Online Sequential Extreme Learning Machine for Vibration-Based Damage Assessment Using Transmissibility Data

V. Meruane[1]

Abstract: Traditional vibration-based damage assessment approaches include the use of feed-forward neural networks. However, the slow learning speed of these networks and the large number of parameters that need to be tuned have been a major bottleneck in their application. This paper proposes to use an emergent learning algorithm called the online sequential extreme learning machine (OS-ELM) algorithm. This algorithm provides good generalization at fast learning speeds, allows data to be learned one by one or block by block, and the only parameter that needs to be tuned is the number of hidden nodes. A single-hidden-layer network is trained to detect, locate, and quantify structural damage using data derived from transmissibility measurements. Two experimental cases are presented to illustrate the approach: an eight-degree-of-freedom (DOF) mass-spring system and a beam under multiple damage scenarios. To demonstrate the potential of the proposed algorithm over existing ones, the obtained results are compared with those of a model updating approach based on parallel genetic algorithms. **DOI: 10.1061/(ASCE)CP.1943-5487.0000517.** *© 2015 American Society of Civil Engineers.*

Introduction

Vibration-based damage assessment algorithms must detect and characterize damage at the earliest possible stage and estimate how much time remains before maintenance is required, a structure fails, or the structure is no longer usable. Damage assessment has tremendous potential in providing life safety and/or economic benefits by reducing maintenance costs and enhancing structure safety and reliability. The main problem of vibration-based damage assessment is to ascertain the presence, location, and severity of structural damage given a structure's dynamic characteristics. The most successful applications of vibration-based damage assessment are model updating methods based on global optimization algorithms (Meruane and Heylen 2010, 2011a, b; Perera and Torres 2006; Kouchmeshky et al. 2007; Teughels et al. 2003; Khaji 2014). Model updating is an inverse method used to identify the uncertain parameters of a numerical model and is usually formulated as an inverse optimization problem. In inverse damage detection, the algorithm uses the differences between models of a structure updated before and after the presence of damage to localize and determine the extent of damage. The basic assumption is that damage can be directly related to a decrease in stiffness in the structure. Nevertheless, these algorithms are exceedingly slow, and the damage assessment process is achieved through a costly and time-consuming inverse process. These limitations hinder the application of real-time health monitoring, in which the structure is continuously monitored and the damage is assessed as they develop. Neural networks have been introduced as an alternative to model updating in damage assessment (Hakim and Abdul Razak 2014; González-Pérez and Valdes-Gonzalez 2011; Sahoo and Maity 2007; Arangio and Beck 2012). Although it takes time to train a neural network, after it has been trained it can potentially detect, locate, and quantify structural damage in a short period. Hence, it can be used for real-time damage assessment. Damage detection by means of neural networks has the advantage of being a general approach. Unlike many other damage detection methods, which are often developed for specific quantities (Messina et al. 1996; Lim and Ewins 1990; Stubbs et al. 1992), neural networks can in principle be applied to any correlation coefficient that is sensitive to damage. Additionally, neural networks can be applied to assess structures that exhibit a nonlinear response (Masri et al. 2000).

An artificial neural network is a data-processing algorithm that tries to emulate the processing scheme of the human brain (Arbib 2003). There are different types of network architectures, among which multilayer feed-forward networks are the most frequently used. The successful application of a feed-forward network depends on the representation and the learning algorithm used. Nevertheless, the selection of the best network is problem dependent and is usually determined by trial and error (Barai and Pandey 1995; Meruane and Mahu 2014). Sahoo and Maity (2007) used a genetic algorithm (GA) to automate the trial-and-error process. The network parameters (e.g., number of neurons and learning rate) were set as the variables of an optimization problem that was handled by a GA. Sahoo and Maity used a Multilayer Perceptron (MLP) network with two hidden layers trained by a back-propagation algorithm. Fang et al. (2005) explored the use of a tunable steepest descent (TSD) algorithm that dynamically adjusted the learning speed during the training process. They demonstrated that this methodology significantly increases the training speed while keeping the learning stable. Over the past decades, researchers have used gradient-based descent methods as learning algorithms for neural networks. However, it is clear that these algorithms are generally slow and may easily converge to local minimums. The slow learning speed and the large number of parameters that need to be tuned have been a major bottleneck in the application of feed-forward networks. Huang et al. (2004) proposed a new learning algorithm called the extreme learning machine (ELM) algorithm, which is suitable for single-layer feed-forward networks. This algorithm provides good generalization at fast learning speeds, and

[1] Assistant Professor, Dept. of Mechanical Engineering, Universidad de Chile, Beauchef 851, Santiago 8370456, Chile. E-mail: vmeruane@ing.uchile.cl

Note. This manuscript was submitted on April 16, 2014; approved on May 27, 2015; published online on July 21, 2015. Discussion period open until December 21, 2015; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801/04015042(10)/$25.00.

the only parameter that needs to be tuned is the number of hidden nodes. Since it was first introduced in 2004, the ELM algorithm has attracted the attention of increasing numbers of researchers (Huang et al. 2011).

In many applications, training data may arrive one by one or block by block. Batch training is the traditional approach; in this case, all data—old and new—are used during training. This training scheme results in a very time-consuming learning process. On the other hand, sequential training algorithms learn only newly arrived data instead of all past data. Many sequential learning schemes have been proposed over the past years, including the resource allocation (RAN) scheme (Platt 1991) and its extensions. Nevertheless, these algorithms are still slow in some applications and can only handle data one by one, not block by block. In addition, most of these algorithms feature several parameters that need to be specified by users. Tuning those parameters is very time consuming. To solve these problems, Liang et al. (2006) proposed the online sequential extreme learning machine (OS-ELM) algorithm. The OS-ELM algorithm is a sequential implementation of the ELM algorithm, and besides being very fast, it has the following characteristics:

- The training observations are learned sequentially block by block with varying or fixed block lengths;
- Only newly arrived data are learned;
- A block of data is discarded as soon as the algorithm learns it;
- The learning algorithm has no prior knowledge regarding how many training observations will be presented; and
- The only parameter that needs to be tuned is the number of hidden nodes.

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 regarding the optimum one is a testament to the difficulty of the problem. This measure can be constructed in the time, frequency, or modal domain. The last two are the most popular domains used.

The idea of using frequency response functions (FRFs) directly to train neural networks has attracted many researchers. Among all 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 too large for neural network applications. The direct use of FRFs will lead to networks with a large number of input variables and connections, making them impractical. Hence, it becomes necessary to extract features from the FRFs and use these features as inputs to neural networks. Castellini and Revel (2000) presented an algorithm for detecting and locating structural damage based on laser vibrometry measurements and a neural network for data processing. They used features extracted from FRFs as inputs to the neural network. Castellini and Revel 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 (2001) applied a principal component analysis (PCA) technique to the measured FRFs. The output of the neural network was the actual state of the structure: undamaged or damaged. The algorithm was able to distinguish between undamaged and damaged cases. Fang et al. (2005) selected key spectral points around the resonance frequencies in the FRF data. These selected points were the inputs of a neural network, whereas the outputs were the stiffness reduction factors. The algorithm showed high accuracy in identifying damage to a simulated cantilever beam under different damage scenarios.

