24th COBEM - 2017

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-0583
DAMAGE DETECTION ON ALUMINUM BEAMS USING VIBRATION-BASED METHOD AND ARTIFICIAL NEURAL NETWORKS.

Luísa Rosenstock Völtz
Eduardo Lenz Cardoso
Ricardo De Medeiros
Santa Catarina State University, Department of Mechanical Engineering, Joinville, Brazil.
email: luisa.voltz@gmail.com, eduardo.cardoso@udesc.br, ricardo.medeiros@udesc.br.

Abstract. *Damage detection is a critical engineering area that allows corrective measures to be applied in order to ensure structural safety. Significant efforts have been devoted to developing Non-Destructive Techniques (NDT) for damage identification and prediction in structures. In this work, an Artificial Neural Network (ANN) is proposed to detect damage in metallic beams. An experimental setup is designed in both, intact and damaged, free-free beam boundary condition, excited by an impact hammer, with the response measured by an accelerometer attached to the beam. A vibration-based method using Frequency Response Functions (FRF) and modal parameters are studied and used to train the ANN. A multilayered feedforward neural networks architecture with a learning algorithm is proposed. In addition, experimental and modeling results are performed and compared. The main idea is training and simulating with different setups for the hyperparameters and topology to reach an optimized ANN architecture for damage detection. Therefore, the strategy presented can be helpful in the study of damage detection systems that uses ANN as part of its process.*

Keywords: *Metallic beam, Artificial Neural Networks (ANNs), Damage detection, Vibration-based method, Experimental analysis.*

1. INTRODUCTION

System Identification (SI) is the process of modeling an unknown system, based on a set of input and outputs. These models have applications in Structural Health Monitoring (SHM) systems on different engineering problems, like steel and concrete beams and suspension cables (Sirca and Adelo, 2012). To guarantee the structural integrity, the continuous monitoring must be carried out. Most classes of damage detection methods use vibration characteristics of the structures, including Frequency Response Functions (FRF), natural frequencies and mode shapes (Bandara *et al.*, 2014). In addition, the main idea behind the damage detection techniques, based on structural dynamic changes, is the fact that the modal parameters are functions of the physical parameters like mass, stiffness and damping, and thus, it is reasonable to assume that the existence of damage leads to changes in the modal properties of the structure (Campbell, 2013). Although, there are several techniques to detect damage, some are slow or even requiring physical access to difficult places. Some challenges in the aerospace industry are the constraints on the weight of the Structural Health Monitoring (SHM) system that can be deployed during flight. In the oil industry, SHM systems in offshore platforms is no longer applied because environmental variability was shown to produce significant shifts in the damage-sensitive features from causes such as marine growth and wave motion (Farrar and Worden, 2013). Besides that, another problem is the high preventive and corrective cost of maintenance. The latter one generating the greatest losses in economy and time, in addition to exposure to hazardous situations. To reduce such problems, the use of damage detection techniques is being increasingly investigated and optimized (Fan and Qiao, 2011).

Following the behavior of the biological nervous system, Artificial Neural Networks (ANNs) methodology is an attractive mathematical tool, which can be used to simulate a wide diversity of scientific and engineering problems. Like their biological counterparts, ANNs can learn from examples and can be trained to find solutions of the complex non-linear, multi-dimensional functions relationships without any prior assumptions about their nature (Zhang and Friedrich, 2003). Therefore, the application of ANNs has proved to be an important tool for signal processing, system identification and pattern classification (Puscasu and Codres, 2011).

Recently (Pleune and Chopra, 2000) predicted the fatigue life of carbon and low-alloy steels by training a backpropagation neural network. They showed that ANN has great potential for predicting environmentally assisted corrosion. Another interesting paper (Samanta and Al-Balushi, 2003) presented an ANN with backpropagation and

Levenberg-Marquardt algorithm for fault diagnosis of rolling element bearings using features extracted directly from time-domain vibration signal. The success rate for training was almost 100% of accuracy. Newly, Zago (2016) used ANN to detect damaged in rolling element bearings using Fast Fourier Transform (FFT) applied in time-domain signals as input to the network. Results showed that the neural network was capable of detecting the presence of a failure with more than 99% accuracy. Recently, Fang *et al.* (2005) studied structural damage detection using backpropagation neural network and frequency-domain functions comparing three different learning rate algorithms. Very high accuracy in predict damage location and severity was reached.

Considering the aspects pointed above, this work consists of developing a methodology for SHM systems using an effective ANN algorithm that should be able to identify damages in the beam. Finally, based on comparisons of the modeling results, it is shown a discussion about the potentialities and limitations of the proposed model to be used for supporting SHM systems design.

