Research Article

Real-Time Structural Damage Assessment Using Artificial Neural Networks and Antiresonant Frequencies

V. Meruane and J. Mahu

Department of Mechanical Engineering, University of Chile, Beauchef 850, 8370448 Santiago, Chile

Correspondence should be addressed to V. Meruane; vmeruane@ing.uchile.cl

The main problem in damage assessment is the determination of how to ascertain the presence, location, and severity of structural damage given the structure's dynamic characteristics. The most successful applications of vibration-based damage assessment are model updating methods based on global optimization algorithms. However, these algorithms run quite slowly, and the damage assessment process is achieved via a costly and time-consuming inverse process, which presents an obstacle for real-time health monitoring applications. Artificial neural networks (ANN) have recently been introduced as an alternative to model updating methods. Once a neural network has been properly trained, it can potentially detect, locate, and quantify structural damage in a short period of time and can therefore be applied for real-time damage assessment. The primary contribution of this research is the development of a real-time damage assessment algorithm using ANN and antiresonant frequencies. Antiresonant frequencies can be identified more easily and more accurately than mode shapes, and they provide the same information. This research addresses the setup of the neural network parameters and provides guidelines for the selection of these parameters in similar damage assessment problems. Two experimental cases validate this approach: an 8-DOF mass-spring system and a beam with multiple damage scenarios.

1. Introduction

Damage assessment must detect and characterize damage at the earliest possible stage and estimate how much time remains before maintenance is required, the structure undergoes failure, or the structure is no longer usable. Damage assessment offers tremendous potential for life and safety and/or economic benefits because it reduces the maintenance cost and increases the safety and reliability of the structure. Damage assessment is a subject of great importance for several industry sectors as well as for the safety of citizens and can be applied to civil engineering structures (such as buildings or bridges), transport vehicles (such as airplanes, helicopters, trains, ships, or cars), and industrial equipment (such as mills, turbines, pumps, and boilers, among others).

The main problem of damage assessment is to ascertain the presence, location, and severity of structural damage given the structure's dynamic characteristics. The most

successful applications of vibration-based damage assessment are model updating methods based on global optimization algorithms [1–5]. Model updating is an inverse method that identifies the uncertain parameters in a numerical model and is commonly formulated as an inverse optimization problem. In inverse damage detection, the algorithm uses the differences between the models of the structure that are updated before and after the presence of damage to localize and determine the extent of damage. The basic assumption is that the damage can be directly related to a decrease of stiffness in the structure. Nevertheless, these algorithms are exceedingly slow, and the damage assessment process is achieved via a costly and time-consuming inverse process, which presents an obstacle for real time health monitoring applications. Recent studies have introduced artificial neural networks (ANN) as an alternative to model updating in damage assessment [6–8]. A trained neural network can potentially detect, locate, and quantify structural damage

in a short period, and, hence, it can be used for real time damage assessment. Damage detection by ANN has the advantage that it is a general approach. Unlike many other damage detection methods that are often developed for specific quantities [9–11], in principle, ANN can be applied to any correlation coefficient that is sensitive to damage. Additionally, ANN can be used with structures that exhibit a nonlinear response [12].

One of the main challenges in structural damage assessment is the selection of an appropriate measure of the system response that is sufficiently sensitive to small damage. The fact that many measures have been studied over the past years (and continue to be investigated) with no consensus as to the optimum choice is a testament to the difficulty of the problem. This measure can be constructed in the time, frequency, or modal domains, and the latter two are the most broadly used.

The idea of directly using the frequency response functions (FRFs) to train the neural networks has attracted many researchers. Among all of the dynamic responses, the FRF is one of the easiest to obtain in real time because the in situ measurement is straightforward. However, the number of spatial response locations and spectral lines is overly large for neural network applications. The direct use of FRFs will lead to networks with a large number of input variables and connections, thus rendering them impractical. Hence, it becomes necessary to extract features from the FRFs and use these features as inputs to the neural networks. Castellini and Revel [13] presented an algorithm to detect and locate structural damage based on laser vibrometry measurements and a neural network for data processing. Using features extracted from the frequency response functions as inputs to the neural network, the authors were able to use the same network to detect and locate damage in three different experimental structures. To reduce the number of input variables, Zang and Imregun [14] applied a principal component analysis (PCA) technique to the measured FRFs. The output of the neural network gives the actual state of the structure: undamaged or damaged. The algorithm was able to distinguish between the undamaged and damaged cases with satisfactory accuracy. Fang et al. [15] selected key spectral points near the resonance frequencies in the FRF data. These selected points were used as the inputs of a neural network, and the outputs were the stiffness reduction factors. The algorithm showed high accuracy in identifying damage to a simulated cantilever beam under different damage scenarios.

The input data can be further reduced if modal analysis is performed first. Thus, the input variables are the modal parameters of the structure. The natural frequencies and mode shapes are the most frequently used parameters; a large body of literature on the use of natural frequencies in damage detection is available [16]. However, the natural frequencies do not always provide the spatial information necessary to locate the damage, and thus additional information is also necessary, such as the mode shapes. In general, the natural frequencies detect and quantify the presence of damage, whereas the mode shapes provide the location [17]. Zapico et al. [18] presented an algorithm designed to identify the damage to an experimental two-floor structure; they used

the natural frequencies as the input variables and trained a neural network with data from a numerical model. The neural network contained two outputs representing the level of damage of each floor. The proposed methodology was able to predict the damage with an error of less than 8.6%. Yun et al. [19] used the natural frequencies and mode shapes as input data for a neural network used to detect damage in the joints of framed structures. The neural network was trained using a noise-injection learning algorithm to reduce the effects of experimental noise. The authors found that the algorithm could estimate damage with reasonable accuracy, but the performance strongly depended on the level of experimental noise. A disadvantage of using mode shape displacements is that the mode shape identification requires many measurement locations. Furthermore, mode shape sensitivity to damage is not significant [3]. The mode shape curvatures are more sensitive to small structural modifications than the mode shape displacements [20]. Sahin and Shenoi [21] studied the effectiveness of different combinations of global (natural frequencies) and local (mode shape curvatures) vibration data as inputs for an artificial neural network. They concluded that the best performance was obtained when the natural frequencies were used as the inputs to a first network used to predict the severity of the damage and the mode shape curvatures were used as the inputs to a second network to predict the damage location.

Recently, researchers have proposed the use of anti-resonant frequencies as an alternative to mode shapes [22]. Anti-resonant frequencies correspond to the zeros (dips) of the FRFs and offer an attractive alternative because they can be determined more easily and with less error than the mode shapes while still providing the same information. The antiresonances can be derived from the point frequency response functions (FRFs), in which the response coordinate is the same as the excitation coordinate, or from transfer FRFs in which the response coordinate differs from the excitation coordinate. Point FRFs are preferred because matching problems arise when the antiresonances from transfer FRFs are used. Moreover, the distribution of the transfer antiresonances can be significantly modified with small structural changes [22]. However, the procedure used to obtain point FRFs differs from common modal testing; that is; the excitation degree of freedom (DOF) is shifted together with the response DOF, which may become impractical or expensive. Williams and Messina [23] introduced antiresonances from point FRFs to the Multiple Damage Location Assurance Criterion (MDLAC) algorithm. They concluded that the incorporation of anti-resonant data improves the accuracy of the damage predictions. Dilena and Morassi [24] studied the problem of crack detection in beams using resonant and anti-resonant frequencies. They found that the use of antiresonances aids in avoiding the nonuniqueness of the damage location that occurs when only natural frequencies are used. However, they also found that the experimental noise and modeling errors are usually amplified when antiresonances are included. Bamnios et al. [25] proposed a scheme for crack location in beams using the shift in the first anti-resonant frequency versus the measuring position to detect and locate a crack. They stated that this

method could be used to roughly locate the crack, and, subsequently, other methods can be applied to determine the crack characteristics more precisely. The changes in the resonant and anti-resonant frequencies were used by Inada et al. [26] to locate and quantify the delamination of a composite beam. This group implemented a two-step procedure: first, the delamination domain is identified from the anti-resonant changes, and, next, the location and size are defined using natural frequency changes. This method was effective in identifying the delamination locations and sizes. Wang and Zhu [27] suggested a method for identifying cracks in beams, which makes use of the natural frequencies and antiresonances from the point FRFs. The methodology is similar to that proposed by Bamnios et al. [25] in that the shift in the first anti-resonant frequency versus the driving point location is used to locate damage. They validated the method with a numerical example of a simply supported beam with three cracks. Their method was able to accurately predict the locations and qualitatively estimate the crack size. Meruane and Heylen [28] showed that antiresonances are a good alternative to mode shapes in damage assessment. However, they stated that further research is required for identification of the experimental antiresonances and for the matching of experimental and numerical antiresonances. Thus, antiresonances present an attractive alternative to mode shapes as the input values of neural networks for the following reasons: they require a lower number of inputs, are less contaminated by noise and provide the same information. Nevertheless, methods that use antiresonances are still under development, and the application of antiresonances to structural damage detection has not been fully investigated, primarily because the inverse optimization problem used with antiresonances is particularly challenging and robust optimization algorithms are needed. However, these limitations should not present a problem for methods based on neural networks.

