USING NEURAL NETWORKS FOR F.E. MODEL UPDATING OF STRUCTURES IN OPERATIONAL CONDITIONS

F. Santamaria\textsuperscript{1}, M. Arras\textsuperscript{2}, and G. Coppotelli\textsuperscript{3}

\textsuperscript{1} Graduated Student, University of Rome La Sapienza, frasan84@gmail.com.
\textsuperscript{2} Ph.D. Student, University of Rome La Sapienza, melissa.arras@uniroma1.it.
\textsuperscript{3} Assistant Professor, University of Rome La Sapienza, giuliano.coppotelli@uniroma1.it.

ABSTRACT
In this paper, a finite element model updating method based on neural networks is presented. The main objective of the paper is to identify the dynamic properties of a structure from response data recorded during operating conditions, extending the use of results from operational modal analysis to the neural networks-based updating methodologies. The neural networks used in this study have a feed-forward architecture and their inputs are the modal parameters, that is natural frequencies, damping ratios, and mode shapes of a structure in its operative conditions, whereas their outputs are the physical properties of the considered structure. Typically, the first step of neural networks is their training and it will be shown that trained neural networks are successful when simulated cases are considered but they have some limits when experimental data are used. For this reason an algorithm based on not-trained neural networks has been developed. Both numerical and experimental analyses carried out on simple structures will be presented to demonstrate the accuracy of the proposed updating approach.

Keywords: Updating, Neural Networks, Operational Modal Analysis

1. INTRODUCTION
Finite element (F.E.) method is frequently used to analyze the dynamic behaviour of real structures. Because a F.E. model represents a structure from the numerical point of view, it is necessary to correlate it with its experimental counterpart that represents the real structure to verify the accuracy of the numerical model and correct it if considerable discrepancies are present. Usually the numerical modal parameters, that is natural frequencies, damping ratios, and mode shapes differ from those of the corresponding experimental model. If the p-values formulation \textsuperscript{1} is adopted, the physical properties of the F.E. model (thickness, density, Young’s modulus..) can be associated to the updating parameters that act on the stiffness and mass matrices of the numerical model, and the differences between the two models can be reduced by performing a structural updating. The updated F.E. takes possible mismodeled regions into
account allowing thus its use in synthesis and/or structural modification prediction problems. Several methods have been developed to update F.E. models using measured data [1, 2, 3, 4] and in this paper, algorithms that use neural networks (NNs) for model updating are presented. Previous studies in this field have shown that neural networks are robust with respect to the experimental noise [5] and that they are able to reduce differences between numerical and experimental modal parameters [6, 7], that is the aim of structural updating methods. Typically, results from a traditional experimental analysis, where both inputs and outputs are measured, are used to update the numerical model. Although this analysis allows to identify the dynamic characteristics of a structure, it is not fully representative of the structure in operating conditions and for this reason, the operational, or output-only, modal analysis (OMA) obtains increasing interest nowadays. The output-only approaches permit to identify the modal parameters of a structure in its operative conditions using the ambient excitation and measuring only the responses of the system. The identification of the operative dynamic properties of systems could be performed both in the time [8], and in the frequency domain [9, 10]. Time domain based methods estimate the modal parameters in the frame of the state-space formulation, whereas the starting point of the frequency domain based methods is the power spectral density matrix evaluated among all the output responses. In this paper the Hilbert Transform Method (HTM) [11] is used for the estimate of the modal parameters of the structure in operating conditions. These parameters are then used as NNs input whereas their outputs are the physical properties of the considered structure that narrow the gap between the numerical and experimental model. Generally, NNs have to be trained through the training process that, in the case of model updating, is done using data coming from modal analysis. Although trained NNs provide good results in numerical simulated cases, they have some limits when experimental data are used so, an updating method based on not-trained NNs has been developed. The paper is provided with a numerical analysis and an experimental investigation on a simple cantilever beam in order to validate and assess the proposed approach.

