freedom because in practical situations measuring the rotational degrees of freedom is difficult.

The nine FRFs are converted into the time domain and Gaussian noise of $\pm 0$, $\pm 1$, $\pm 2$, $\pm 3$, and $\pm 4\%$ are added to the time domain data. Using the Fourier transform technique, the FRFs are calculated from the time domain data. From the FRFs calculated from the time domain data, five modal properties are extracted, and the pseudomodal energies are calculated by integrating the FRFs at various frequency ranges. On calculating the pseudomodal energies, the following bandwidths are chosen in order to bracket each of the first five natural frequencies of the beam: 18–44, 155–240, 484–620, 1014–1151, and 1726–1863 Hz.

Ninety pseudomodal energies are extracted from one set of simulations (45 imaginary and 45 real parts of the pseudomodal energies). Forty-five conform to nine coordinates corresponding to translation along the $y$ direction and five modes. The total number of modal properties calculated is 50 (45 mode shape coordinates and five natural frequencies).

To generate different fault cases, the structure is divided into three substructures: substructure 1 contains elements 1 to 16, substructure 2 contains elements 17 to 32, and substructure 3 elements 33 to 50. Faults are introduced to the structure by simultaneously reducing the cross-sectional areas of three elements of the beam by 5 to 10%. If the reduction is 0 to 2%, then this is defined as the absence of faults. The fault present in substructure 1 is restricted to elements 7 to 10; for substructure 2 elements 24 to 27; and for substructure 3 elements 42 to 45. For a given fault in one substructure, the maximum reduction in cross-sectional area corresponds to a reduction of 0.8% of the total volume of the beam.

The data are generated by assigning a reduction in the cross-sectional areas, of 5 to 10% for the presence of faults and 0 to 2% for the absence of faults, of four of the elements in a substructure. Each fault case is assigned a fault identity, corresponding to a location of fault in the substructure. For example, a fault existing in substructure 1 and having a value of 8% would yield an output of [1 0 0]. In this paper we define the presence of fault in one substructure, for example, [1 0 0], as a single fault and the presence of faults in more than one substructure, for example, [1 1 0], as a multiple-fault case. The types of fault cases simulated are a zero-fault case [0 0 0]; one-fault cases [1 0 0], [0 1 0], and [0 0 1]; two-fault cases [1 1 0], [1 0 1], and [0 1 1]; and a three-fault case [1 1 1]. For each noise contamination of the data, 2400 data, are collected—300 for each of the 8 fault cases.

If all of the simulated degrees of freedom are used for training the networks with the pseudomodal energies, then the size of the input units will be 90 ($2 \times 45$ for the real and imaginary parts of the pseudomodal energies). For training the modal-property network, the network will have 50 (45 mode shapes and five natural frequencies) input units. In this paper both these input data sets are reduced from 90 and 50, respectively, for the pseudomodal energies and modal properties to 10. The pseudomodal energies are reduced from 90 to 40 using the SOF [see Eq. (9)] by choosing 40 input data that show the highest SOF between damaged and undamaged population. The remaining 40 pseudomodal energies are reduced to 10 using the PCA. The modal properties are reduced from 50 to 30 using the SOF by choosing 30 modal properties that show the highest SOFs. The remaining 30 modal properties are reduced to 10 using the principal component analysis.

## Training, Validation, and Testing

The procedure followed for training the MLP networks, in this example, uses the training, validation, and testing stages.[5] Each stage has a data set assigned to it, and ideally these three data sets should be of equal size and must be independent of one another. The network weights that map the input to output data identified by minimizing the network error and using the training data can overfit the training data. The neural network that overfits the training data might not necessarily perform well on the validation and test data sets. Overfitting the training data set is a situation where a network stops learning how to approximate the hidden dynamics of the system and learns the noise in the data. To combat this problem, more networks than required are trained, and the network that gives the least mean-squared errors on the validation data set is chosen. The chosen network might also overfit the validation data in addition to the training data set, and so the test data set is used to evaluate the performance of the trained network.

By using the finite element model, 800 vibration data are generated by perturbing the cross-sectional areas of the beam and used as a training set. This data set contains eight fault cases and 100 examples for each fault case. The validation data set with 800 examples is generated and used to select the neural-network architectures. The test data set is also generated and contains 800 examples. The validation and testing data sets each contain eight fault cases and 100 examples for each fault case.