2. METHODOLOGY

To organize the methodology for a damage detection approach, that yields information of presence of crack and depending on the size of the crack, whether the ANN is able to recognize or not the presence of the damage, a few steps were defined. First, a vibration analysis was done to measure the FRF for damaged and undamaged beams, and then a theoretical background about neural networks was carrying out, followed by the computational implementation. The data was divided into three groups: training, test and validating data. In addition, multilayer feedforward neural network architecture was implementing with others optimization techniques to avoid classical problems associated to ANNs. The proposal is to develop different networks topology configurations, such the number of hidden neurons, nodes and the learning rate, to identify its influence on classification efficiency, and determine the most suitable one for the proposed problem.

2.1 Vibration-based-method

The FRF is an essential frequency measurement that isolates the intrinsic dynamic properties of a mechanical structure, describing the input-output association between two points. Basically, the structure is excited with a known input force and then measures the force and responses on the structure using an accelerometer (He and Fu, 2001). The FRF is defined as

$$H(\omega) = \frac{output\_response}{input\_force} = \frac{X(\omega)}{F(\omega)}. \tag{1}$$

The output data is obtained from 24 beams. Each beam is made of aluminum, with a length of 496 mm and a rectangular cross-section of 19.7 mm *vs.* 4.9 mm. First, experimental tests on free–free beams were carried out. Prior analytical and modeling analysis was performed to find the natural frequencies and nodal points of the beam. For each intact specimen, the dynamic behavior is obtained using an impact hammer. The excitation was applied in two different locations on the beam (position 1 = 0.08 m and position 2 = 0.42 m) and an accelerometer (position = 0.02 m) attached to the beam. In a second step, a drilling machine is used to produce a central hole (position = 0.220 m) on the specimens, the crack depth is 4.9 mm (equal to the thickness of the beam) and different crack diameters were studied: 2, 4 and 8 mm. The damaged beams were analyzed by dynamic tests for each crack diameter.

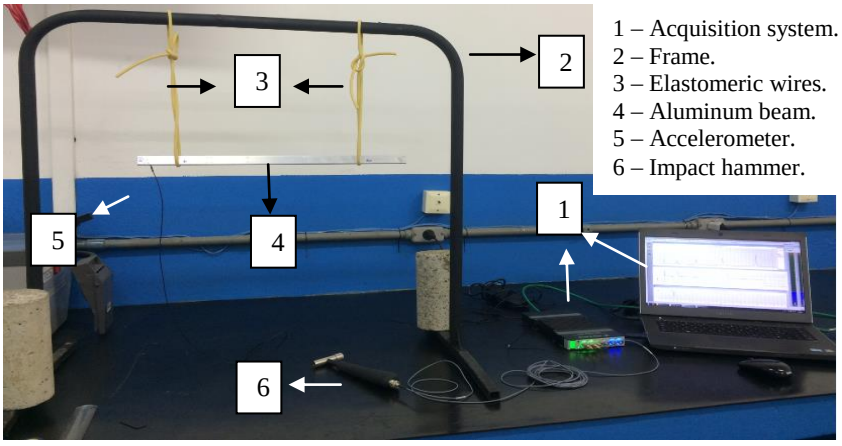

Figure 1. Experimental procedure.

The FRF´s were obtained from Pulse LabShop software by *Brüel & Kjaer*, the accelerometer model used was 4397 (sensitivity 10 mV/g) and the impact hammer model was 8203-006 (sensitivity 1.12 mV/N), both manufactured by *Brüel & Kjaer*. The experimental procedure is shown in Fig. 1. In total 172 FRF´s were obtained with 3201 spectral lines (or frequency points) with a frequency resolution of 0.5 Hz per data point, for each measurement another two were made to make the measurement as realistic as possible. To get the first five vibrational modes, a frequency range of 0-1600 Hz was analyzed. The modes shapes and the positions of the accelerometer, excitation and the crack are shown in Fig. 2.