Input data can be further reduced if modal analysis is performed first. Thus, the input variables are the modal properties of a structure. Natural frequencies and mode shapes are the most frequently used properties. Recently, researchers proposed the use of antiresonant frequencies as an alternative to mode shapes (D'Ambrogio and Fregolent 2000). Antiresonant frequencies correspond to the zeroes (dips) of FRFs; these features are an attractive alternative because they can be determined more easily and with less error than mode shapes while still providing the same information. Williams and Messina (1999) introduced antiresonant frequencies from point FRFs into the multiple damage location assurance criterion (MDLAC) algorithm. They concluded that the incorporation of antiresonant data improves the accuracy of damage predictions. Dilena and Morassi (2004) studied the problem of crack detection in beams using resonant and antiresonant frequencies. They found that the use of antiresonant frequencies helps to avoid the nonuniqueness of damage locations that occurs when only natural frequencies are used. However, they also found that experimental noise and modeling errors are usually amplified when antiresonant frequencies are implemented. Bamnios et al. (2002) proposed a scheme for crack location in beams. They used the shift in the first antiresonant frequency versus the measuring position to detect and locate cracks. Bamnios et al. reported that the method could be used to locate cracks roughly. Other methods can then be employed to determine the crack characteristics more precisely. Changes in resonant and antiresonant frequencies were used by Inada et al. (2004) to locate and quantify the delamination of a composite beam. Inada et al. implemented a two-step procedure: first, the delamination domain was identified from the antiresonant changes, and the location and size were then defined using natural frequency changes. The method was effective in identifying delamination locations and sizes. Wang and Zhu (2005) proposed another method for identifying cracks in beams. The method makes use of resonant and antiresonant frequencies from point FRFs. The methodology is similar to the one proposed by Bamnios et al. (2002): the shift in the first antiresonant frequency versus the driving point location is used to locate damage. Wang and Zhu validated the method using the numerical example of a simply supported beam with three cracks. Their method was able to predict the locations of cracks accurately, though it only estimated crack size qualitatively. Meruane and Heylen (2011b) demonstrated that antiresonant frequencies are a good alternative to mode shapes in damage assessment. However, they stated that further research is required for the identification of experimental antiresonant frequencies and for the matching of experimental and numerical antiresonant frequencies. Meruane (2013) presented a model updating and damage assessment algorithm that uses antiresonant frequencies derived from transmissibility data. Antiresonant frequencies correspond to the dips in FRFs and consequently to the dips and peaks in transmissibility functions. Hence, it is possible to identify antiresonant frequencies using transmissibility information. The main advantage of using transmissibility functions and not frequency response functions is that in the first case it is not necessary to measure the excitation force, which can be very challenging for structures in service. Antiresonant frequencies are an attractive alternative to mode shapes as input values of neural networks because the number of input variables is lower and the values are less contaminated by noise and still offer the same information. In addition, antiresonant frequencies can be identified from transmissibility measurements.

The primary contribution of this research is the development of a real-time damage assessment algorithm using an OS-ELM algorithm and antiresonant frequencies. Antiresonant frequencies are identified from transmissibility functions using the algorithm presented in Meruane (2013). The performance of the proposed methodology is validated by considering two experimental structures: an eight-degree-of-freedom (DOF) mass-spring system and

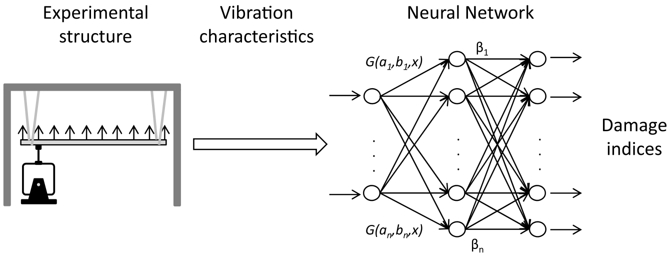

Fig. 1. Damage assessment with a single-hidden-layer feed-forward network

a beam under multiple damage scenarios. To demonstrate the potential of the proposed algorithm over existing ones, the obtained results are compared with those of a model updating approach based on parallel genetic algorithms.

OS-ELM for Damage Assessment

Fig. 1 illustrates the principle of a damage assessment algorithm using a single-hidden-layer feed-forward network (SLFN), which is the one used by the OS-ELM algorithm. The vibration characteristics of the structure, which in this case correspond to antiresonant frequencies, act as the inputs to the neural network, and the outputs are the damage indices of each element in the structure.

The output vector, $\mathbf{t}$, of a SLFN with n hidden nodes is given by

$$\mathbf{t} = \sum_{i=1}^{n} \beta_i g(\mathbf{a}_i \cdot \mathbf{x} + b_i), \quad \mathbf{t} \in \mathbf{R}^m, \mathbf{x} \in \mathbf{R}^n, \quad \mathbf{a}_i \in \mathbf{R}^n \tag{1}$$

where $\boldsymbol{a}_i$ = weight vector connecting the input layer to the ith hidden node; b_i = bias of the ith hidden node; β_i = weight connecting the ith hidden node to the output nodes; and $g(\mathbf{a}_i, b_i, \mathbf{x})$ = activation function of the ith hidden node with respect to the input vector $\mathbf{x}$.

Huang et al. (2004) demonstrated that one can randomly assign the parameters of the hidden layer, $\mathbf{a}_i$ and $b_{i,}$ and thus analytically compute the output weights as follows:

$$\boldsymbol{\beta} = \mathbf{H}^{+}\mathbf{T} \tag{2}$$

where

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{a}_{11} \cdot x_1 + b_1) & \cdots & g(\mathbf{a}_{1n} \cdot x_1 + b_n) \\ \vdots & \ddots & \vdots \\ g(\mathbf{a}_{N1} \cdot x_N + b_1) & \cdots & g(\mathbf{a}_{Nn} \cdot x_N + b_n) \end{bmatrix}_{N \times n},$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_n^T \end{bmatrix}_{n \times m} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \tag{3}$$

where N = number of training patterns; n = number hidden nodes; m = number of outputs; and $\mathbf{t}_i$ = target output for the input vector $\mathbf{x}_i$. $\mathbf{H}^{+}$ is the pseudoinverse of the matrix $\mathbf{H}$. Hence, all of the network parameters are defined without needing an iterative learning algorithm.

The set of input and output vectors $[\{\mathbf{x}_1, \mathbf{x}_1, \cdots, \mathbf{x}_N\}, \{\mathbf{t}_1, \mathbf{t}_1, \cdots, \mathbf{t}_N\}]$ corresponds to the training database that contains the vibration characteristic $(\mathbf{x}_i)$ of the structure for different damage scenarios $(\mathbf{t}_i)$. A disadvantage of neural networks is the need for large training sets; it is 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 (2000) 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 depends on the accuracy of the numerical model. There are two approaches 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 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 (Friswell et al. 1998). Another difficulty of using numerical data to build the training database is that 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 (Hjelmstad and Shin 1997).