2. THEORETICAL BACKGROUND

2.1. Neural Networks

Neural networks are mathematical tools that simulate the relationship between inputs and outputs of a system. Starting from some known input-output pairs of a system, the training process allows to iteratively change the free parameters of the network in order to reduce, for the same input, the error between the output of the system and that of the network. Several computational units linked to each other through some parameters called weights constitute a neural network. One of the most common unit is the perceptron, [12], shown in Fig. 1. The output of perceptron is:

$$y = S \left( \sum_{i=1}^{n} w_i x_i + b \right)$$  \hspace{1cm} (1)
where \( n \) is the number of inputs, \( x_i \) the value of the \( i \)-th input, \( w_i \) the \( i \)-th weight of perceptron, \( b \) is called bias of perceptron, and \( S \) is the transfer function or activation function. Weights and bias are the network free parameters that can be changed by the training process. Further details on the NNs can be found in \([13,14,15]\). The commonly used transfer functions are the sigmoid symmetric and the pure linear function:

\[
S(h) = \frac{1 - e^{-h}}{1 + e^{-h}} \quad \text{(sigmoid symmetric)}
\]

\[
S(h) = h \quad \text{(pure linear)}
\]  

(2)

Usually, perceptrons are combined together in a multilayer network, where the input and output units are grouped into the input and output layer respectively, and the layers between inputs and outputs are called hidden layers, Fig. 2. The architecture of NNs, i.e. the type of units, their number, and how they are linked, depends on the problem to solve and the choice can be done through trial and error methods. In this paper a feedforward neural network, that is a multilayer network where units of one layer are linked to those of the next layer without loops, has been used. Moreover, the sigmoid symmetric and the pure linear transfer functions have been used for the hidden and the output layers, respectively. The first step in the use of NNs is the initialization of weights and the Nguyen-Widrow algorithm, \([16]\), is one of the most efficient method. Then the training process is usually performed through the backpropagation algorithm, \([17]\). After the training process, training data have to be generalized, \([13]\). For the sake of brevity, a detailed description of various steps is not provided and further details can be found in the above-cited references.

### 2.2. Neural Networks and Model Updating

In the field of model updating, it is possible to define modal-based or response-based techniques. The first ones minimize the differences between the measured and predicted modal parameters, whereas the second ones compare the measured and predicted Frequency Response Functions (FRFs). FRFs are defined at each frequency point and, depending on the frequency range and frequency resolution, they could result in a large number of input data for the NNs slowing down the training process. For this reason, a modal-based technique is usually used when NNs are introduced to update a F.E. model. The difference between the measured and predicted modal parameters can be reduced by altering the updating parameters corresponding to the physical properties of the F.E. model. Neural networks allow to reproduce the relationship between the natural frequencies and the mode shapes, inputs of the network, of the numerical model and its physical properties, outputs of the network. As introduced in the previous section, the training process is the essential step when using NNs. After that, the experimental data are introduced to the network that provides the updating parameters and the updated modal parameters are calculated. The iterative algorithm goes on until the predicted natural frequencies and mode shapes
match the experimental ones, according to some predetermined coefficients. Specifically, the numerical and experimental natural frequencies can be compared through the percentage difference defined as:

$$d_j = \frac{f_j - \bar{f}_j}{f_j} \cdot 100$$  \hspace{1cm} (3)$$

where $f_j$ is the $j-$th experimental natural frequency whereas $\bar{f}_j$ is the corresponding numerical natural frequency. The Modal Assurance Criterion\(^1\) allows to compare the numerical and experimental mode shapes. In order to have a global information on the differences between the measured and predicted quantities, two coefficients are introduced:

$$\text{MFE} = \frac{1}{N} \sum_{j=1}^{N} d_j \hspace{1cm} \text{Mean Frequency Error}$$  \hspace{1cm} (4)$$

$$\overline{\text{MAC}} = \frac{1}{N} \sum_{j=1}^{N} \text{MAC}_{jj} \hspace{1cm} \text{Mean of MAC}$$  \hspace{1cm} (5)$$

where $N$ is the number of natural frequencies and mode shapes available from experimental tests. The aim of the updating process is to have a MFE that tends to a zero value, and an unitary value for $\overline{\text{MAC}}$. The algorithm can be stopped when the two coefficients, MFE and $\overline{\text{MAC}}$, are lower than a given threshold or when a given number of iteration steps is reached. While trained neural networks give excellent results when simulated cases are considered, they are not so good when experimental data are used. For this reason, a new algorithm has been developed by replacing the training phase with the generation of different neural networks. Then, through a selection process it is possible to choose the network that minimizes an objective function $J$, defined as:

$$J = J_1 + \alpha \cdot J_2 = \text{MFE} + \alpha \cdot (1 - \overline{\text{MAC}})$$  \hspace{1cm} (6)$$