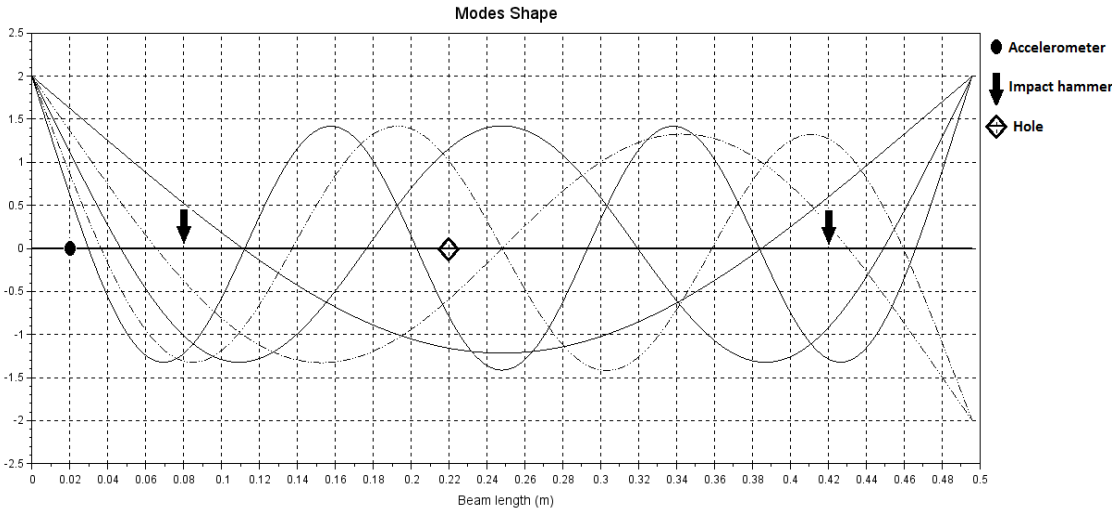

Figure 2. Modes shape of the beam.

2.2 Artificial neural networks

The role of ANN is to provide general non-linear parameterized mappings between a set of inputs and outputs variables (Bishop, 2013). A single artificial neuron model is shown in the Fig. 3, where $x_1,\ldots,x_n$ are the inputs signals (a signal x_j at the input synapse j connected to neuron k). Signals $w_{k1},\ldots,w_{kn}$ are the connective weights responsible for weighting each variable, allowing to quantify its relevance to the functionality of the respective neuron. The total input to a neuron is the weighted outputs from other neurons to it, as Eq. (2). The neuron cannot produce output until the total input exceeds a certain value, vitalizing the neuron. This value is the bias (b_k), or threshold, of the neuron. The bias has the effect of increasing or lowering the net input of the activation function, depending of its signal. Σ is the linear combiner; u_k is the linear combiner output, it´s the result between the linear combiner and the bias, Eq. (3); *g(.)* is the activation function, whose purpose is to limit the neuron output within a range of reasonable values; *y* is the output signal of the neuron, as Eq. (4) (Haykin, 2011).

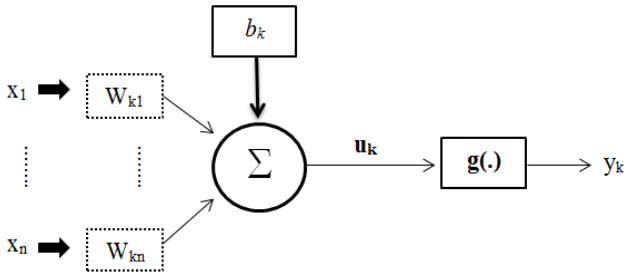

Figure 3. Artificial neuron model.

$$I_k = \sum_{j=1}^{n} x_j w_{kj} \tag{2}$$

$$u_k = I_k + b_k \tag{3}$$

$$y_k = g(u_k) \tag{4}$$

In the literature there are many options of the activation function, as heavyside function, symmetric hard limiter function, linear functions, hyperbolic tangent function, gaussian function and logistics function used in the present work (Silva, *et al*., 2016), as follows

$$g(u) = \frac{1}{1+e^{-u}}. \tag{5}$$

The network architectures are mainly divided in three fundamentally classes: single-layer feedforward networks, multilayer feedforward networks and recurrent networks (Haykin, 2011). One of the most relevant highlights of the ANN is in the ability to learn from its environment, and after the network has learned the relationship between inputs and outputs, it is able to generalize solutions. The set of ordered steps aimed at training the network is called the learning (or training) algorithm, and there is no unique learning algorithm presented in the literature (Silva, *et al.*, 2016). Basically, the difference between them is the way they determine the weights (and bias) on the connections. The outputs (*z)* are compared with the target outputs (*y)* for each (*M)* training samples, and a root-square-error function (*E)* is used in this work

$$E = \sqrt{\sum_{m=1}^{M} (z^m - y^m)^2}. \tag{6}$$

The learning algorithm provides an approximation to the trajectory in weight (and bias) space computed by the method of steepest descent. The main objective of the iterative optimization process is to reach a minimum point for the cost function (*E*), finding the desirable search direction (***d***) to reach this point (Arora, 2004), The search direction can be determined, for each *t* iterations

$$[d]^t = -[\nabla E]^t, \tag{7}$$

where

$$[\nabla E] = \begin{Bmatrix} \partial E / \partial w \\ \partial E / \partial b \end{Bmatrix}. \tag{8}$$

The gradients were evaluated using automatic differentiation, which works by systematically applying the chain rule of differential calculus at the elementary operator level (Kedem, 1980).