Fifty pseudomodal-energy networks and 50 modal-property networks are trained using the training data set with zero-noise contamination and with the number of hidden units randomly chosen to fall from 8 to 16. On training these networks, the cross-entropy cost function shown in Eq. (8) is used because it has been found to be better suited for classification problems than the sum-of-square-of-errors cost function.[5] On training all of these networks, 100 iterations

are used and are found to be sufficient for convergence of the training error. These networks have 10 input data, which are chosen using the SOF and the PCA and three output units corresponding to three substructures. The logistic function, described by Eq. (6), is chosen as the output activation function, and the hyperbolic tangent function, described by Eq. (7), is chosen as the activation function in the hidden layer. The network is trained using the scaled conjugate gradient method.[9] On training these networks, the coefficient of the contribution of the regularization parameter $\alpha$, shown in Eq. (8), to the training error is set to 15. This value is chosen because it is found that it sufficiently smoothes out the network weights without compromising the abilities of the networks to generalize the data. Here smooth weight vectors are defined as vectors with components of the same order of magnitudes. Of the two sets of 50 trained networks, the two sets of networks that give the least classification errors on the validation data set are chosen. These classification errors are calculated by rounding off the fault identities given by the networks and calculating the proportion of fault cases classified correctly. From these two chosen networks the optimal sizes of the hidden units are 11 for the pseudomodal-energy network and 10 for the modal-property network.

Using the data contaminated with 1% Gaussian noise, 10 pseudomodal-energy networks and 10 modal-property networks are trained with different network-weights initializations, with the number of hidden units set to 11 and 10, respectively. The two sets of networks that give the least classification errors on the validation data set are chosen. The same process is repeated for 2, 3, and 4% Gaussian noise contamination of the data.

## Results and Discussion

The results showing the classification errors between the training and validation data sets are shown in Table 1. Each row in Table 1 shows a given noise contamination of the vibration data while the columns show the pseudomodal-energy network and modal-property network. For a given noise level and a given network the training and validation data sets give classification errors of the same order of magnitudes implying that the networks have not overfitted the training data. Table 1 shows that the higher the noise contamination of the data the higher the classification error. It also shows that, on average, the pseudomodal-energy network gives more accurate classification of faults than the modal-property network.

The trained networks are used to classify faults from the test data set into their respective classes. Table 2 shows the results obtained when the networks are used to classify faults in the test data into eight fault cases. The pseudomodal-energy network gives, on average, more accurate results than the modal-property-network. Table 2 shows that, in general, the accuracy of the methods decreases with the increase in the levels of noise contamination of the vibration data.

**Table 1 Classification errors when using the two approaches and various noise levels added to the data**

| | Training, % | | Validation, % | |
|---|---|---|---|---|
| Noise, % | PMEN[a] | MPN[b] | PMEN | MPN |
| 0 | 1.75 | 4.70 | 1.78 | 4.98 |
| 1 | 2.95 | 7.30 | 3.15 | 7.65 |
| 2 | 4.05 | 9.97 | 4.21 | 10.83 |
| 3 | 8.92 | 11.84 | 9.67 | 11.91 |
| 4 | 15.22 | 21.71 | 16.33 | 23.10 |

[a]PMEN, pseudomodal-energy network.
[b]MPN, modal-property network.

**Table 2 Accuracy results obtained when the networks are used to classify faults into eight fault cases[a]**

| Noise level, % | PMEN[b] | MPN[c] |
|---|---|---|
| ±0 | 98.1 | 94.9 |
| ±1 | 96.5 | 92.0 |
| ±2 | 95.6 | 88.5 |
| ±3 | 89.8 | 87.1 |
| ±4 | 83.1 | 75.6 |

[a]These results are obtained when the trained networks are assessed on the test data set.
[b]PMEN, pseudomodal-energy network.
[c]MPN, modal-property network.

## Example 2: Cylindrical Structure

**Experimental procedure**

In this section the procedure of using pseudomodal energies and neural networks is experimentally validated and compared to the procedure in existence of using modal properties and neural networks. The experiment is performed on a population of cylinders, which are supported by inserting a sponge rested on a bubblewrap, to simulate a "free–free" environment (see Fig. 3). The sponge is inserted inside the cylinders to control boundary conditions. This will be discussed further in the following paragraphs.