Sequential implementation of the least-squares solution of Eq. (2) results in the OS-ELM algorithm, which uses a recursive least-squares algorithm. The OS-ELM algorithm is implemented in two phases, namely initialization and sequential learning. During initialization, an initial matrix $\mathbf{H}_0$ is constructed for use in the learning phase. The number of training samples used to construct $\mathbf{H}_0$ should be at least the number of hidden nodes. Sequential learning commences after initialization; in this phase, data can be learned one by one or block by block (with fixed or varying size). Once the data are learned, they are no longer required and discarded. The OS-ELM algorithm, as described by Liang et al. (2006), is implemented as described as follows:

Initialize the learning using a small block of initial training data $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^{N_0}$ from the given training set, $n \leq N_0 < N$:

1. Randomly generate the hidden nodes parameters $(\mathbf{a}_i, b_i), I = 1, \ldots, n$;
2. Calculate the initial hidden layer output matrix $\mathbf{H}_0$

$$\mathbf{H}_0 = \begin{bmatrix} g(\mathbf{a}_{11} \cdot x_1 + b_1) & \cdots & g(\mathbf{a}_{1n} \cdot x_1 + b_n) \\ \vdots & \ddots & \vdots \\ g(\mathbf{a}_{N_0 1} \cdot x_{N_0} + b_1) & \cdots & g(\mathbf{a}_{N_0 n} \cdot x_{N_0} + b_n) \end{bmatrix}_{N_0 \times n} \tag{4}$$

3. Estimate the initial output weight $\boldsymbol{\beta}_0 = \mathbf{H}_0^{+}\mathbf{T}_0$, where $\mathbf{T}_0 = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_0}]^T$;
4. Set $k = 0$;
5. Select the $(k+1)$th block of new observations $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=\sum_{j=0}^{k} N_j + 1}^{\sum_{j=0}^{k+1} N_j}$, where N_j denotes the number of observations in the jth block;

6. Calculate the partial hidden layer output matrix $\mathbf{H}_{k+1}$

$$\mathbf{H}_{k+1} = \begin{bmatrix} g\left(\mathbf{a}_{\sum_{j=0}^{k} N_j+1,1} \cdot x_{\sum_{j=0}^{k} N_j+1} + b_1\right) & \cdots & g\left(\mathbf{a}_{\sum_{j=0}^{k} N_j+1,n} \cdot x_{\sum_{j=0}^{k} N_j+1} + b_n\right) \\ \vdots & \ddots & \vdots \\ g\left(\mathbf{a}_{\sum_{j=0}^{k+1} N_j,1} \cdot x_{\sum_{j=0}^{k+1} N_j} + b_1\right) & \cdots & g\left(\mathbf{a}_{\sum_{j=0}^{k+1} N_j,n} \cdot x_{\sum_{j=0}^{k+1} N_j} + b_n\right) \end{bmatrix}_{N_{k+1}\times n} \quad (5)$$

set $\mathbf{T}_{k+1} = \left[\mathbf{t}_{\sum_{j=0}^{k} N_j+1}, \ldots, \mathbf{t}_{\sum_{j=0}^{k+1} N_j}\right]^T$.

7. Calculate the output weight $\boldsymbol{\beta}^{(k+1)}$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{P}_k \mathbf{H}_{k+1}^T (\mathbf{I} + \mathbf{H}_{k+1} \mathbf{P}_k \mathbf{H}_{k+1}^T)^{-1} \mathbf{H}_{k+1} \mathbf{P}_k$$
$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \mathbf{P}_{k+1} \mathbf{H}_{k+1}^T [\mathbf{T}_{k+1} - \mathbf{H}_{k+1} \boldsymbol{\beta}^{(k)}] \quad (6)$$

8. Set $k = k + 1$. Go to Step 5.

Once the network has been trained and the weights $\mathbf{a}_i$, b_i, and β_i are known, the damage state, $\mathbf{t}$, of the structure associated to the vector of vibration characteristics $\mathbf{x}$ is determined by Eq. (1).

Identification of Antiresonant Frequencies

Experimental Antiresonant Frequencies

FRFs correspond to the ratio in the frequency domain between the response measured at a certain point and the excitation force (F_k)

$$H_{ik}(\omega) = \frac{X_{ik}(\omega)}{F_k(\omega)} \quad (7)$$

$H_{ik}(\omega)$ is the FRF between the output DOF i and the input DOF k, when all the remaining DOFs have zero inputs. Transmissibility functions, on the other hand, are the ratio in the frequency domain between two measured outputs

$$T_{ij}^k(\omega) = \frac{X_{ik}(\omega)}{X_{jk}(\omega)} \quad (8)$$

where $X_{ik}(\omega)$ and $X_{jk}(\omega)$ = output responses at DOFs i and j due to an input force at DOF k. In the case of a single force, the transmissibility functions only depend on the location of the force but not on the amplitude. Hence, the estimation of transmissibility functions does not involve the measurement of the excitation force. Assuming a single input force at DOF k, transmissibilities are related to the FRFs as

$$T_{ij}^k(\omega) = \frac{X_{ik}(\omega)}{X_{jk}(\omega)} = \frac{H_{ik}(\omega)F_k(\omega)}{H_{jk}(\omega)F_k(\omega)} = \frac{H_{ik}(\omega)}{H_{jk}(\omega)} \quad (9)$$

Antiresonant frequencies correspond to the dips in FRFs, and consequently to the dips and peaks in transmissibility functions. Hence, it is possible to identify antiresonant frequencies using transmissibility information. Here, the experimental antiresonant frequencies are identified from transmissibility functions using the algorithm presented in Meruane (2013). To determine the antiresonant frequencies of the j, kth FRF, more accurately, the algorithm uses the summation of the amplitudes of all measured transmissibility functions whose denominators are the response in j. This function is calculated as

$$T_j^k(\omega) = \sum_{p=1,p\neq j}^{N} |\mathrm{Re}[T_{pj}^k(\omega)]| + j \sum_{p=1,p\neq j}^{N} |\mathrm{Im}[T_{pj}^k(\omega)]| \quad (10)$$

where N = number of responses measured. The superscript k indicates the location of the excitation and the subscript j indicates location of the response. Summing the transmissibility functions helps to reduce the noise to signal ratio, and hence to increase the accuracy of the detected antiresonant frequencies. The resulting function, $T_j^k(\omega)$, contains only peaks at the antiresonant frequencies of the j, kth FRF.

By curve fitting the absolute value of $T_j^k(\omega)$ to a rational fraction form and solving the roots of the denominator, the antiresonant frequencies of the j, kth FRF are identified. As shown in Fig. 2, a stabilization diagram assists in separating physical poles from mathematical poles. The order on the left Y-axis corresponds to the order of the fitted orthogonal polynomials. The xes and circles correspond to the roots of the denominator polynomial; a circle indicates a stable root. The following stabilization criterion is used: 1% for frequency stability and 5% for damping stability. The algorithm automatically identifies an antiresonant frequency if there are more than 10 stable roots in a row.