where $\alpha$ is a coefficient that balances $J_1$ and $J_2$ during the updating process. The value of $\alpha$ can be calculated at each $k-$th step of algorithm as:

$$\alpha^k = \frac{J_1^{k-1}}{J_2^{k-1}}$$  \hspace{1cm} (7)$$

The selection of the neural network that minimizes $J$ is obtained by calculating, for each network, the variations of $J_1$, $J_2$, and $J$ at the $k-$th iteration step with respect to their values at the previous step:

$$
\begin{align*}
\Delta J_{1l} &= J_{1l}^k - J_{1l}^{k-1} \\
\Delta J_{2l} &= J_{2l}^k - J_{2l}^{k-1} \\
\Delta J_l &= \Delta J_{1l} + \alpha \cdot \Delta J_{2l}
\end{align*}
$$  \hspace{1cm} (8)$$

where the subscript $l$ indicates that quantities are calculated for the $l-$th network. Then, a probability value is assigned to each network. The probability function is defined as:

$$P(\Delta J_l) = \begin{cases} 
1 & \text{if } \Delta J_{1l} < 0 \text{ and } \Delta J_{2l} < 0 \\
\log\operatorname{sig}(-\Delta J_l) & \text{in the other cases}
\end{cases}$$  \hspace{1cm} (9)$$

\(^1\)The Modal Assurance Criterion, MAC, is defined as:

$$\text{MAC}_{ij} = \frac{|\Phi_i H \Phi_j|^2}{|\Phi_i H \Phi_i|^2}$$

where $\Phi_i$ is the $i-$th vector containing the components of the complex experimental mode shape, $\Phi_j$ is the corresponding numerical $j-$th vector, and superscript $H$ indicates the hermitian operator. It assumes a value between zero and unity indicating a perfect correlation with the unitary value, whereas no correlation exists when it is zero.
where \( \text{logsig}(v) = 1/[1+\exp(-v)] \) is called logistic sigmoid. Negative values in Eq. 9 cause probability to tend to the unitary value for negative values of \( \Delta J_l \) (\( J \) is lower) and to zero value for positive values. In this way the selection occurs even if no network with unit probability is found, avoiding loops. Again, the not-trained networks algorithm can be stopped when the two coefficients, MFE and MAC, are lower than a given threshold or when a given number of iteration steps is reached.

2.3. Hilbert Transform Method

The aim of this study is to update a F.E. model by using experimental data coming from operative conditions. In this section some theoretical background on the frequency-domain operational modal analysis approach, called Hilbert Transform Method (HTM), is presented. [11]. This approach allows to estimate the operational natural frequencies and mode shapes of the structure under investigation. By measuring the output responses \( y_i, i = 1, 2, \ldots, N_o \), (being \( N_o \) the number of measurement points) over \( N_t \) time samples, and from the evaluation of the spectral density functions, \( G_{y_i y_j}(\omega_k) \), defined between the \( i \)-th and \( j \)-th output responses, at the \( k \)-th spectral line, it is possible to build the so called output spectral density function matrix \( G_{yy}(\omega_k) \) as:

\[
G_{yy}(\omega_k) = \begin{pmatrix}
G_{y_1 y_1}(\omega_k) & \cdots & G_{y_1 y_{N_o}}(\omega_k) \\
G_{y_{N_o} y_1}(\omega_k) & \cdots & G_{y_{N_o} y_{N_o}}(\omega_k)
\end{pmatrix}
\]

(10)

The advantage in using the Hilbert transform is the capability of estimating the imaginary part of a causal function starting from the real part. [13]. The polar representation of a driving point FRF \( H_{ii}(\omega) \) is given by introducing the amplitude \( |H_{ii}(\omega)| \) and the phase \( \phi_{ii}(\omega) \) functions as:

\[
H_{ii}(\omega) = |H_{ii}(\omega)| e^{-j\phi_{ii}(\omega)}
\]

(12)

that, by introducing the natural logarithm, could be expressed as

\[
\ln |H_{ii}(\omega)| = G_{ii}(\omega) - j\phi_{ii}(\omega)
\]

(13)

where \( G_{ii}(\omega) = \ln |H_{ii}(\omega)| \) is the gain function. Since the real and imaginary parts of the FRF are even and odd functions respectively, the gain and the phase are even and odd too. As a result, the left-hand side of the Eq. 13 could be expressed as the sum of a pair of Hilbert transform functions:

\[
\phi_{ii}(\omega) = -G_{ii}(\omega)
\]