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. In this paper 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 the wrap is below 100 Hz. The first natural frequency of cylinders being analyzed is over 300 Hz, and this value is three times higher than the natural frequency of a cylinder on a bubble wrap. Therefore the cylinder on the wrap is effectively decoupled from the ground. The use a 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. The impulse hammer test is performed on each of the 20 steel seam-welded cylindrical shells ($1.75 \pm 0.02$ mm thickness, $101.86 \pm 0.29$ mm diam, and height $101.50 \pm 0.20$ mm). The impulse is applied at 19 different locations as indicated in Fig. 3: nine on the upper half of the cylinder and 10 on the lower half of the cylinder.

The structure is excited using a modal hammer of sensitivity of 4 pC/N, with the head mass of 6.6 g, and cutoff frequency of 3.64 kHz. The response is measured using an accelerometer with a sensitivity of 2.6 pC/ms$^{-2}$, which has a mass of 19.8 g. A small hole of size 3 mm is drilled into the cylinder, and the accelerometer is attached by screwing it through the hole. Mounting acceleration on curved surfaces can result in errors in the measurements. In this study the error in the measurements was observed not to distort the mode shapes identified, leading to a conclusion that the errors in measurements were low.

The sponge is inserted inside the cylinder to control boundary conditions by rotating it every time a measurement is taken. This controls the responses of different segments of the cylinders. 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 deg.

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 can 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 one 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 eight different types of fault cases considered (including [0 0 0]).

Because the zero-fault scenarios and the three-fault scenarios are overrepresented, 12 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 there are more zero-fault and three-fault scenarios than other fault types is because all cylinders tested give [0 0 0] and [1 1 1] fault types, whereas not all cylinders tested give all three one-fault, for example, [1 0 0], and three two-fault cases, for example, [1 1 0]. 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 (that is, drilled holes) do not influence damping.

Each cylinder is measured three times under different boundary conditions by changing the orientation of a rectangular sponge inserted inside the cylinder. The number of sets of measurements taken for an undamaged population is 60 (20 cylinders × 3 for different boundary conditions). All of the possible fault types and their respective number of occurrences are listed in Table 3. In Table 3 the numbers of one- and two-fault cases are each 72. This is because, as already mentioned, 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 Fig. 4. This figure indicates that the measurements are generally repeatable at low frequencies and are not repeatable at high frequencies. The reason for this is because higher-frequency modes are more sensitive to variations in structural properties than low-frequency modes. And because Fig. 4 shows data from a population of cylinders that are not necessarily dynamically identical, the variation in data is much more severe on high-frequency modes than on low-frequency modes. Axisymmetric structures such as cylinders have repeated modes as a result of their symmetry.[13] The presence of an accelerometer and the imperfection of cylinders destroys the axisymmetry of the structures. The incidence of repeated natural frequencies is destroyed, making the process of modal analysis easier to perform.[14]

From the FRFs the modal properties are extracted using modal analysis, and the pseudomodal energies are calculated using the

**Table 3 Number of different types of fault cases generated**

| Fault | Number |
|---|---|
| [000] | 60 |
| [100] | 24 |
| [010] | 24 |
| [001] | 24 |
| [110] | 24 |
| [101] | 24 |
| [011] | 24 |
| [111] | 60 |