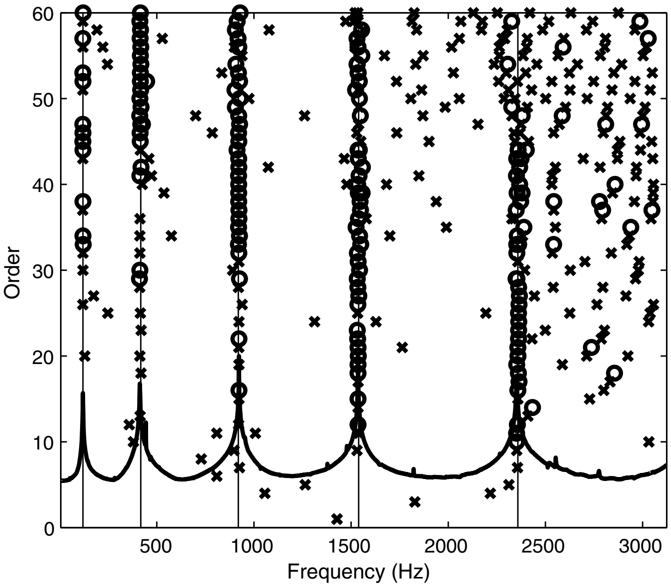

Fig. 2. Stabilization diagram used in the identification of antiresonant frequencies

Numerical Antiresonant Frequencies

For a lightly damped structure, the antiresonant 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

$$\mathbf{H}(\omega) = (\mathbf{K} - \omega^2\mathbf{M})^{-1} = \frac{\mathrm{adj}(\mathbf{K} - \omega^2\mathbf{M})}{\det(\mathbf{K} - \omega^2\mathbf{M})} \tag{11}$$

The operators adj(.) and det(.) indicate the adjoint and determinant, respectively. The antiresonant frequencies correspond to the zeroes of the FRFs. The zeroes 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 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, k denote that the ith row and kth column have been deleted. Consequently, the antiresonant frequencies of the i, kth FRF are the frequency values that satisfy

$$\det(\mathbf{K}_{i,k} - \omega^2\mathbf{M}_{i,k}) = 0 \tag{12}$$

which is equivalent to solving the eigenvalue problem

$$(\mathbf{K}_{i,k} - \omega^2\mathbf{M}_{i,k})\mathbf{u} = 0 \tag{13}$$

If $i = k$, Eq. (13) 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.

If $i \neq k$, Eq. (13) does not represent any physical system and some of the eigenvalues may be negative or complex, these values must not be considered as antiresonant frequencies.

Construction of the Network

The number of hidden nodes is defined after a sensitivity analysis for each application case. The network uses a logarithmic sigmoid activation function, even though the results demonstrate that the network performance is nearly insensitive to the activation function selected.

Input Vector

As proposed by Lee et al. (2005), 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 finite-element model.

Therefore, the input vector corresponds to the experimental changes in the antiresonant frequencies with respect to the intact case

$$\mathbf{x} = \left\{ \frac{\omega^D_{1,1} - \omega^U_{1,1}}{\omega^U_{1,1}}, \frac{\omega^D_{2,1} - \omega^U_{2,1}}{\omega^U_{2,1}}, \ldots, \frac{\omega^D_{n_1,1} - \omega^U_{n_1,1}}{\omega^U_{n_1,1}}, \ldots, \frac{\omega^D_{n_N,N} - \omega^U_{n_N,N}}{\omega^U_{n_N,N}} \right\} \tag{14}$$

The superscripts D and U refer to undamaged and damaged, respectively, $\omega_{i,n}$ is the ith antiresonant frequency of the nth FRF, and N is the number of responses measured.

To reduce the effects of experimental noise, simulated data are polluted with random noise. As proposed by Hjelmstad and Shin (1997), each set of perturbed data is created by adding a uniformly distributed random noise to the numerical data

$$\omega_{i,n} = \omega_{i,n}(1 + \xi) \tag{15}$$

where ξ = uniform random number with an estimated range of ±0.015. The range of the perturbing noise should be the same as the range of the measurement noise, which in this case was estimated to be 1.5%.

Output Vector

The network outputs correspond to a parameterization of the damage present in the structure. Damage parameterization is a very important aspect of damage assessment. The success of a damage assessment algorithm relies on the quality of the parameterization used to model the damage. There are three usual approaches used to model local damage such as a crack: local stiffness reduction, discrete spring models, and complex models in two or three dimensions. The last approach produces detailed and accurate results. Nevertheless, these models are difficult to apply in structural damage detection because they require a large number of degrees of freedom, and they need to revise the mesh as the damage location changes. Furthermore, Friswell and Penny (2002) demonstrated that at low frequencies, simple methods such as the stiffness reduction and the spring model correctly model a crack. Friswell and Penny show that a more detailed model does not substantially improve the results from damage detection.

Here, the output is represented by elemental stiffness reduction factors, α_i, defined as the ratio of the stiffness reduction to the initial stiffness

$$\mathbf{t} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\} \tag{16}$$

where m = number of structural elements.

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

$$\mathbf{K}_d = \sum_{i=1}^{m}(1 - \alpha_i)\mathbf{K}_i \tag{17}$$

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

Training and Validation Patterns

The distribution of training patterns plays a crucial role in the success of a neural network. The relationship between antiresonant frequencies and different damage levels is not linear. Therefore, a network might not be able to interpolate data. Fang et al. (2005) recommended using training patterns with evenly distributed damaged levels. By combining multiple damages into the training, the number of training patterns is increased and a large number of training patterns could overwhelm the training procedure. In this study, training patterns were generated by considering up to two simultaneous damages, with 10 damage levels evenly distributed between 5 to 95%. Hence, the total number of training patterns depends on the number of elements, N, was as follows:

$$\text{Number of training patterns} = 10^2 \times C(N, 2) + 1 \tag{18}$$

where $C(N, 2)$ = number of two-element combinations from a given set of N elements. The plus-one term refers to the undamaged case.

In this study, after training, the performance of the network was evaluated using a set of validation patterns, which were different from the training patterns. To ensure that the last condition was met, the validation patterns were created with nine damage levels evenly distributed between 10 and 90%, considering up to two simultaneous damages.

Measures of Network Performance

Three indicators measure the performance of the network: the mean sizing error (MSE), damage missing error (DME), and false alarm error (FAE), as defined by Yun et al. (2001). The MSE is the average quantification error

$$\mathrm{MSE} = \frac{1}{\mathrm{NO}}\sum_i |y_i - t_i| \tag{19}$$

where y_i and t_i = estimated and desired output for node i; and NO = number of output nodes.

The DME is given by

$$\mathrm{DME} = \frac{1}{\mathrm{NT}}\sum_i \varepsilon_i^I, \qquad 0 \leq \mathrm{DME} \leq 1 \tag{20}$$

where $\varepsilon_i^I = 0$ if the ith damaged element is correctly detected and $\varepsilon_i^I = 1$ if it is not. 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:

$$\varepsilon_i^I = \begin{cases} 1 & \text{if } t_i > 0 \quad \text{and} \quad y_i \leq \alpha_c \\ 0 & \qquad\text{otherwise} \end{cases} \tag{21}$$

It is assumed that an element is detected as being 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 equal to the average MSE, which is the minimum damage the network can reasonably assess.