1.1. Artificial Neural Networks.
An ANN is a data processing algorithm that attempts to emulate the processing scheme of the human brain [29]. An ANN is formed by "neurons" that are interconnected to build a complex network. Knowledge is acquired by a learning process and stored in the interneuron connections known as the "synaptic weights."

Different types of network architectures exist, and among them, the multilayer perceptron (MLP) is the most frequently used. An MLP network consists of an array of input neurons known as the input layer, an array of output neurons known as the output layer, and a number of hidden layers. Each neuron receives a weighted sum from the neurons in the preceding layer and provides an input to every neuron of the next layer. The activation of each neuron is governed by a function known as the transfer function. Typical selections for the transfer function are linear, log-sigmoid, and hyperbolic tangent-sigmoid, as shown in Figure 1.

Figure 2 illustrates the principle of a damage assessment algorithm using ANN. The vibration characteristics of the structure act as the inputs to the neural network, and the outputs are the damage indices of each element in the structure.

The outputs of a three-layer MLP are given by

$$y_i = f\left(\sum_{j=1}^{\text{NH}} \left(w_{ij} f\left(\sum_{k=1}^{\text{NI}} v_{jk} x_k + b_{vk}\right) + b_{wi}\right)\right), \quad (1)$$

where $x = \{x_1, x_2, \ldots, x_n\}$ and $y = \{y_1, y_2, \ldots, y_n\}$ are the input and output vectors, w_{ij} and v_{jk} are the interconnection weights, b represents the bias (or threshold) terms, $f(\cdot)$ is the transfer function, and NI and NH are the number of input and hidden nodes, respectively.

The values of the weights are updated by training the neural network with data from the structure. The training problem consists of finding the weights that will minimize the mean square error E_s:

$$E_s = \frac{1}{\text{NO}} \sum_{k=1}^{\text{NO}} (y_k - o_k)^2, \quad (2)$$

where o_k is the desired output at the kth output node, and NO is the number of output nodes. Many optimization techniques are able to address the network-training problem. The simplest method is gradient descent, also known as the steepest descent. This algorithm begins with an initial weight vector and iteratively updates it by moving in the direction of the greatest error decrease. The gradient descent rule can be written as

$$\Delta w_i^\tau = -\eta \frac{\partial E_s}{\partial w_i^\tau}, \quad (3)$$

where Δw_i^τ is the change for the ith weight at step τ. The backpropagation algorithm calculates the gradient at each step. The convergence of this algorithm strongly depends on the learning rate η. If η is too large, the algorithm may overshoot, leading to divergent oscillations. However, if η is too small, the search might proceed quite slowly. Fang et al. [15] showed that an adaptive learning rate significantly improves the training performance. Another improvement to the gradient descent algorithm involves ensuring that each search direction is conjugate to all previous directions, thus avoiding unnecessary loops in the search process. The scale-conjugated gradient algorithm proposed by Møller [30] combines this concept and variable steps. Many additional optimization algorithms have been proposed to train neural networks, but the Levenberg-Marquant algorithm has been the most efficient [31]. This method is a gradient-based algorithm specifically designed to minimize the sum-of-squares error [32].

A disadvantage of ANN is the need for large training sets. It is highly difficult and time-consuming to produce sufficiently large training data sets from experiments. An alternative to generating training samples is to use a numerical model of the structure. Castellini and Revel [13] showed that it is possible to produce correct damage predictions in an experimental structure using a neural network that was trained with samples generated by a finite element model. Nevertheless, this approach is highly dependent on the accuracy of the numerical model. There are two approaches
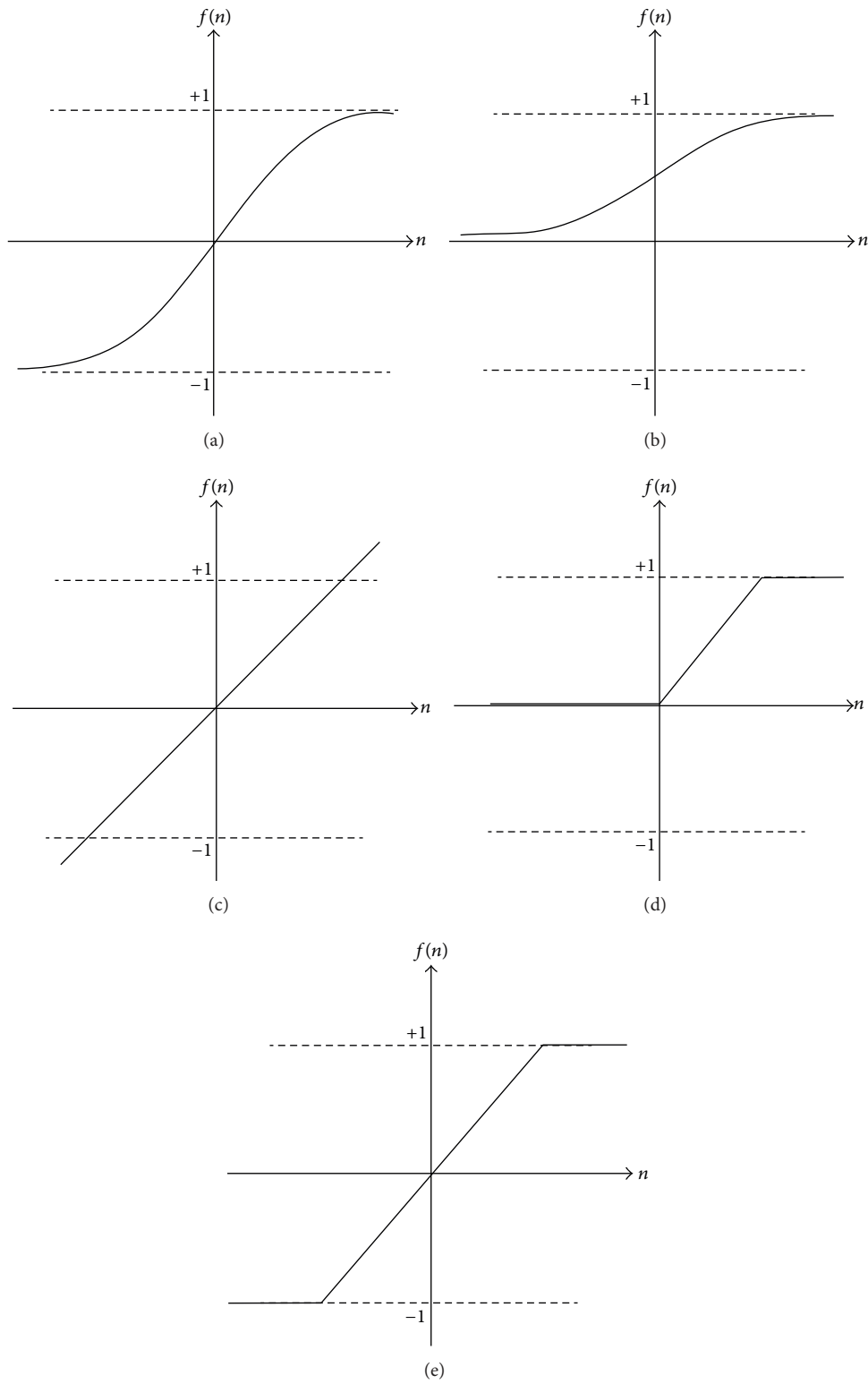
FIGURE 1: Transfer functions: (a) hyperbolic tangent sigmoid, (b) logarithm sigmoid, (c) linear, (d) saturating linear, and (e) symmetric saturating linear.