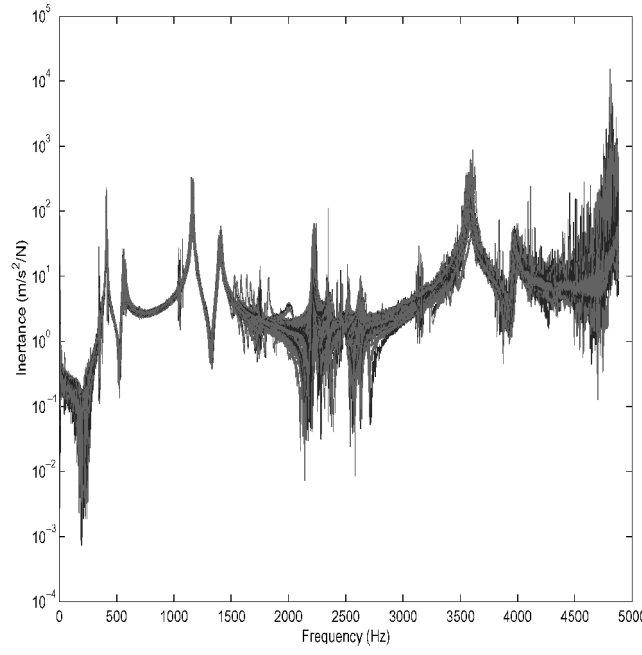


**Fig. 4 Measured FRFs from a population of cylinders.**

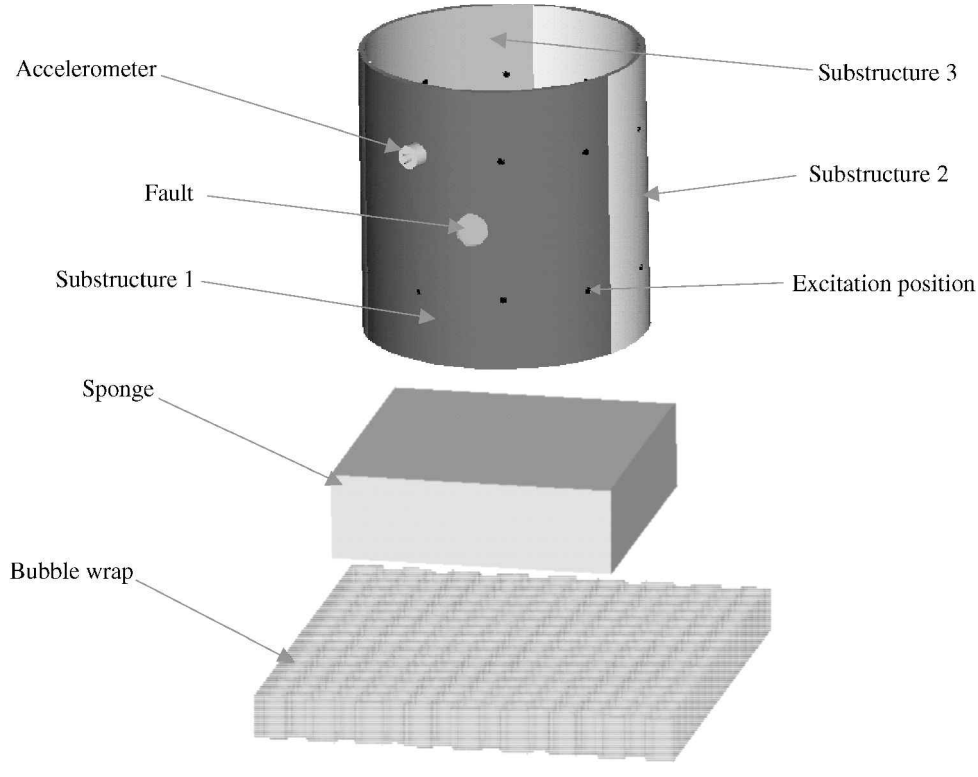


**Fig. 3 Illustration of a cylindrical shell showing the positions of the impulse, accelerometer, substructures, fault position, and supporting sponge.**

**Table 4 Fault cases used to train, cross validate, and test the networks[a]**

| Fault | Training set | Test set |
|---|---|---|
| [000] | 21 | 39 |
| [100] | 21 | 3 |
| [010] | 21 | 3 |
| [001] | 21 | 3 |
| [110] | 21 | 3 |
| [101] | 21 | 3 |
| [011] | 21 | 3 |
| [111] | 21 | 39 |

[a]The multifold cross-validation technique is used because of the lack of availability of the data.

integrals under the peaks for a given frequency bandwidth using the trapezoidal technique. The frequency spacing of the FRFs is 1.22 Hz. When the pseudomodal energies are calculated, frequency ranges spanning over 6% of the natural frequencies are chosen. These bandwidths are as follows in hertz: 393–418, 418–443, 536–570, 1110–1180, 1183–1254, 1355–1440, 1450–1538, 2146–2280, 2300–2440, 2250–2401, 2500–2656, 3140–3340, 3350–3565, 3800–4039, and 4200–4458. The guidelines outlined in Ref. 2 are taken into consideration when choosing these frequency ranges. These guidelines state that the frequency bandwidth 1) must be sufficiently narrow to capture the resonance behavior, 2) must be sufficiently wide to capture the smoothing out of zero-mean noise, and 3) must not include the regions of the antiresonance, which are generally noisy.