The FAE is defined as

$$\mathrm{FAE} = \frac{1}{\mathrm{NF}}\sum_i \varepsilon_i^{II}, \qquad 0 \leq \mathrm{FAE} \leq 1 \tag{22}$$

where $\varepsilon_i^{II} = 0$ if the ith detected damage is truly damaged and $\varepsilon_i^{II} = 1$ if it is not. 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:

$$\varepsilon_i^{II} = \begin{cases} 1 & \text{if } y_i \geq \alpha_c \quad \text{and} \quad t_i = 0 \\ 0 & \qquad\text{otherwise} \end{cases} \tag{23}$$

Application Cases

The results of the OS-ELM algorithm are compared with those obtained by a model updating method based on parallel genetic algorithms (Meruane 2013), which uses the same antiresonant frequencies to assess damage. The procedure to assess the experimental damage using the OS-ELM algorithm was implemented as follows:

1. Construct a database of training patterns using a numerical model of the structure;
2. Select the number of hidden nodes and number of initialization patterns;
3. Train the network following the OS-ELM procedure described previously;
4. Perform an experimental test of the damaged structure and identify the antiresonant frequencies;
5. Construct the input vector using Eq. (14); and
6. Estimate the experimental damage from Eq. (1).

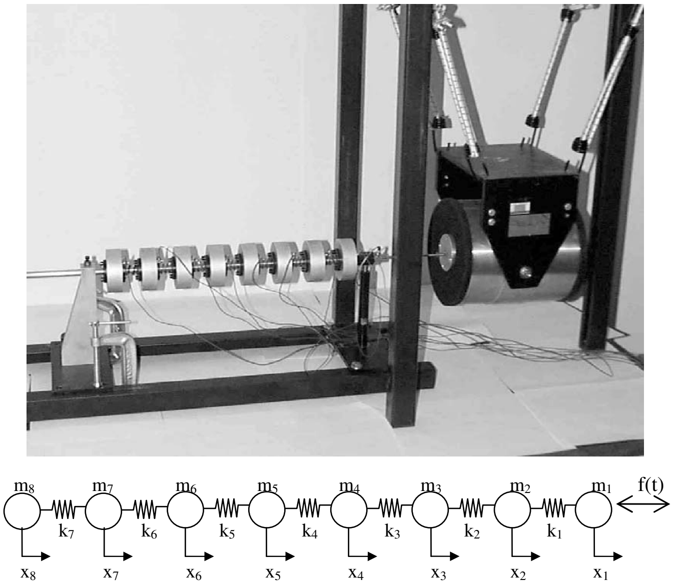

Fig. 3. Experimental eight-degrees-of-freedom system (reprinted from Duffey et al. 2001, with permission from ASME)

The next subsections present each application case and the results obtained by the two approaches.

Eight-DOF Spring-Mass System

The structure shown in Fig. 3 consists of an eight-DOF spring-mass system. Los Alamos National Laboratory (LANL) designed and constructed this system to study the effectiveness of various vibration-based damage identification techniques (Duffey et al. 2001).

Eight translating masses connected by springs form the system. Each mass is a disc of aluminum with a diameter of 76.2 mm and a thickness of 25.4 mm. The masses slide on a polished steel rod and are fastened together with coil springs. The positions of the springs and masses are designated sequentially, with the first ones being closest to the shaker attachment.

In the undamaged configuration, all springs are identical and have a linear stiffness coefficient. Damage was simulated by replacing the fifth spring with another spring that has a lower stiffness (55% stiffness reduction). Acceleration was measured horizontally at each mass, yielding eight measured DOFs. The structure was excited randomly by an electrodynamic shaker. Twenty-eight antiresonant frequencies were identified from the transmissibility measurements.

The numerical model was built in *MATLAB* with springs and concentrated masses, and it was updated using the algorithm described in Meruane (2013). The value of each mass and spring constant was updated individually to match the numerical and experimental antiresonant frequencies. After updating, the maximum difference between the experimental and numerical antiresonant frequencies was 3.7%. Therefore, the numerical model provided a faithful representation of the experimental structure and could be used to train the neural network. The final parameters for the spring-mass system are presented in Table 1; the first mass is greater than the others because of the hardware required to attach the shaker.

To set up the network the only parameter that needs to be selected is the number of hidden nodes. This parameter was selected after a sensitivity analysis; the performance of the network was

Table 1. Parameters of the Eight-DOF Spring-Mass System

Spring number	Spring constant (kN/m)	Disc number	Mass (kg)
1	56.73	1	0.520
2	57.94	2	0.404
3	57.82	3	0.417
4	55.05	4	0.414
5	58.78	5	0.424
6	61.84	6	0.421
7	59.83	7	0.419
—	—	8	0.416

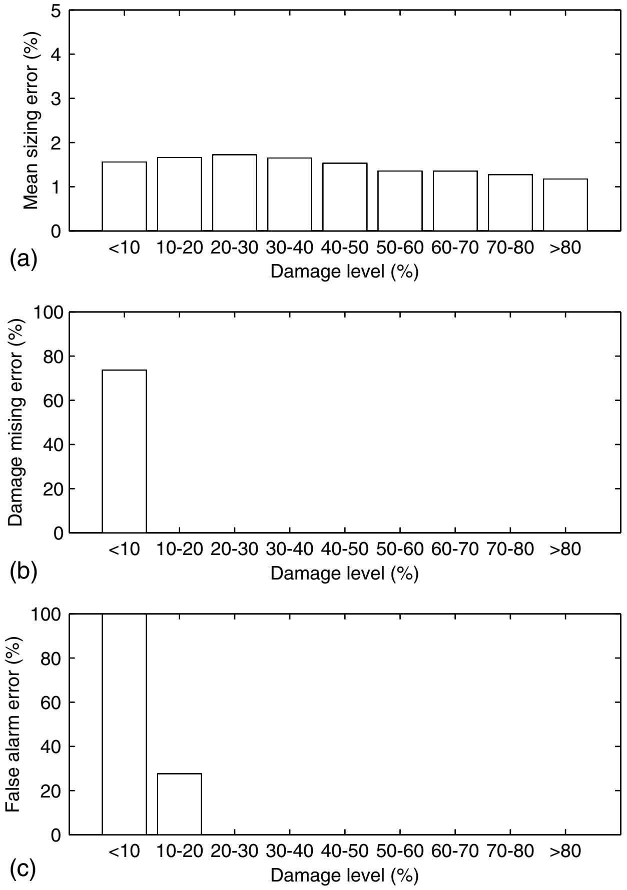

Fig. 4. Evaluation of network performance using validation patterns: (a) mean sizing error; (b) damage missing error; (c) false alarm error

tested with an increasing number of hidden nodes until there was no improvement in the network performance. The performance was measured by the MSE, DME, and FAE. The best performance was obtained using a network with 165 hidden nodes.

In total, there were 2,101 training patterns, which were divided in initialization and training patterns. The number of initialization patterns were defined by trial and error, the only condition being that this number has to be larger than the number of hidden nodes. The best training performance was obtained when approximately 23% of the patterns are used to initialize the network. Therefore, the patterns were divided into an initial set of 501 patterns to initialize the network and 16 blocks of 100 patterns used during the training phase.

Training and validation patterns were polluted with 1.5% noise. The time needed to train the network was 13 s.

The validation patterns verified, as a first step, the performance of the network. Fig. 4 shows the MSE, DME, and FAE divided by the damage levels. The mean sizing error was approximately 1.5% independent of the damage level, which implies that if the network quantifies a damage as 20%, the actual value is between 18.5 and 21.5%. The DME results indicate that the detection of damage of low severity is poor. In fact, 74% of the damages with severities lower than 10% were not detected. Nevertheless, the algorithm could detect, with confidence, damages with severities greater than 10%; it correctly detected all damages with severities greater than 10%. The false alarm results indicate that most of the damages detected at levels lower than 10% were false damages (near 99%). The degree of false damage detection was drastically reduced with the increment in the level of damage. All detected damages with levels over 20% were true damages.