to overcome this problem. The first is to update the numerical model using experimental data from the undamaged structure. However, even after updating, differences will still remain between the numerical and experimental models. The second alternative is to define an input parameter that considers the initial errors in the numerical model, thus avoiding the need for an accurate numerical model. This goal is achieved using the changes in the data instead of their absolute values. The main assumption is that any change in the structural properties is caused by damage. Thus, any error in the undamaged model of the structure that is also present in the damaged model will be removed [33]. Lee et al. [34] showed that natural frequency changes because structural damage in a system without modeling errors is approximated the same as those in a system with modeling error. Hence, changes in the natural frequencies are less sensitive to modeling errors than the natural frequencies themselves. This group demonstrated the applicability of a neural network trained with mode shape changes. The damage locations were estimated with reasonable accuracy, although false alarms were detected at several locations.

Simulated data derived from a numerical model are noise-free, whereas actual measurements are never free from experimental noise. Noise in the measurements will cause the network to estimate parameters that are different from the actual properties of the structure. A solution is to introduce artificial noise into the numerical data used to train the network. This process is known as data perturbation scheme [35]. Yun et al. [19] used a noise-injection learning algorithm and a data perturbation scheme. They implemented a neural-network-based damage assessment algorithm to detect the damage in structural joints. The joints are modeled as semirigid connections using rotational springs, and accurate results are obtained in the cases of moderated noise. Sahin and Shenoi [21] trained a neural network with artificially added noise to detect single damage in beam-like composite laminates. The network contains two outputs: the location and the amount of damage. They studied different combinations of features extracted from the resonant frequencies and mode shape curvatures as inputs to the neural network, and their results show that feature selection plays a crucial role in the predictions accuracy. Zang and Imregun [14] proposed a different approach to address experimental noise and used principal component projection of the FRF data as an input to a neural network with two outputs of healthy or damaged. The authors stated that reconstructing the response using the highest principal components should not only achieve data compression but also remove a proportion of the noise. The PCA compression acts as a noise filter, which can be useful in modal analysis as well.

The successful application of a neural network depends on the representation and the learning algorithms. Nevertheless, their selection is problem-dependent and is usually determined by trial and error [36]. Sahoo and Maity [7] used a Genetic Algorithm (GA) to automate the trial-and-error process. The network parameters (number of neurons, learning rate, etc.) are set as variables in an optimization problem handled by a GA. They used an MLP network with two hidden layers trained by a backpropagation algorithm.
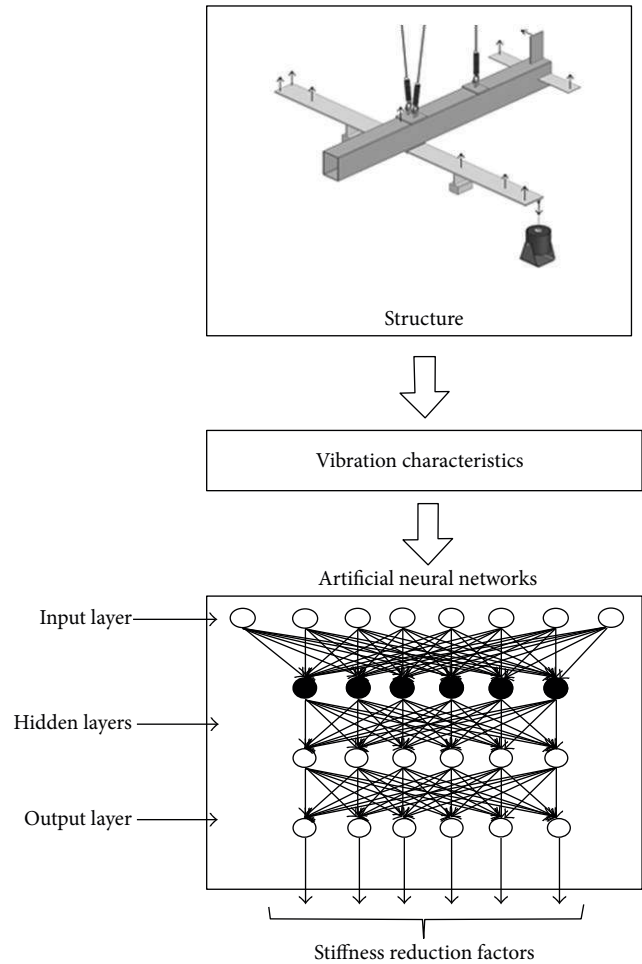

Figure 2: Damage assessment with artificial neural networks.

The inputs are the natural frequencies and strains in certain locations of the structure. Fang et al. [15] explored the use of a tunable steepest descent (TSD) algorithm that dynamically adjusts the learning speed during the training process. They showed that this methodology significantly increases the training speed while maintaining the learning stability. These authors used FRF data measured at key spectral points around the resonant frequencies.

The primary contribution of this research is the development of a real time damage assessment algorithm using ANN and anti-resonant frequencies. In literature, similarly to mode shapes, anti-resonant frequencies are used as complementary information to natural frequencies. Nevertheless, Mottershead [37] demonstrated that the sensitivity of anti-resonant frequencies can be expressed as a combination of the sensitivities of mode shapes and natural frequencies. Hence, anti-resonant frequencies contain the same information as natural frequencies and mode shapes together, which is the hypothesis of the present study. This study is restricted to anti-resonant frequencies obtained from point FRFs. Although measurement of the point FRFs is time-consuming because the excitation point is moved together with the response

location, this method offers several advantages for damage assessment as follows.

(i) All antiresonances contain independent information because they correspond to the resonant frequencies of the system grounded at different degrees of freedom.

(ii) For a given FRF, the number of anti-resonant frequencies does not change from one damage scenario to the next; this is a valuable property if we require these frequencies to act as inputs to a neural network.

(iii) An anti-resonant frequency always lies between two resonant frequencies. Hence, there is no doubt whether a dip in an FRF is an anti-resonant frequency or a minimum.

This study addresses the setup of the neural network parameters and provides guidelines for their selection in similar damage detection problems. The proposed methodology is evaluated with two experimental structures: an 8-DOF mass-spring system and a beam with multiple damage scenarios.

2. Neural Network for Damage Assessment

This study intends to train a neural network using anti-resonant frequencies and to determine its feasibility in assessing experimental damage. Hence, this study works with the simplest neural network that has been able to detect, locate, and quantify structural damage, the multilayer perceptron (MLP) with three layers (input, hidden, and output) as presented by Fang et al. [15]. The outputs of the network correspond to the damage indices of each element, whereas the inputs are the changes in anti-resonant frequencies. The number of hidden nodes is defined after a sensitivity analysis for each application case.

2.1. Inputs. As proposed by Lee et al. [34], the inputs to the neural network are defined as the changes in the modal parameters rather than their absolute values. With this approach, the network is less sensitive to errors in the baseline FE model.

Therefore, the inputs correspond to the experimental change in the anti-resonant frequencies with respect to the intact case:

$$x_{i \times n} = \frac{\omega_{r,i,n}^{D} - \omega_{r,i,n}^{U}}{\omega_{r,i,n}^{U}}. \tag{4}$$

The superscripts D and U denote undamaged and damaged respectively, and $\omega_{r,i,n}$ is the ith anti-resonant frequency of the nth FRF.