The pseudomodal energies are used to train the pseudomodal-energy network, and modal properties are used to train the modal-property network. To train the MLP neural networks, conventionally training, validation, and testing data sets are used,[5] as was done in the simulated example. Because there is a limited amount of data available for the experimental example, there is no validation data set put aside. The networks are validated using the multifold cross-validation technique.[5] For each fault case 21 data are randomly selected from the total number of fault cases measured, which are shown in Table 4.

The numbers of pseudomodal energies and modal properties identified are 646 (corresponding to 17 natural frequencies × 19 measured mode-shape coordinates × 2 for real and imaginary parts of the pseudomodal energy) and 340 (17 modes × 19 measured mode-shape coordinates + 17 natural frequencies), respectively. The SOF and the PCA are used to reduce the dimensions of the input data from 646 × 264 pseudomodal energies and 340 × 264 modal properties to 10 × 264 for both of these data types. Here 264 corresponds to the number of fault cases measured. The first stage of the input dimension reduction scheme is to use the SOF. Here 50 × 264 pseudomodal energies and 50 × 264 modal properties are selected from 646 × 264 and 340 × 264 identified, respectively. The 50 × 264 pseudomodal energies and modal properties that are chosen are sufficiently repeatable for a population of faultless cylinders yet sufficiently sensitive to the introduction of faults.

The main problem with using SOF is that the data selected might be correlated. To ensure that the data chosen are uncorrelated, the PCA is employed to reduce 50 × 264 pseudomodal energies and 50 × 264 MPs selected using statistical overlap factors to 10 × 264. (Here 10 rows are independent.) From the 50 × 264 pseudomodal energies and 50 × 264 modal properties the covariance matrix is calculated using Eq. (10). The eigenvalues and eigenvectors of the covariance matrix are calculated. Ten eigenvectors corresponding to the 10 largest eigenvalues are retained. The variance of the data retained when truncating 50 × 264 data to 10 × 264 is 90% for the pseudomodal-energy network and 85% for the modal-property network. This is calculated by dividing the sum of the first 10 dominant eigenvalues by the sum of 50 eigenvalues. The input data (50 × 264) are then projected onto the corresponding eigenvectors. The new input data are of dimension 10 × 264. In this section a procedure followed to choose the number of hidden units is described. The output vector to the neural networks is of dimension 3 × 1. For example, a fault in substructure $n$ has a fault-identity vector with the $n$th component containing a 1. Given the input data of size 10 × 168, the output vector of size 3 × 168 for the training set, a network with 157 weights (11 hidden units) is the largest network that could be constructed. Here a value 168 corresponds to the number of training examples, as indicated in Table 4.

**Table 5 Accuracy of the classification of fault cases**

| Network | Accuracy, % |
|---|---|
| Pseudomodal energy | 82.29 |
| Modal property | 81.25 |

In this paper 20 pseudomodal-energy networks and 20 modal-property networks are trained by randomly choosing the number of hidden units to fall from 7 and 11. The networks (one pseudomodal-energy network and one modal-property network) that give the least mean-squared errors, during cross validation, are selected. The next paragraph describes the details on how the networks are trained, cross validated using multifold method, and tested.

The training data set of size 168 is partitioned into 21 subsets. Each partition has eight different fault cases. This ensures that the training case is balanced in terms of the proportion of fault cases present. The first sets of networks, that is, pseudomodal-energy network and modal-property network (20 for each method), are trained with 160 data (from partitions 2 to 21), and the networks are validated on the remaining eight fault cases (from partition 1). The network weights identified in the preceding sentence are used as initial weights for training case 2. The training for this case is conducted using all partitions except partition 2, which is used to validate the trained networks. The complete training and validation of the networks is repeated 21 times until all of the validation partitions have been used.