(14)

The gain function is related to the spectral density function as:

\[
G_{yy}(\omega_k) = H(\omega_k) G_{ff}(\omega_k) H^H(\omega_k)
\]

(15)

where the input spectral density matrix, defined between the \( N_i \) inputs, i.e. \( G_{ff}(\omega_k) \in \mathbb{C}^{N_i \times N_i} \), is assumed to be derived from a white noise excitation. Therefore, it implies that \( G_{ff}(\omega_k) \) is frequency independent, that is \( G_{ff}(\omega_k) = G_{ff} \), where \( G_{ff} \) is a diagonal matrix when the input excitation is uncorrelated in the space domain. As a consequence, Eq. 15 by applying the natural logarithm and performing the Hilbert transform, becomes:

\[
\mathcal{H}[\ln(G_{y_i y_j}(\omega))] = 2\mathcal{H}[\ln |H_{ii}(\omega)|]
\]

(16)

The Hilbert transform of a signal \( x(t) \) is defined as the Cauchy principal value of

\[
\hat{x}(t) = \mathcal{H}[x(t)] = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{x(\tau)}{t-\tau} d\tau
\]

(11)
in which the input spectral density contribution, $G_{f_i}$, is null, as the Hilbert transform of a constant is zero. Combining the previous Eqs. [14 and 16] it is possible to write

$$\phi_{ii}(\omega) = -\frac{1}{2} H[\ln(G_{yy}(\omega))]$$

Therefore, the FRF in the i-th driving point is available. It is possible to demonstrate that the off-diagonal terms of the FRF are derivable from the comparison between the commonly used $H_1$ and $H_2$ estimators, [10 11]:

$$H_{ij}(\omega) = \frac{G_{yy}(\omega)}{\sqrt{G_{f_i f_i} H_{ii}^*(\omega)}}$$

The modal parameters are evaluated with a least square approximation, considering the expression of FRF in pole-residue terms. In the frequency range of definition of the FRF, the number of modes, $N_m$, is not known, therefore a stabilization diagram is introduced to estimate it by means of an iterative procedure. It is worth remarking that the structural properties are independent from the order used to describe the system, thus stable poles are representative of natural frequencies, [19].

3. RESULTS

3.1. Simulated case

Several simulated cases have been performed to verify the effectiveness of the updating algorithms, and only one is reported for the sake of brevity. The considered case study refers to an aluminum cantilever beam modeled as an Euler-Bernoulli beam, i.e., shear and axial strains are neglected, and discretized with 5 finite elements. According to the Euler-Bernoulli theory, two degrees of freedom (DOFs) are considered for each node, specifically the vertical displacement and the rotation about the transverse axis, so the considered cantilever beam has 10 DOFs. The numerical model, that is the starting model, has $\overline{\rho} = 2700 \text{ kg/m}^3$ and $\overline{E} = 70 \text{ GPa}$. In Fig. 3 the main geometrical properties of the beam are shown. The simulated experimental model is derived by changing the physical properties of some elements of the beam. As one can see from Table 1, the density of the 5-th element has been reduced by 30% and the Young’s modulus of the 1-st element has been increased by 30%. For the sake of clarity, the starting model denoted by overlined letters has to match the simulated experimental model. The updating algorithm should identify the modified quantities, i.e. the Young’s modulus for the 1-st and the density for the 5-th element, keeping unchanged the other quantities. The first five natural frequencies and the corresponding mode shapes were considered as inputs of the neural networks. So the number of inputs was thirty, namely five frequencies and twenty-five coordinates of mode shapes, five for each mode. The node in correspondence of the constraint was not considered for the analysis. The density and the Young’s modulus of each element were considered as the updating parameters, so ten outputs were provided by the neural networks. Neural networks in the algorithm with training had feedforward architecture with a single hidden layer. The number of units in the hidden layer was 60. Not-trained neural networks had the same architecture but with 5 units in the hidden layer. The reason for this choice

<table>
<thead>
<tr>
<th>Element #</th>
<th>Density</th>
<th>Young’s modulus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\rho \ [\text{kg/m}^3]$</td>
<td>$\overline{\rho} \ [\text{kg/m}^3]$</td>
</tr>
<tr>
<td>1</td>
<td>2700.00</td>
<td>2700.00</td>
</tr>
<tr>
<td>5</td>
<td>3857.14</td>
<td>2700.00</td>
</tr>
</tbody>
</table>
is that without training, generalization is not necessary and few perceptrons are sufficient to find the solution.