To reduce the effects of experimental noise, the simulated data are polluted with random noise. As proposed by Hjelmstad and Shin [35], each set of perturbed data is created by adding uniformly distributed random noise to the numerical data:

$$\omega_{r,i,n} = \omega_{r,i,n} \left(1 + \xi\right), \tag{5}$$

where ξ is a uniform random number with a specified amplitude. The variance of the perturbing noise should be the same as the variance of the measurement noise.

2.2. Outputs. The damage indices are represented by the elemental stiffness reduction factors β_i defined as the ratios between the stiffness reduction and the initial stiffness:

$$y_i = \beta_i. \tag{6}$$

The stiffness matrix of the damaged structure $\mathbf{K}_d$. is expressed as a sum of element matrices multiplied by reduction factors,

$$\mathbf{K}_d = \sum_i \left(1 - \beta_i\right) \mathbf{K}_i. \tag{7}$$

The value $\beta_i = 0$ indicates that the element is undamaged, whereas $0 < \beta_i \leq 1$ implies partial or complete damage.

2.3. Training and Validation Patterns. The distribution of the training patterns plays a crucial role in the success of a neural network. The relationship between the anti-resonant frequencies and the different damage levels is not linear, and, as a consequence, the network might be unable to interpolate. Fang et al. [15] recommend the use of training patterns with evenly distributed damaged levels. Combining simultaneous damages into the training increases the number of training patterns, and a large number of training patterns could overwhelm the training procedure. In this study, the training patterns were generated by considering up to two simultaneous damage incidences with nine damage levels evenly distributed between 0% and 80%. Hence, the total number of training patterns depends on the number of elements N as

$$\text{number of training patterns} = 9^2 \times C\left(N, 2\right), \tag{8}$$

where $C(N, 2)$ is the number of combinations of N elements taken two at a time.

The algorithm trains the network using the *early stopping technique* [37]. In this technique, two sets of data are used during training: a *training set* and a *validation set*. The training set is used for updating the network weights and biases. As defined in (2), the mean square error evaluated on the validation set is monitored during training. When the validation error increases for six iterations, the algorithm stops the training. The weights of the network at the minimum validation error become the final weights.

The validation set is a group of patterns that must be different from the training patterns. To ensure this condition, the validation patterns were created with nine damage levels evenly distributed between 5% and 85% and considered up to two simultaneous damage incidences. Note that the number of training patterns is equal to the number of validation patterns. This set of validation patterns is also used to check the network performance after training.

2.4. Measures of Network Performance. The performance of the network is measured by three indicators: the mean sizing

error, the damage missing error, and the false alarm error as defined by Yun et al. [19]. The mean sizing error (MSE) is the average quantification error:

$$\text{MSE} = \frac{1}{\text{NO}} \sum_i |y_i - o_i|, \qquad (9)$$

where y_i and o_i are the estimated and desired outputs for node i, respectively, and NO is the number of output nodes.

The damage missing error (DME) is given by

$$\text{DME} = \frac{1}{\text{NT}} \sum_i \varepsilon_i^I, \quad 0 \le \text{DME} \le 1, \qquad (10)$$

where $\varepsilon_i^I = 0$ if the ith damaged element is correctly detected and $\varepsilon_i^I = 1$, otherwise. The NT corresponds to the number of true damage locations. If DME = 0, all damage locations are correctly detected. The value of ε_i^I is calculated using the following equation:

$$\varepsilon_i^I = \begin{cases} 1 & \text{if } o_i > 0, \ y_i \le \alpha_c, \\ 0 & \text{otherwise.} \end{cases} \qquad (11)$$

It is assumed that an element is detected as damaged if the estimated damage y_i is greater than a prescribed critical value α_c. The critical damage level α_c is defined as being equal to the average MSE. This value is the minimum damage that the network can reasonably assess.

The false alarm error (FAE) is defined as

$$\text{FAE} = \frac{1}{\text{NF}} \sum_i \varepsilon_i^{II}, \quad 0 \le \text{FAE} \le 1, \qquad (12)$$

where $\varepsilon_i^{II} = 0$ if the ith detected damage is truly damaged and $\varepsilon_i^{II} = 1$, otherwise. The NF is the number of predicted damage locations. If FAE = 0, all of the detected locations are actual damage locations. The value of ε_i^{II} is calculated using the following equation:

$$\varepsilon_i^{II} = \begin{cases} 1 & \text{if } y_i \ge \alpha_c, \ o_i = 0, \\ 0 & \text{otherwise.} \end{cases} \qquad (13)$$

3. Identification and Matching of Antiresonant Frequencies

3.1. Experimental Antiresonances. The experimental antiresonances are identified from experimental point FRFs by "dip-picking" [22]. In this technique, the antiresonances are selected by picking the dips from the magnitude plot of a given FRF that have an associated change of +180° in the phase plot. This method is similar to the "peak picking" technique for resonant frequencies. Figure 3 shows an example of the anti-resonant picking procedure. When using this technique, the errors in the estimated antiresonances originate primarily from the frequency resolution of the FRFs. If the dips in the FRFs are difficult to determine, the antiresonances can be identified by a curve-fitting technique based on the rational fraction polynomial representation of the FRFs [38].

3.2. Numerical Antiresonances. For a lightly damped structure, the anti-resonant frequencies are nearly unaffected by damping, and, therefore, they can be obtained from the undamped system using only the stiffness and mass matrices.

By definition, the FRF matrix is the inverse of the dynamic stiffness matrix:

$$\mathbf{H}(\omega) = \left(\mathbf{K} - \omega^2 \mathbf{M}\right)^{-1} = \frac{\text{adj}\left(\mathbf{K} - \omega^2 \mathbf{M}\right)}{\det\left(\mathbf{K} - \omega^2 \mathbf{M}\right)}. \qquad (14)$$

The operators $\text{adj}(\cdot)$ and $\det(\cdot)$ indicate the adjoint and determinant, respectively. The anti-resonant frequencies correspond to the zeros of the FRFs. The zeros of the i, kth FRF are the values of ω for which the numerator of $H_{ik}(\omega)$ vanishes. The numerator of $H_{ik}(\omega)$ is the i, kth term of $\text{adj}(\mathbf{K} - \omega^2 \mathbf{M})$, which is given by $(-1)^{i+k} \det(\mathbf{K}_{i,k} - \omega^2 \mathbf{M}_{i,k})$. The subscripts i and k denote that the ith row and kth column have been deleted. As a consequence, the antiresonances of the i, kth FRF are the frequency values that satisfy

$$\det\left(\mathbf{K}_{i,k} - \omega^2 \mathbf{M}_{i,k}\right) = 0, \qquad (15)$$

which is equivalent to solving the eigenvalue problem:

$$\left(\mathbf{K}_{i,k} - \omega^2 \mathbf{M}_{i,k}\right) \mathbf{u} = 0. \qquad (16)$$

If $i = k$, (16) represents a physical system obtained by grounding the ith degree of freedom. Therefore, the antiresonant frequencies obtained from point FRFs ($i = k$) are equivalent to the resonant frequencies of the structure with the ith degree of freedom grounded.

Because we are working with point FRFs, matching the experimental and numerical anti-resonant frequencies is an easy task because each anti-resonant frequency lies between two resonant frequencies.

4. Eight-DOF Spring-Mass System

The structure shown in Figure 4 consists of an 8-DOF spring-mass system. The Los Alamos National Laboratory (LANL) designed and constructed this system to study the effectiveness of various vibration-based damage identification techniques [39].

The system consists of eight translating masses connected by springs, and each mass consists of a disc of aluminum with a diameter of 76.2 mm and a thickness of 25.4 mm. The masses slide on a highly polished steel rod and are fastened together with coil springs. The spring and mass locations are designated sequentially with the first located the closest to the shaker attachment. In the undamaged configuration, all springs are identical and have linear stiffness coefficients. Damage is simulated by replacing the fifth spring with a different spring with a lower stiffness (55% stiffness reduction). The acceleration is measured horizontally at each mass, giving a total of eight measured DOFs. The structure is excited randomly by an electrodynamic shaker. The responses are measured in the eight DOFs in the undamaged and damaged cases.