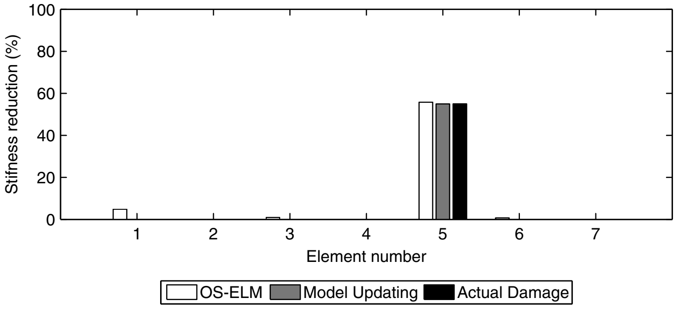

Fig. 5. Identification of experimental damage in the eight-DOF mass-spring system using two methods: OS-ELM and model updating with parallel genetic algorithms

Fig. 5 shows the results of the experimental damage case; the results are compared with those obtained using a model updating with parallel genetic algorithms approach. Both algorithms were able to identify the experimental damage represented by a 55% stiffness reduction in element 5. Unlike the GA approach, the neural network algorithm identified a few false damages, but all of them exhibit magnitudes lower than 10%. In terms of time, the GA approach required 206 s to yield a solution, whereas the network approach required only 0.005 s. The solution time does not include the time needed to train the network because the network needs to be trained once and then can assess any damage scenario. Hence, if only the time to assess damage is considered, the neural network approach reached nearly the same solution as the GA approach, but 41,200 times faster.

Experimental Beam

In the second experimental case, the structure consisted of a steel beam with a rectangular cross section. The beam measured 1 m in length and had a cross-sectional area of $25 \times 10\,\mathrm{mm}^2$. As shown in Fig. 6, soft springs suspend the structure to simulate a free-free boundary condition. A shaker excites the beam at one end, and the response is measured by 11 accelerometers. Both the excitation force and the measured responses are in the horizontal direction. In this direction, antiresonant frequencies are more sensitive to the experimental damage. Thirty-six antiresonant frequencies were identified from the transmissibility measurements.

The numerical model was built in *MATLAB* with two-dimensional beam elements. The model featured 20 beam elements and 40 degrees of freedom, as shown in Fig. 7. Shadowed elements represent possible locations of damage.

The structure was subjected to three different damage scenarios containing single and double cracks. Cracks were introduced into the structure by saw cuts measuring l_c in length, as illustrated in

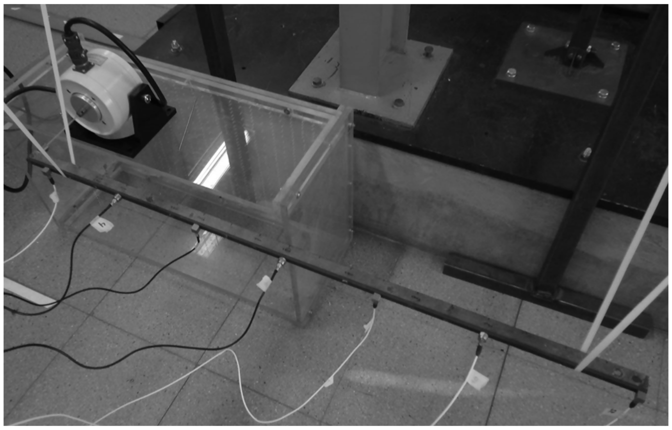

Fig. 6. Experimental beam

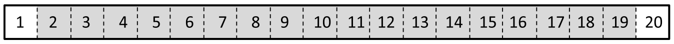

Fig. 7. Numerical model and element numbering

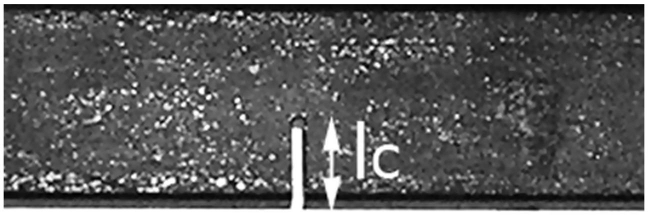

Fig. 8. Saw cut introduced into the beam

Table 2. Damage Scenarios Introduced to the Beam

Damage scenario	Distance from the left end (mm)	Element number	Saw cut length (mm)
1	685	14	9
2	360	8	12
3	360	8	5
1	—	—	—
2	697	14	6
3	810	17	17

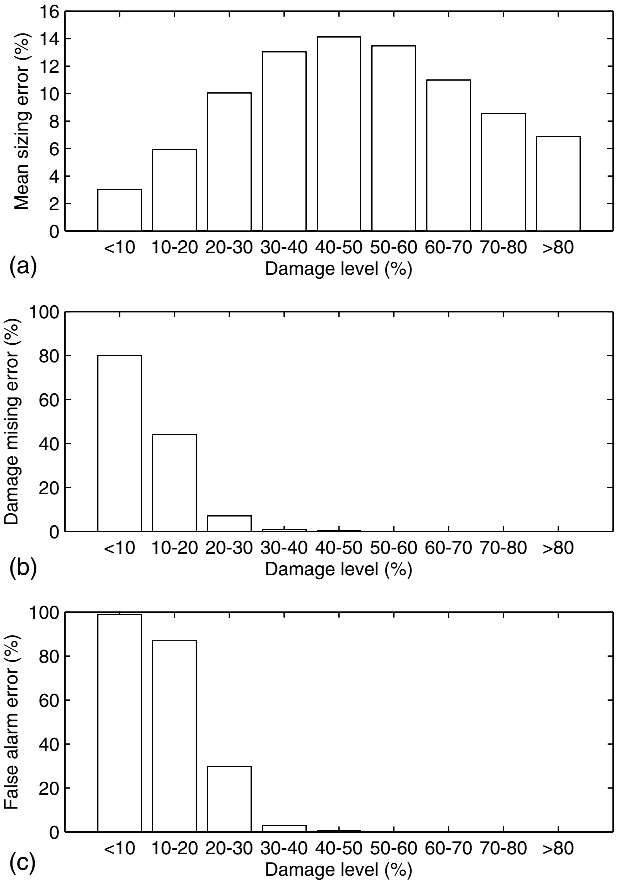

Fig. 9. Evaluation of network performance using validation patterns: (a) mean sizing error; (b) damage missing error; (c) false alarm error

Fig. 8. Table 2 summarizes the different damage scenarios, indicating the distance from the left end to the cut, the corresponding element in the numerical model, and the cut length.

The network featured 480 hidden nodes. In total, there were 15,301 training patterns, which were divided into an initial set of 1,701 patterns to initialize the network and 68 blocks of 200 patterns used during the training phase. The selection of the number of hidden nodes and the number of initialization patterns was performed following the same procedure described for the eight-DOF spring-mass system.

Training and validation patterns were polluted with 1.5% random noise. The time needed to train the network was 687 s.