Fig. 4 shows trends of MFE and mean MAC for trained and not-trained neural networks. Algorithms were stopped when MFE reached a value less than $10^{-3}\%$. It is possible to note from Fig. 4 that not-trained networks appear slower than trained ones but both algorithms allow to reduce the natural
frequencies errors and increase the values of MAC. The training process helps to find solutions and it makes the algorithm with trained networks the best choice when experimental data are simulated. Table 2 shows the results in terms of natural frequency shifts and MAC values for the considered simulated case. After the updating process, excellent results have been obtained. Indeed, natural frequency shifts are very low, and MAC values are unitary. These results show that not-trained networks could be used for updating problems, thus validating the proposed approach. Regarding the updating parameters, Table 3 shows that trained NNs identify completely the unknown model, whereas not-trained NNs provide parameters with errors equal to 5.47% and 9.32% for the density and the Young’s modulus respectively. Moreover, not-trained networks modify the properties of all elements, confirming that trained networks are better for simulated cases.

Table 2: Simulated case - Natural frequency shifts and MAC before and after the updating.

<table>
<thead>
<tr>
<th>Mode #</th>
<th>Before updating</th>
<th>After updating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trained NNs</td>
<td>Not-trained NNs</td>
</tr>
<tr>
<td></td>
<td>f [Hz]</td>
<td>f [Hz]</td>
</tr>
<tr>
<td>1</td>
<td>3.3817</td>
<td>4.1127</td>
</tr>
<tr>
<td>2</td>
<td>23.7150</td>
<td>25.7862</td>
</tr>
<tr>
<td>3</td>
<td>68.2766</td>
<td>72.4252</td>
</tr>
<tr>
<td>4</td>
<td>133.6011</td>
<td>143.0748</td>
</tr>
<tr>
<td>5</td>
<td>221.3986</td>
<td>237.4685</td>
</tr>
</tbody>
</table>

Table 3: Simulated case - Updating parameters.

<table>
<thead>
<tr>
<th>Element #</th>
<th>Trained NNs</th>
<th>Not-trained NNs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ρ [kg/m³]</td>
<td>E [GPa]</td>
</tr>
<tr>
<td>1</td>
<td>2700.13</td>
<td>53.84</td>
</tr>
<tr>
<td>2</td>
<td>2700.02</td>
<td>70.00</td>
</tr>
<tr>
<td>3</td>
<td>2699.88</td>
<td>70.00</td>
</tr>
<tr>
<td>4</td>
<td>2700.07</td>
<td>69.99</td>
</tr>
<tr>
<td>5</td>
<td>3857.18</td>
<td>70.01</td>
</tr>
</tbody>
</table>