Seven anti-resonant frequencies are identified from the FRF corresponding to the first mass, which corresponds to
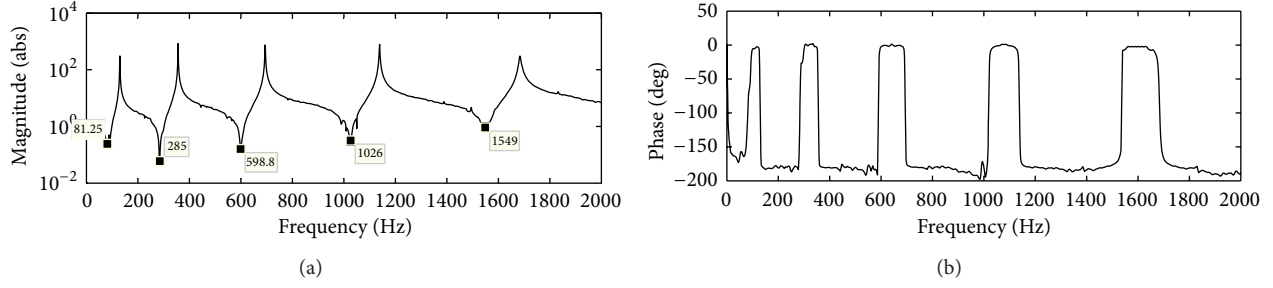
(a)

(b)

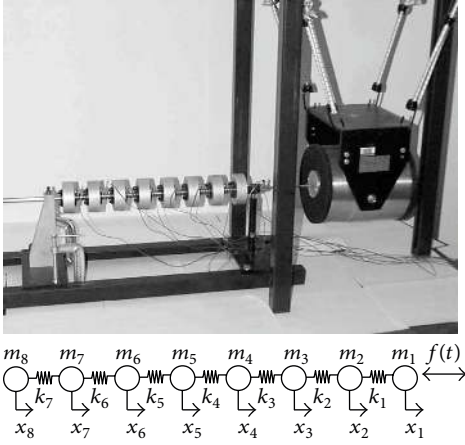Figure 3: Identification of experimental anti-resonant frequencies.

Figure 4: Experimental system with eight degrees of freedom.

Table 1: Experimental antiresonances.

| Antiresonant frequencies (Hz) | | Variation (%) |
Undamaged	Damaged	
12.31	11.66	−5.28
36.94	32.44	−12.18
59.61	59.8	0.32
79.96	71.74	−10.28
96.12	93.02	−3.23
109.44	106.86	−2.36
115.72	112.67	−2.64

Table 2: Experimental and numerical antiresonances.

| Antiresonant frequencies (Hz) | | Difference (%) |
Experimental	Numerical	
12.3	12.3	0.43
36.9	36.7	0.54
59.6	59.2	0.70
80.0	79.5	0.55
96.1	95.6	0.52
109.4	109.0	0.40
115.7	115.5	0.19

a point FRF. Given that the anti-resonant frequencies are independent, seven antiresonances should be sufficient to assess the stiffness of each of the seven springs.

Table 1 lists the anti-resonant frequencies identified in the undamaged and damaged cases. The second and fourth antiresonances provide the larger variation due to the experimental damage.

The numerical model is built in Matlab with springs and concentrated masses, and the initial parameters are set as follows:

(i) mass 1: 559.3 g (this mass is greater than the others because of the hardware required to attach the shaker)

(ii) masses 2 to 8: 419.4 g,

(iii) spring constants: $56.7 \, \text{kN} \, \text{m}^{-1}$.

Table 2 shows the numerical and experimental anti-resonant frequencies. The maximum difference between the experimental and numerical antiresonances is 0.7%. As a consequence, the numerical model provides a faithful representation of the experimental structure and can be used to train the neural network.

4.1. Construction of the Neural Network. The network contains seven inputs and seven outputs. The best parameters for the neural network are defined after performing a sensitivity analysis. The network is trained under different configurations, and the selected configuration is the one that provides the lowest validation error.

First, the effects of five different transfer functions used for the output layer are studied, and these transfer functions are formulated as follows: (a) hyperbolic tangent sigmoid, (b) logarithm sigmoid, (c) linear, (d) saturating linear, and (e) symmetric saturating linear. Figure 2 illustrates these functions. During the experimental runs, the remaining parameters of the network are characterized as follows: the hidden transfer function is a logarithm sigmoid, the number of hidden neurons is 30, the training data are polluted with 2% random noise, and the Levenberg-Marquardt algorithm is used to train the network. The network is trained 10 times with each of the output transfer functions. Table 3 summarizes the average validation performance obtained

Table 3: Network validation performance with different transfer functions.

Output transfer function	E_s	Hidden transfer function	E_s
Hyperbolic tangent sigmoid	0.0067	Hyperbolic tangent sigmoid	0.0045
Logarithm sigmoid	0.0134	Logarithm sigmoid	0.0044
Linear	0.0060	Linear	0.0130
Saturating linear	0.0127	Saturating linear	0.0050
Symmetric saturating linear	0.0046	Symmetric saturating linear	0.0047

Table 4: Network validation performance with different training algorithms.

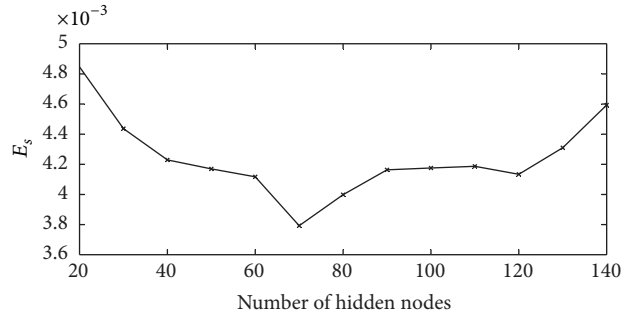| Training algorithms	Time (s)	E_s
Gradient descent	237	0.0467
Gradient descent with adaptive learning	9	0.0289
Scaled conjugate gradient	12	0.0158
Levenberg-Marquardt	15	0.0046

Figure 5: Network validation performance as a function of the number of hidden nodes.

Figure 6: Experimental mean sizing error for different levels of injected noise.

for each function. The symmetric saturating linear function provides the best performance.

Second, the performance of each of these five transfer functions in the hidden layer is investigated. The remaining network parameters are the symmetric saturating linear output transfer function, 30 hidden neurons, 2% noise, and the Levenberg-Marquardt training algorithm. The network is trained 10 times with each of these output transfer functions; Table 3 summarizes the average validation performance. The logarithm sigmoid function provides the best performance.

Next, the performance of four training algorithms is studied. Table 4 summarizes the average results for 10 runs. The gradient descent algorithm provides the worst performance in terms of training time and validation mean square error. An adaptive learning rate substantially improves the results; the training time is 26 times lower and the mean error is reduced by a factor of 1.6. The scale conjugate-gradient reaches a lower mean square error with a slightly higher training time. Nevertheless, the Levenberg-Marquardt algorithm provides the lowest mean square validation error by far, and its training time is nearly the same as that with the scale conjugate gradient algorithm.

Figure 5 shows the average validation error for different numbers of hidden nodes. Seventy nodes in the hidden layer give the lower validation error E_s. Less than 70 neurons are not sufficient to fit the data, whereas greater than 70 neurons overfit the neural network.

The last parameter corresponds to the level of noise in the training data. The selection of this parameter is related to the level of experimental noise. Hence, this parameter was defined by studying the performance of the trained network with the experimental data and different levels of perturbation noise. Figure 6 shows the mean sizing error (MSE) obtained when the network is tested with experimental data. This case corresponds to a 55% stiffness reduction in element 5. The MSE is the difference between the predicted damage level on each element and the actual damage level, as defined in (9). A data perturbation noise of 1.5% provides the minimum error in the predicted damage.
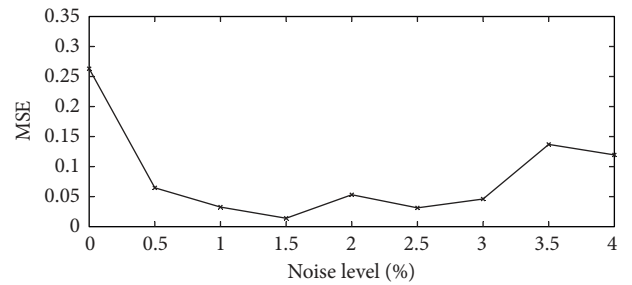
4.2. Network Validation.