Twenty pseudomodal energies with the number of hidden units randomly chosen to fall from 7 and 11 are trained and validated using the multifold cross-validation technique.[5] The same procedure is used to train 20 modal-property networks. From these two sets of 20 trained networks, the pseudomodal-energy network and modal-property network that give the least mean-squared errors over the validation partitions are chosen. Each validation partition gives a mean-squared error. The average of the mean-squared errors of all of the partitions is the validation error used to select the networks. The pseudomodal-energy network and modal-property network that have the least mean-squared errors have eight and nine hidden units, respectively. The classification accuracy rate, defined as the proportion of fault cases that are classified correctly into eight fault cases, when using the pseudomodal-energy network on the training data set is 84.1%. The classification accuracy rate when using the modal-property network on the training data set is 83.5%.

### Classification of Faults

In this section the pseudomodal-energy network and modal-property network are used to classify faults in the test data set listed in Table 4. This is achieved by classifying fault cases into eight classes.

A summary of the classification results are in Table 5. This table is obtained by calculating the proportion of fault cases that are classified correctly into the eight fault cases. This table shows that on classifying all fault cases the pseudomodal-energy network gives marginally more accurate results than the modal-property network.

## Conclusions

In this paper the pseudomodal energies and neural networks are used to classify faults in structures and are compared to when the modal properties are used as inputs to the neural networks. Two examples are considered: a simulated beam and a population of cylindrical shells. The statistical overlap factor and the principal component analysis are used to reduce the dimension of the input data. The results obtained show that the pseudomodal-energy networks perform marginally better than the modal-property network on classifying fault cases.

## Acknowledgments

The author thanks Hugh Hunt and Ruddy Blonbou for going over the manuscript and Cambridge University Engineering Department for supplying the equipment to conduct this work.

## References

[1]Doebling, S. W., Farrar, C. R., Prime, M. B., and Shevitz, D. W., "Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in Their Vibration Characteristics: a Literature Review," Los Alamos National Lab., TR LA-13070-MS, Albuquerque, NM, May 1996.

[2]Marwala, T., "On Fault Identification Using Pseudo-Modal-Energies and Modal Properties," *AIAA Journal*, Vol. 39, No. 8, 2001, pp. 1608–1618.

[3]Marwala, T., and Hunt, H. E. M., "Fault Identification Using Finite Element Models and Neural Networks," *Mechanical Systems and Signal Processing*, Vol. 13, No. 3, 1999, pp. 475–490.

[4]Ewins, D. J., *Modal Testing: Theory, Practice and Application*, Research Studies Press, Letchworth, England, U.K., 2000, pp. 30–35.

[5]Bishop, C. M., *Neural Networks for Pattern Recognition*, Oxford Univ. Press, Oxford, 1995, pp. 384–433.

[6]Haykin, S., *Neural Networks*, Prentice–Hall, Upper Saddle River, NJ, 1995, pp. 156–255.

[7]Hinton, G. E., "Learning Translation Invariant Recognition in Massively Parallel Networks," *Proceedings PARLE Conference on Parallel Architectures and Languages Europe*, Vol. 1, edited by J. W. de Bakker, A. J. Nijman, and P. C. Treleaven, Springer, Berlin, 1987, pp. 1–13.

[8]Shanno, D. F., "Conjugate Gradient Methods with Inexact Searches," *Mathematics of Operations Research*, Vol. 3, No. 3, 1978, pp. 244–256.

[9]Møller, M., "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning," *Neural Networks*, Vol. 6, No. 4, 1993, pp. 525–533.

[10]Bellman, R., *Adaptive Control Processes: A Guided Tour*, Princeton Univ. Press, Princeton, NJ, 1961, pp. 1–274.

[11]Jollife, I. T., *Principal Component Analysis*, Springer-Verlag, New York, 1986, pp. 1–502.

[12]Balmès, E., *Structural Dynamics Toolbox User's Manual*, Scientific Software Group, Ver. 2.1, Sèvres, France, 1997.

[13]Royston, T. J., Spohnholtz, T., and Ellingson, W. A., "Use of Non-Degeneracy in Nominally Axisymmetric Structures for Fault Detection with Application to Cylindrical Geometries," *Journal of Sound and Vibration*, Vol. 230, No. 4, 2000, pp. 791–808.

[14]Maia, N. M. M., and Silva, J. M. M., *Theoretical and Experimental Modal Analysis*, Research Studies Press, Letchworth, England, U.K., 1997, pp. 1–488.

C. Pierre
*Associate Editor*