3.2. Experimental investigation

The experimental analysis was performed on a cantilever aluminum beam of dimensions 0.2 x 0.0154 x 0.00285 m. The vertical translations of the beam were measured at four equally spaced locations along the span. The structure was excited with a pencil crawling on one side of the beam and responses were recorded in the period of 40.96 s, Fig. 5 and the auto and cross spectral densities, considered in the 0-400 Hz frequency band, were evaluated using 16 data block records of length of 2048 sampling points, 20. Once the spectral density matrix was evaluated, the biased FRFs were estimated using the HTM approach and then the modal parameters were gained from a residue/pole fitting. An assessment of the estimates of the modal parameters is reported in Fig. 6, where the driving point FRFs corresponding to the tip measuring point (4-th channel) only are shown for brevity. Specifically, the solid curve is the biased FRF directly obtained from the HTM approach, whereas the dashed line correspond to the same FRF synthesized using the estimated modal parameters. Two eigenvalues were clearly identified from the frequencies where the singular values exhibited their local maxima. In addition, the mode shapes corresponding to the estimated frequencies were identified to be the expected first and second bending mode shape of a cantilever beam. Regarding the numerical model, the cantilever beam was modeled as an Euler-Bernoulli beam using 8 finite elements, for a total of 16 DOFs. In addition, four lumped masses were added to numerical nodes corresponding to experimental ones, Fig. 7, to take into account the weight of the accelerometers and increase the initial correlation of the F.E. dynamic model. Values of ρ =
2700 kg/m³ and $E = 70$ GPa were considered as starting values for the density and Young’s modulus of the numerical model, respectively. Since two natural frequencies and the corresponding mode shapes were identified in the considered frequency range, ten inputs were considered for the neural networks, that is two frequencies and eight coordinates of mode shapes, four for each mode (in correspondence of location of the accelerometers). The node in correspondence of the joint was not considered. Both density and Young’s modulus of each element were considered as the updating parameters, so sixteen outputs were provided by the neural networks. A feedforward architecture with 20 and 5 units in the hidden layer was considered for the trained and not-trained networks, respectively. Fig. 8 shows trends of MFE and mean MAC for trained and not-trained neural networks. As one can see, trained networks allow to reach convergence faster than not-trained ones. Nevertheless, better results were obtained with
Table 4 shows natural frequency shifts and MAC before and after the updating process. Note that \( f \) refers to experimental natural frequencies, whereas \( \overline{f} \) refers to numerical ones. Excellent results were obtained after the updating especially with the not-trained networks. Indeed, they allow to improve the MAC values whereas the trained networks do not improve the initial correlation between the mode shapes. It is possible to note from Table 5 that trained and not-trained networks behave differently when considering the updated values of the physical properties of the numerical model. Trained networks suggest a maximum variation of about 20% for the density of the 3rd element and a maximum variation of the Young’s modulus of about 38% in the 7th element, with respect to their starting values. On the other hand, not-trained networks give a maximum variation of about 15% and 22% for the density of the 4th element and the Young’s modulus of the 1st element, respectively. Since the parameters variation must have physically meaningful values, not-trained networks allow to achieve more realistic values than those obtained with trained networks on the base of the actual properties of the beam and the boundary conditions considered in the experimental analysis.
Table 5: Experimental case - Updating parameters.

| Element # | Trained NNs | | | Not-trained NNs |
|-----------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|           | $\rho$ [kg/m$^3$] | $\Delta$ $\rho$ [%] | $E$ [GPa] | $\Delta E$ [%] | $\rho$ [kg/m$^3$] | $\Delta$ $\rho$ [%] | $E$ [GPa] | $\Delta E$ [%] |
| 1         | 2825.43     | +4.65       | 71.69      | +2.41        | 2679.88      | -0.75       | 54.33       | -22.39      |
| 2         | 2223.16     | -17.66      | 66.25      | -5.35        | 2896.28      | +7.27       | 76.33       | +9.05       |
| 3         | 3214.62     | +19.06      | 78.08      | +11.55       | 2371.36      | -12.17      | 70.46       | +0.66       |
| 4         | 3005.14     | +11.30      | 70.01      | +0.02        | 3129.09      | +15.89      | 63.70       | -9.00       |
| 5         | 2948.38     | +9.20       | 84.40      | +20.56       | 2611.80      | -3.27       | 74.48       | +6.41       |
| 6         | 2828.13     | +4.75       | 58.21      | -16.84       | 2424.68      | -10.20      | 75.23       | +7.47       |
| 7         | 2667.95     | -1.19       | 96.35      | +37.65       | 2710.16      | +0.38       | 73.19       | +4.56       |
| 8         | 2993.52     | +10.87      | 76.60      | +9.44        | 2324.30      | -13.91      | 74.81       | +6.87       |

4. CONCLUDING REMARKS

In this paper an updating methodology based on neural networks has been proposed and applied to identify the dynamic properties of a simple structure using data coming from operational modal analysis. This kind of analysis allows the identification of the modal parameters of a structure in its operating environment, thus providing an experimental modal model representative of the actual structure. In this study the operational natural frequencies and mode shapes of the structure under investigation have been estimated through the frequency-domain approach called Hilbert Transform Method. Differently from the classical trained networks methods, the proposed methodology uses not-trained neural networks and the updating process is performed by generating a given number of networks and selecting the best one through a probability function. This method goes ahead by minimizing an objective function that contains the model errors, in terms of mean frequencies error and mean of MAC. Simulated and experimental analyses have been performed to demonstrate the accuracy of the proposed updating approach. Results have shown that trained networks work better in simulated cases but they have some limits when experimental operative data are used. In this case not-trained networks give better results especially in terms of the updating parameters that must assume physically meaningful values. Although a simple structure has been used to validate the proposed approach, the analyses performed in this study extend the use of operational data to the neural networks-based updating methodologies.

REFERENCES