The selected network parameters are

(i) network: three layers of multilayer perceptron,

(ii) number of input nodes: 7,

(iii) number of hidden nodes: 70,

(iv) number of output nodes: 7,

(v) transfer function in the hidden layer: Logarithm sigmoid,

(vi) transfer function in the output layer: Symmetric saturating linear,

(vii) training method: Levenberg-Marquardt.

The validation patterns verify the performance of the network, and these patterns are polluted with 1.5% noise. The resulting mean sizing error is MSE = 3.37%, and thus the network quantification accuracy is 97%. Figure 7 shows the damage missing error (DME) and false alarm error (FAE) divided by the damage levels. The results of the DME indicate
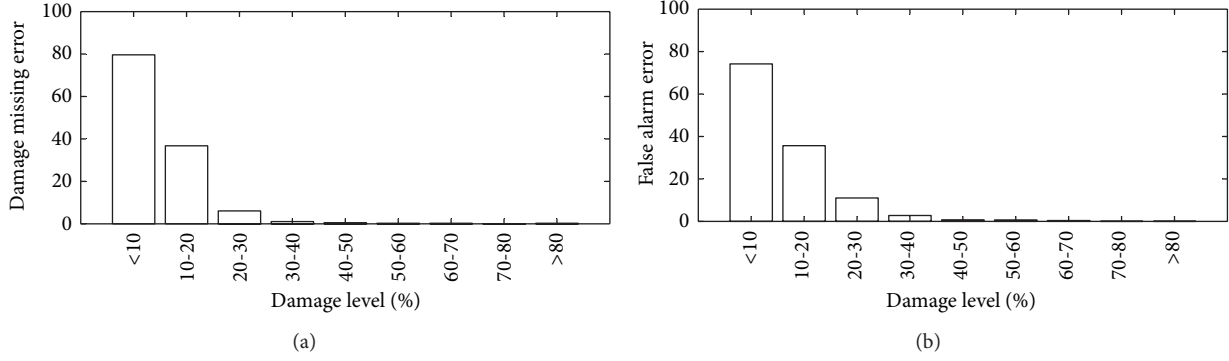
(a)

(b)

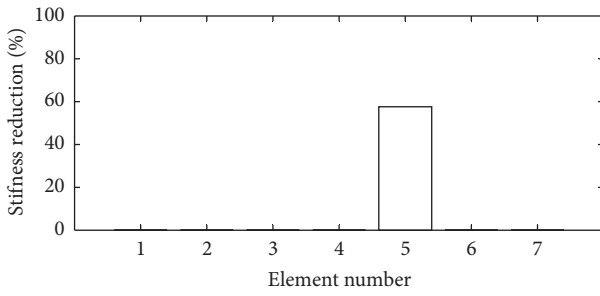FIGURE 7: Damage missing error (DME) and false alarm error (FAE) for different damage levels.

FIGURE 8: Assessment of experimental damage (55% stiffness reduction in element 5).

that the detection of damage with low severities is poor. In fact, 80% of the damage with severities lower than 10% and 40% of the damage with severities of 10–20% are not detected. The algorithm can detect damage with severities larger than 30% with confidence; it correctly detects 99.6% of the damage with severities over 30%. The false alarm results indicate that most of the damage detected with levels lower than 10% are false damage (near 75%). The amount of false damage detection is drastically reduced with an increment in the level of damage. In fact, 99.24% of the damage detected with levels over 30% is true damage.

Figure 8 shows the results of the damage detected using the experimental case. The damaged element is correctly assessed, the location is correct, and the quantification error is 5%.

5. Experimental Beam

The structure consists of a steel beam with a rectangular cross-section. The dimensions of the beam are length = 1 m and section = $25 \times 10 \text{ mm}^2$. As shown in Figure 9, soft springs suspend the structure to simulate a "free-free" boundary condition.

A hammer excites the beam at four points distributed along the beam, and an accelerometer measures the response at each excitation location. Both the excitation force and the measured responses lie in the horizontal direction. In this direction, the anti-resonant frequencies are more sensitive to

TABLE 5: Experimental and numerical antiresonances.

| | Antiresonant frequencies (Hz) | | Difference (%) |
	Experimental	Numerical	
	90.3	89	1.10
	293.4	289	1.35
FRF_1	606.3	604	0.40
	1032.8	1033	0.00
	1552.8	1576	1.51
	128.8	128	0.69
	260.9	258	1.00
FRF_2	583.1	586	0.56
	1112.8	1127	1.26
	1621.9	1650	1.70
	128.8	128	0.69
	260	258	0.66
FRF_3	584.4	586	0.34
	1113.8	1127	1.17
	1617.5	1650	1.98
	90.3	89	1.10
	292.9	289	1.35
FRF_4	606.1	604	0.40
	1032.8	1033	0.00
	1553.4	1576	1.51

FIGURE 9: Experimental beam.

the experimental damage. Five anti-resonant frequencies are identified at each of the four experimental FRFs.

The numerical model is built in Matlab with 2D beam elements. The model contains 20 beam elements and 40 degrees of freedom, as shown in Figure 10. The shadowed elements indicate possible locations of damage. Table 5 shows the experimental and numerical frequencies and their differences, and the maximum difference is 1.98%.

The structure is subjected to four different damage scenarios containing single and double cracks. Cracks are

TABLE 6: Damage cases introduced to the beam.

Case	Distance from the left end (mm)	Element number	Saw cut length (mm)	Distance from the left end (mm)	Element number	Saw cut length (mm)
1	315	7	5	—	—	—
2	635	13	10	—	—	—
3	303	7	5	640	13	10
4	360	8	5	810	17	15

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10|11|12|13|14|15|16|17|18|19|20|

FIGURE 10: Numerical model and element numbering.

FIGURE 11: Saw cut introduced to the beam.

FIGURE 12: Validation performance as a function of the number of hidden nodes.
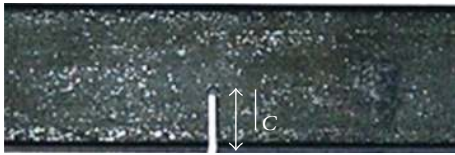
introduced to the structure by saw cuts of length l_c, as illustrated in Figure 11. Table 6 summarizes the different damage scenarios and indicates the distance from the left end to the cut, the corresponding element in the numerical model, and the cut length.

5.1. Network Validation. The network uses the same parameters selected for the first structure except for the number of input, hidden, and output nodes. The number of input nodes corresponds to the number of anti-resonant frequencies, which is 20. Elements 2 to 19 were defined as possible locations of damage, thus giving 18 output nodes (damage indices). The number of hidden nodes was defined after the sensitivity analysis shown in Figure 12. The minimum validation error is obtained with 80 hidden nodes.

Hence, the network parameters are

(i) network: three layers of multilayer perceptron,

(ii) number of input nodes: 20,

(iii) number of hidden nodes: 80,

(iv) number of output nodes: 18,

(v) transfer function in the hidden layer: logarithm sigmoid,

(vi) transfer function in the output layer: symmetric saturating linear,

(vii) training method: Levenberg-Marquardt.