Fig. 9 shows the MSE, DME, and FAE separated by damage levels. The MSE was greater for damages in the range of 40–50% stiffness reduction, which possess a MSE of 14%. This discrepancy implies that quantification was not accurate: if the algorithm detects a damage represented by a 45% stiffness reduction, the actual value is in the range of 31–59%. The DME results indicate that the detection of damages with low severities is poor. In fact, nearly 80% of the damages with severities lower than 10%, and 43% of the damages with severities between 10 and 20%, were not detected. The algorithm can detect, with confidence, damages with severities greater than 30%; in fact, it correctly detected 99.8% of the damages with severities greater than 30%. The false alarm results indicate that most of the damages detected with levels lower than 10% were false damages (near 98.9%). The degree of false damage detection was reduced with the increase in the level of damage; 99.2% of the damages detected with levels over 30% were true damages.

These results are slightly worse than those obtained for the eight-DOF system. This difference is explained by the fact that antiresonant frequencies are less sensitive to damage in the free beam than in the eight-DOF structure, thus making it more difficult to detect small damages.

Fig. 10 shows the results of the damage detected in the three experimental damage scenarios. The actual damage was estimated by using a full three-dimensional model of the beam with the real cut in conjunction with a simplified model using a stiffness reduction to represent the cut. The magnitude of the stiffness reduction is the one that minimizes the difference between both models. In the first two cases, damage is correctly located by both approaches, though the neural network approach detects a few false damages. In the last case, neither of the algorithms detects the small crack in element 8, which is because the larger cut hides the effect of the smaller cut.

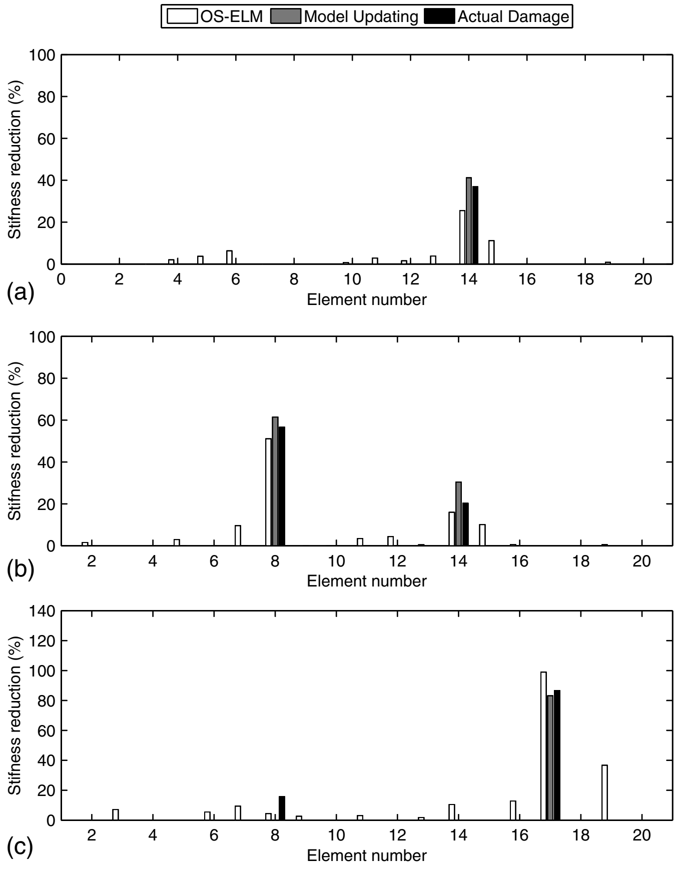

Fig. 10. Identification of experimental damage in the beam using two methods: OS-ELM and model updating with parallel genetic algorithms: (a) Case 1; (b) Case 2; (c) Case 3

In terms of time, the GA approach requires approximately 900 s to assess the experimental damage in each case, whereas the neural network approach requires only 0.001 s.

Discussion

Real-time damage assessment allows continuous monitoring of the state of a structure, which is critical to avoid catastrophic failures. In this context, there are two essential attributes that a damage assessment method must meet. First, it must detect the damage early enough so repair actions can be performed before the structure fails. Second, the algorithm should be able to provide a diagnosis in real time, and thus the damage is detected as soon as it appears.

The next step of damage assessment is the prediction of the remaining life. This can be accomplished by assessing the current damage state, estimating the future loading environments, and predicting through numerical simulations and experience the remaining useful life of the system. With this information it is possible to plan maintenance actions. Therefore, damage must be detected in time to perform maintenance before the structure fails.

Laory et al. (2013) investigated the minimum detectable damage and the time to detection for different model-free damage detection methods. Using a combined approach, Laory et al. were able to detect damage as small as 3% stiffness reduction in a simulated steel truss bridge. Nevertheless, when experimental structures are tested, the minimum detectable damage increases due to the experimental noise. To detect damage in an experimental structure, the changes produced by the damage must be larger than the experimental variability. Thus, the minimum detectable damage is directly related to the level of experimental noise. Yun et al. (2001) trained a neural network with a noise-injection algorithm to assess joint damage of frame structures. They demonstrate that if the level of experimental noise increases the minimum detectable damage also increases, obtaining a network performance similar to that presented in this paper.

The minimum detectable damage of 20–30% in stiffness reduction could be detected by nondestructive testing (NDT) methods such as visual inspections. Nevertheless, vibration-based damage assessment has two important advantages over these methods. First, it has the potential to provide information during the life cycle of the structure, while inspections are usually suited to check the immediate condition of specific structural members after an event or during preventive maintenance. Second, vibration-based damage assessment can detect damage throughout the complete structure even if the damage is located in hidden or internal areas, while inspections require access to the area being inspected.

Conclusions

This paper presents a new methodology for assessing experimental damage using an OS-ELM algorithm and antiresonant frequencies identified from transmissibility measurements. A single-hidden-layer feed-forward network was trained with data obtained from a numerical model and tested with experimental data. Two experimental cases validate the performance of the algorithm: an eight-DOF mass-spring system and a beam under multiple damage scenarios. The results are compared with those obtained by a model updating with parallel genetic algorithms approach.

In both structures, the algorithm is successful in assessing the experimental damage. The damage detected corresponds well to the experimental damage in all cases. These results demonstrate that it is possible to locate and quantify structural damage using only antiresonant information obtained from transmissibility data. Hence, antiresonant frequencies are an attractive feature to use in damage assessment.

The present study demonstrates that it is possible to assess experimental damage in real time in two structures. The network approach was able to assess damage in approximately 0.001 s, whereas the GA approach required approximately 15 min. Hence, neural networks provide the possibility of continuously monitoring the state of a structure. Nevertheless, according to the validation results, the algorithm has a minimum level of damage that can be assessed with confidence, which is 20% in the case of the eight-DOF system and 30% for the beam. In addition, the quantification of damage is not very accurate in the neural network case. It is recommended that a neural network be used to yield rapid damage identification and that a GA approach be used to provide a more accurate solution.

Acknowledgments

The author acknowledges the partial financial support of the Chilean National Fund for Scientific and Technological Development (Fondecyt) under Grant No. 11110046.

References

Arangio, S., and Beck, J. L. (2012). "Bayesian neural networks for bridge integrity assessment." *Struct. Control Health Monit.*, 19(1), 3–21.

Arbib, M. (2003). *The handbook of brain theory and neural networks*, MIT Press, Cambridge, MA.