The training and validation patterns were polluted with 1.5% random noise. The mean sizing error obtained for the validation patterns is MSE = 1.53%, and thus the network quantification accuracy is 98%. Figure 13 shows the damage missing error (DME) and false alarm error (FAE) separated by damage levels. The results of the DME indicate that the detection of damage with low severities is poor. In fact,
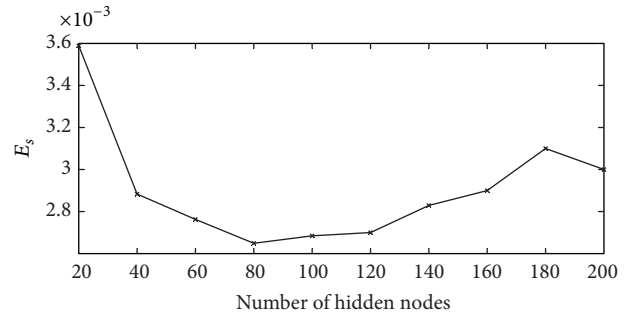
near 90% of the damage with severities lower than 10% and 70% of the damage with severities between 10 and 20% is not detected. The algorithm is able to detect damage with severities larger than 40% with confidence; it correctly detects 99.5% of the damage with severities over 40%. The false alarm results indicate that most of the damage detected with levels lower than 10% is false damage (near 80%). The amount of false damage detection decreases with an increment of the level of damage; 99% of the damage detected with levels over 40% is true damage.

Note that these results are slightly worse than the results obtained from the 8-DOF system. This result is explained by the fact that the anti-resonant frequencies are less sensitive to damage in the free beam than the 8-DOF structure, thus making it more difficult to detect small damage.

Figure 14 shows the results of the damage detected in the four experimental scenarios. An arrow indicates the actual damage location. In the first three cases, the damage is correctly located although false damage appears next to the actual locations. Nevertheless, the assumption that a crack only affects the stiffness of the corresponding element might not always be true. For instance, in the fourth case, the effect of the larger cut is represented more accurately by a stiffness reduction of two elements rather than one. In the last case, the network does not detect the small crack in element 8 because the effect of the larger cut hides the effect of the smaller cut.

6. Conclusions

This paper presents a new methodology designed to assess experimental damage using neural networks and anti-resonant frequencies. A multilayer perceptron network was trained with data obtained from a numerical model and
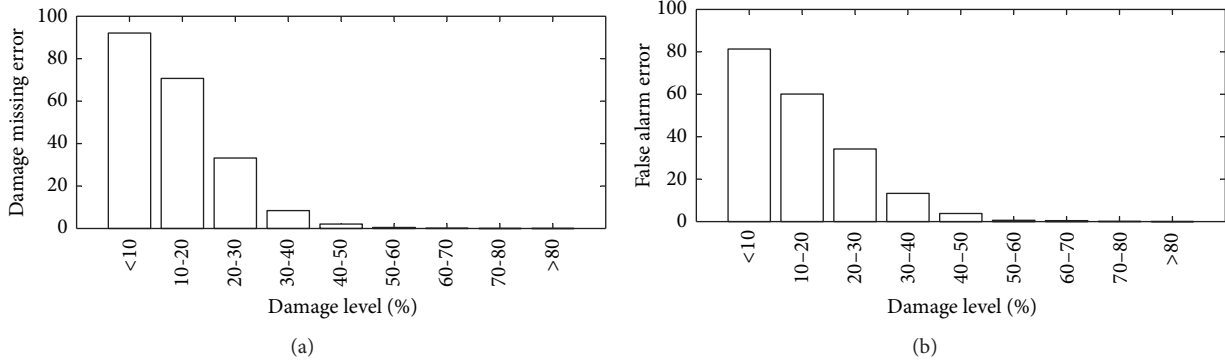
(a)

(b)

FIGURE 13: Damage missing error (DME) and false alarm error (FAE) for different damage levels.
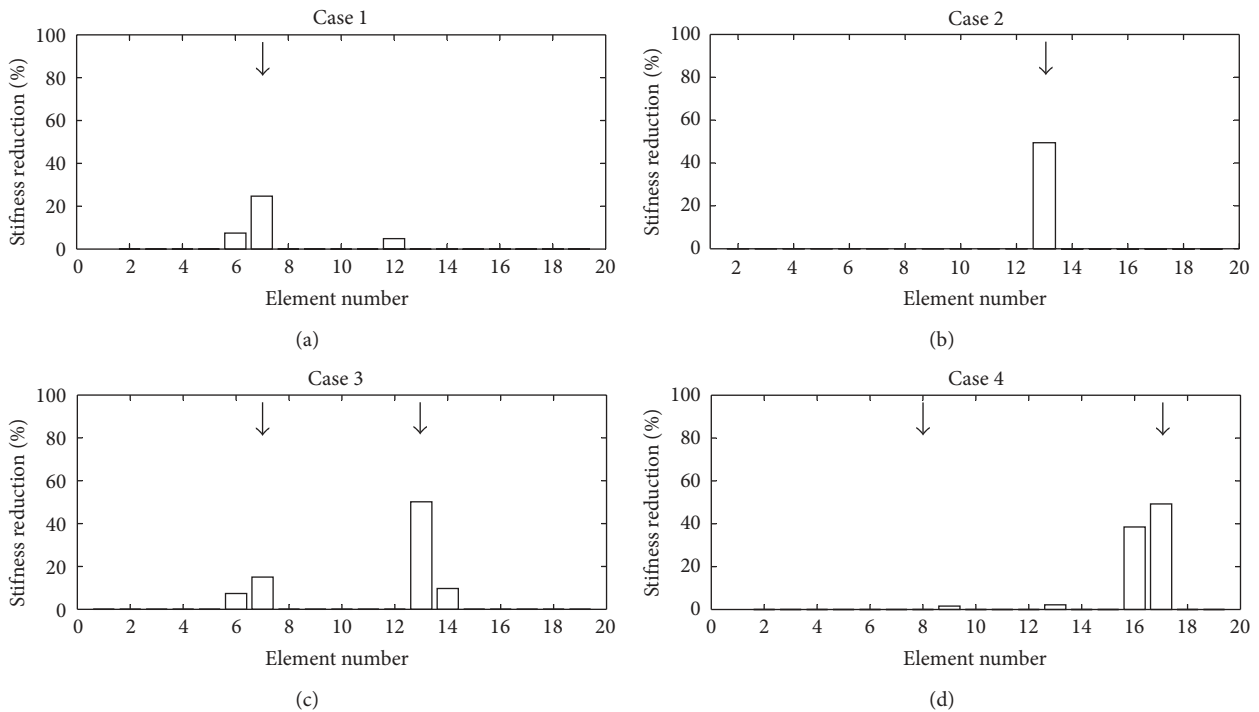

(a)

(b)

(c)

(d)

FIGURE 14: Experimental reduction of stiffness detected at each case.

tested with experimental data. The study addresses the setup of the neural network parameters and provides guidelines for their selection in similar damage assessment problems. Two experimental cases verify the algorithm: an 8-DOF mass-spring system and a beam with multiple damage scenarios.

In both structures, the algorithm is successful in assessing the experimental damage. The damage detected corresponds closely with the experimental damage in all cases. These results show that it is possible to locate and quantify structural damage using only the anti-resonant information obtained from point frequency response functions. Hence, anti-resonant frequencies are an attractive feature for use in damage assessment.

This study shows that it is possible to assess experimental damage in real time in two different structures, and, hence,

this work demonstrates the possibility of continuous monitoring of the structural condition. Nevertheless, according to the validation results, the algorithm displays a minimum level of damage that can be assessed with confidence, which is 30% in the case of an 8-DOF system and 40% for the beam. These results can be improved using a larger training database but these changes imply larger training times.

It is important to note that, in this study, anti-resonant frequencies were identified with the same precision as resonant frequencies, which is true for lightly damped structures with low levels of experimental noise. If this is not the case, anti-resonant frequencies might be difficult to identify and the damage identification algorithm will be affected. An area of further research is the identification of anti-resonant frequencies in structures with high levels of damping or high

levels of experimental noise. Another area of further research is the use of anti-resonant frequencies obtained from the transfer frequency response functions, which are more sensitive to structural changes but require additional attention, especially, when pairing experimental and numerical anti-resonant frequencies.

Conflict of Interests

The authors declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] V. Meruane and W. Heylen, "Damage detection with parallel genetic algorithms and operational modes," *Structural Health Monitoring*, vol. 9, no. 6, pp. 481–496, 2010.

[2] V. Meruane and W. Heylen, "An hybrid real genetic algorithm to detect structural damage using modal properties," *Mechanical Systems and Signal Processing*, vol. 25, no. 5, pp. 1559–1573, 2011.