Bamnios, Y., Douka, E., and Trochidis, A. (2002). "Crack identification in beam structures using mechanical impedance." *J. Sound Vib.*, 256(2), 287–297.

Barai, S., and Pandey, P. (1995). "Vibration signature analysis using artificial neural networks." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)0887-3801(1995)9:4(259), 259–265.

Castellini, P., and Revel, G. (2000). "An experimental technique for structural diagnostic based on laser vibrometry and neural networks." *Shock Vib.*, 7(6), 381–397.

D'Ambrogio, W., and Fregolent, A. (2000). "The use of antiresonances for robust model updating." *J. Sound Vib.*, 236(2), 227–243.

Dilena, M., and Morassi, A. (2004). "The use of antiresonances for crack detection in beams." *J. Sound Vib.*, 276(1–2), 195–214.

Duffey, T., Doebling, S., Farrar, C., Baker, W., Rhee, W., and Doebling, S. (2001). "Vibration-based damage identification in structures exhibiting axial and torsional response." *Trans. ASME-L-J. Vib. Acoust.*, 123(1), 84–91.

Fang, X., Luo, H., and Tang, J. (2005). "Structural damage detection using neural network with learning rate improvement." *Comput. Struct.*, 83(25–26), 2150–2161.

Friswell, M. I., and Penny, J. E. (2002). "Crack modeling for structural health monitoring." *Struct. Health Monit.*, 1(2), 139–148.

Friswell, M. I., Penny, J. E., and Garvey, S. D. (1998). "A combined genetic and eigensensitivity algorithm for the location of damage in structures." *Comput. Struct.*, 69(5), 547–556.

Gonzalez-Perez, C., and Valdes-Gonzalez, J. (2011). "Identification of structural damage in a vehicular bridge using artificial neural networks." *Struct. Health Monit.*, 10(1), 33–48.

Hakim, S. J. S., and Abdul Razak, H. (2014). "Modal parameters based structural damage detection using artificial neural networks—A review." *Smart Struct. Syst.*, 14(2), 159–189.

Hjelmstad, K., and Shin, S. (1997). "Damage detection and assessment of structures from static response." *J. Eng. Mech*, 10.1061/(ASCE)0733-9399(1997)123:6(568), 568–576.

Huang, G.-B., Wang, D. H., and Lan, Y. (2011). "Extreme learning machines: A survey." *Int. J. Mach. Learn. Cybern.*, 2(2), 107–122.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2004). "Extreme learning machine: A new learning scheme of feedforward neural networks." *Neural Networks, 2004. Proc. 2004 IEEE Int. Joint Conf. on*, Electrical and Electronics Engineers (IEEE), New York, 985–990.

Inada, T., Shimamura, Y., Todoroki, A., and Kobayashi, H. (2004). "Development of the two-step delamination identification method by resonant and anti-resonant frequency changes." *Key Eng. Mater.*, 270–273(1), 1852–1858.

Khaji, N. A. (2014). "Crack detection in a beam with an arbitrary number of transverse cracks using genetic algorithms." *J. Mech. Sci. Technol.*, 28(3), 823–836.

Kouchmeshky, B., Aquino, W., Bongard, J., and Lipson, H. (2007). "Co-evolutionary algorithm for structural damage identification using minimal physical testing." *Int. J. Numer. Methods Eng.*, 69(5), 1085–1107.

Laory, I., Trinh, T. N., Posenato, D., and Smith, I. F. (2013). "Combined model-free data-interpretation methodologies for damage detection during continuous monitoring of structures." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000289, 657–666.

Lee, J., Lee, J., Yi, J., Yun, C., and Jung, H. (2005). "Neural networks-based damage detection for bridges considering errors in baseline finite element models." *J. Sound Vib.*, 280(3–5), 555–578.

Liang, N.-Y., Huang, G.-B., Saratchandran, P., and Sundararajan, N. (2006). "A fast and accurate online sequential learning algorithm for feedforward networks." *IEEE Trans. Neural Networks*, 17(6), 1411–1423.

Lim, R., and Ewins, D. (1990). "Model updating using FRF data." *Proc., 15th Int. Seminar on Modal Analysis*, Katholieke Universiteit Leuven (KULeuven), Leuven, Belgium, 141–163.

MATLAB [Computer software]. Natick, MA, Math Works.

Masri, S., Smyth, A., Chassiakos, A., Caughey, T., and Hunter, N. (2000). "Application of neural networks for detection of changes in nonlinear systems." *J. Eng. Mech.*, 10.1061/(ASCE)0733-9399(2000)126:7(666), 666–676.

Meruane, V. (2013). "Model updating using antiresonant frequencies identified from transmissibility functions." *J. Sound Vib.*, 332(4), 807–820.

Meruane, V., and Heylen, W. (2010). "Damage detection with parallel genetic algorithms and operational modes." *Struct. Health Monit.*, 9(6), 481–496.

Meruane, V., and Heylen, W. (2011a). "An hybrid real genetic algorithm to detect structural damage using modal properties." *Mech. Syst. Signal Process.*, 25(5), 1559–1573.

Meruane, V., and Heylen, W. (2011b). "Structural damage assessment with antiresonances versus mode shapes using parallel genetic algorithms." *Struct. Control Health Monit.*, 18(8), 825–839.

Meruane, V., and Mahu, J. (2014). "Real-time structural damage assessment using artificial neural networks and anti-resonant frequencies." *Shock Vib.*, 2014, 14.

Messina, A., Jones, A., and Williams, E. (1996). "Damage detection and localisation using natural frequency changes." *Proc., 1st Conf. on Identification in Engineering Systems*, Univ. of Wales, Swansea, U.K., 67–76.

Perera, R., and Torres, R. (2006). "Structural damage detection via modal data with genetic algorithms." *J. Struct. Eng.*, 10.1061/(ASCE)0733-9445(2006)132:9(1491), 1491–1501.

Platt, J. (1991). "A resource-allocating network for function interpolation." *Neural Comput.*, 3(2), 213–225.

Sahoo, B., and Maity, D. (2007). "Damage assessment of structures using hybrid neuro-genetic algorithm." *Appl. Soft Comput.*, 7(1), 89–104.

Stubbs, N., Kim, J., and Topole, K. (1992). "An efficient and robust algorithm for damage localization in offshore platforms." *Proc., ASCE Tenth Structures Congress*, ASCE, Reston, VA, 543–546.

Teughels, A., Roeck, G. D., and Suykens, J. (2003). "Global optimization by coupled local minimizers and its application to FE model updating." *Comput. Struct.*, 81(24–25), 2337–2351.

Wang, D., and Zhu, H. (2005). "Wave propagation based multi-crack identification in beam structures through anti-resonances information." *Key Eng. Mater.*, 293–294, 557–564.

Williams, E., and Messina, A. (1999). "Applications of the multiple damage location assurance criterion." *Key Eng. Mater.*, 167–168(1), 256–264.

Yun, C. B., Yi, J. H., and Bahng, E. Y. (2001). "Joint damage assessment of framed structures using a neural networks technique." *Eng. Struct.*, 23(5), 425–435.

Zang, C., and Imregun, M. (2001). "Structural damage detection using artificial neural networks and measured FRF data reduced via principal component projection." *J. Sound Vib.*, 242(5), 813–827.