[3] R. Perera and R. Torres, "Structural damage detection via modal data with genetic algorithms," *Journal of Structural Engineering*, vol. 132, no. 9, pp. 1491–1501, 2006.

[4] B. Kouchmeshky, W. Aquino, J. C. Bongard, and H. Lipson, "Co-evolutionary algorithm for structural damage identification using minimal physical testing," *International Journal for Numerical Methods in Engineering*, vol. 69, no. 5, pp. 1085–1107, 2007.

[5] A. Teughels, G. de Roeck, and J. A. K. Suykens, "Global optimization by coupled local minimizers and its application to FE model updating," *Computers and Structures*, vol. 81, no. 24-25, pp. 2337–2351, 2003.

[6] C. Valdés-González and J. González-Pérez, "Identification of structural damage in a vehicular bridge using artificial neural networks," *Structural Health Monitoring*, vol. 10, no. 1, pp. 33–48, 2011.

[7] B. Sahoo and D. Maity, "Damage assessment of structures using hybrid neuro-genetic algorithm," *Applied Soft Computing Journal*, vol. 7, no. 1, pp. 89–104, 2007.

[8] S. Arangio and J. L. Beck, "Bayesian neural networks for bridge integrity assessment," *Structural Control and Health Monitoring*, vol. 19, no. 1, pp. 3–21, 2012.

[9] A. Messina, A. Jones, and E. J. Williams, "Damage detection and localisation using natural frequency changes," in *Proceedings of the 1st Conference on Identification in Engineering Systems*, pp. 67–76, Swansea, UK, 1996.

[10] R. M. Lim and D. J. Ewins, "Model updating using FRF data," in *Proceedings of the 15th International Seminar on Modal Analysis*, pp. 141–163, Leuven, Belgium, 1990.

[11] N. Stubbs, J. T. Kim, and K. Topole, "An efficient and robust algorithm for damage localization in offshore platforms," in *Proceedings of the ASCE 10th Structures Congress*, pp. 543–546, San Antonio, Tex, USA, 1992.

[12] S. F. Masri, A. W. Smyth, A. G. Chassiakos, T. K. Caughey, and N. F. Hunter, "Application of neural networks for detection of changes in nonlinear systems," *Journal of Engineering Mechanics*, vol. 126, no. 7, pp. 666–676, 2000.

[13] P. Castellini and G. M. Revel, "Experimental technique for structural diagnostic based on laser vibrometry and neural networks," *Shock and Vibration*, vol. 7, no. 6, pp. 381–397, 2000.

[14] C. Zang and M. Imregun, "Structural damage detection using artificial neural networks and measured FRF data reduced via principal component projection," *Journal of Sound and Vibration*, vol. 242, no. 5, pp. 813–827, 2001.

[15] X. Fang, H. Luo, and J. Tang, "Structural damage detection using neural network with learning rate improvement," *Computers and Structures*, vol. 83, no. 25-26, pp. 2150–2161, 2005.

[16] O. S. Salawu, "Detection of structural damage through changes in frequency: a review," *Engineering Structures*, vol. 19, no. 9, pp. 718–723, 1997.

[17] A. H. Gandomi, M. G. Sahab, A. Rahaei, and M. S. Gorji, "Development in mode shape-based structural fault identification technique," *World Applied Sciences Journal*, vol. 5, no. 1, pp. 29–38, 2008.

[18] J. L. Zapico, M. P. González, and K. Worden, "Damage assessment using neural networks," *Mechanical Systems and Signal Processing*, vol. 17, no. 1, pp. 119–125, 2003.

[19] C. B. Yun, J. H. Yi, and E. Y. Bahng, "Joint damage assessment of framed structures using a neural networks technique," *Engineering Structures*, vol. 23, no. 5, pp. 425–435, 2001.

[20] A. K. Pandey, M. Biswas, and M. M. Samman, "Damage detection from changes in curvature mode shapes," *Journal of Sound and Vibration*, vol. 145, no. 2, pp. 321–332, 1991.

[21] M. Sahin and R. A. Shenoi, "Vibration-based damage identification in beam-like composite laminates by using artificial neural networks," *Proceedings of the Institution of Mechanical Engineers C*, vol. 217, no. 6, pp. 661–676, 2003.

[22] W. D'Ambrogio and A. Fregolent, "Use of antiresonances for robust model updating," *Journal of Sound and Vibration*, vol. 236, no. 2, pp. 227–243, 2000.

[23] E. J. Williams and A. Messina, "Applications of the multiple damage location assurance criterion," *Key Engineering Materials*, vol. 167, pp. 256–264, 1999.

[24] M. Dilena and A. Morassi, "The use of antiresonances for crack detection in beams," *Journal of Sound and Vibration*, vol. 276, no. 1-2, pp. 195–214, 2004.

[25] Y. Bamnios, E. Douka, and A. Trochidis, "Crack identification in beam structures using mechanical impedance," *Journal of Sound and Vibration*, vol. 256, no. 2, pp. 287–297, 2002.

[26] T. Inada, Y. Shimamura, A. Todoroki, and H. Kobayashi, "Development of the two-step delamination identification method by resonant and anti-resonant frequency changes," *Key Engineering Materials*, vol. 270–273, no. 3, pp. 1852–1858, 2004.

[27] D. Wang and H. Zhu, "Wave propagation based multi-crack identification in beam structures through anti-resonances information," *Key Engineering Materials*, vol. 293-294, pp. 557–564, 2005.

[28] V. Meruane and W. Heylen, "Structural damage assessment with antiresonances versus mode shapes using parallel genetic algorithms," *Structural Control and Health Monitoring*, vol. 18, no. 8, pp. 825–839, 2011.

[29] M. A. Arbib, *The Handbook of Brain Theory and Neural Networks*, Bradford Book, 2003.

[30] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[31] M. Beale and H. Demuth, *Neural Network Toolbox*, Version 7.0.3 Release, The MathWorks, 2012.

[32] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[33] M. I. Friswell, J. E. T. Penny, and S. D. Garvey, "A combined genetic and eigensensitivity algorithm for the location of damage in structures," *Computers and Structures*, vol. 69, no. 5, pp. 547–556, 1998.

[34] J. J. Lee, J. W. Lee, J. H. Yi, C. B. Yun, and H. Y. Jung, "Neural networks-based damage detection for bridges considering errors in baseline finite element models," *Journal of Sound and Vibration*, vol. 280, no. 3-5, pp. 555–578, 2005.

[35] K. D. Hjelmstad and S. Shin, "Damage detection and assessment of structures from static response," *Journal of Engineering Mechanics*, vol. 123, no. 6, pp. 568–576, 1997.

[36] S. V. Barai and P. C. Pandey, "Vibration signature analysis using artificial neural networks," *Journal of Computing in Civil Engineering*, vol. 9, no. 4, pp. 259–265, 1995.

[37] J. E. Mottershead, "On the zeros of structural frequency response functions and their sensitivities," *Mechanical Systems and Signal Processing*, vol. 12, no. 5, pp. 591–597, 1998.

[38] L. Prechelt, "Early stopping—but when?" in *Neural Networks: Tricks of the Trade*, vol. 1524 of *Lecture Notes in Computer Science*, pp. 55–69, Springer, New York, NY, USA, 1998.

[39] M. H. Richardson and D. L. Formenti, "Parameter estimation from frequency response measurements using rational fraction polynomials," in *Proceedings of the 1st International Modal Analysis Conference*, pp. 167–181, 1982.

Journal of
Engineering

The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

International Journal of
Distributed
Sensor Networks

Advances in
Civil Engineering

Journal of
Control Science
and Engineering

Journal of
Robotics

Journal of
Electrical and Computer
Engineering

Advances in
OptoElectronics

VLSI Design

International Journal of
Navigation and
Observation

Modelling &
Simulation
in Engineering

International Journal of
Aerospace
Engineering

International Journal of
Chemical Engineering

International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components

Shock and Vibration

Advances in
Acoustics and Vibration

Hindawi

Submit your manuscripts at
http://www.hindawi.com