One very import parameter in the procedure is the learning rate *(*α*)* that expresses how fast the network training process is being conducted towards its convergence. The learning rate can also be seen as a constant step size to reach the minimum of the cost function along the search direction (Silva, *et al.*, 2016). The smaller we make it, the smaller the changes to the weights (and bias) in the network will be from one iteration to the next, and the smoother will be the trajectory in weight space. If α is too big to accelerate the learning process, the changes in the weights (and bias) may assume a form that the network become unstable. The momentum term (γ) is a parameter to accelerating gradient descent in direction of reduction the error function and can be incorporate to the line search to reach the minimum of the function. The values for momentum term are between $0 \leq \gamma \leq 1$ (Haykin, 2011). The learning rate was calculated using a line search method called Backtracking *(or Armijo-Goldstein)*. The algorithm starts with a value of step size and repeatedly shrinks it by a factor η, called relaxation step, until the Armijo-Goldstein condition is fulfilled. The structure of the algorithm is shown below, where *T* means a transpose, and ρ is the relaxation of the initial slope (Rao, 2009).

Step 1: determine the parameters *(*α,η,ρ*);* set the iteration counter *t=0.*

Step 2: until the condition is satisfied that $E(\boldsymbol{\omega},\boldsymbol{b}) - E((\boldsymbol{\omega},\boldsymbol{b}) + \alpha^t\boldsymbol{d})) \geq -\alpha^t\rho\boldsymbol{d}^T \nabla E(\boldsymbol{\omega},\boldsymbol{b})$, repeatedly increment t and set $\alpha^t=\eta\alpha^{t-1}$.

Step 3: return α^t as the solution.

To calculate the weight and bias, for *t* iterations, the following equations were used.

$$[\Delta w]^t = \gamma[\Delta w]^{t-1} + \alpha^t[d_w]^t \text{ and } [w]^t = [w]^{t-1} + [\Delta w]^t. \tag{9}$$

$$[\Delta b]^t = \gamma[\Delta b]^{t-1} + \alpha^t[d_b]^t \text{ and } [b]^t = [b]^{t-1} + [\Delta b]^t. \tag{10}$$

Pre-processing data is often one of the most significant steps in the development of reliable ANN, and the choice of pre-processing steps can often have an important consequence on generalization performance. One of the most important model of pre-processing involves a reduction in the dimensionality of the input data. At the simplest level, this could involve rejecting a subset of the original inputs. The main reason for dimensionality reduction is that it can support to mitigate the worst effects of the *curse of dimensionality*, which was introduced by Richard Bellman (Bishop,

2013). The phenomenon can be explained basically as the number of training elements required for a classifier to have a good performance is an exponential function of the dimension space of the characteristics (Haykin, 2011).

If 3201 spectral points (raw data) were used as the input to a single large neural network, which would give 3202 adaptive weights (including the bias) for every unit in the first hidden layer. This suggests that a very large training set would be needed to ensure that the weights and bias were well determined, as well huge computational resources would be necessary to find an appropriate minimum of the error function and the problem becomes infeasible.

Principal component analysis can be viewed as a statistical technique for achieving a dimensional reduction. The goal is to map vectors $\boldsymbol{x}^k$ in a *N*-dimensional space $(x_1,\ldots,x_N)$ onto vector $\boldsymbol{z}^k$ in an *P*-dimensional space $(z_1,\ldots,z_P)$, where $P << N$, using an orthogonal projection, where this new set of variables ($\boldsymbol{z}^k$) are called principal components (PCs) (Zang, and Imregun, 2001). Let the matrix $[H(\omega)]_{MxN}$ be formed with all the FRF data, where *M* is the number of training samples (FRF´s) and *N* the spectral points. The procedure to evaluate the PC´s starts with the determination of the mean response matrix $([\overline{H}_j])$ for *j*th column, (Eq. (11)), and evaluating the standard deviation (S_j) using Eq. (12). The FRF matrix can now be replace using Eq. (13) and the new matrix is called response variation matrix, with $\hat{h}_{ij}$ elements. The covariance matrix [*C*] is calculating using Eq. (14) and finally the PCs as obtained from Eq. (15), where *i* is the principal component index with λ_i being the *i*th eigenvalue and $\{\varphi_i\}$ the corresponding eigenvector. The first PC, which is the highest eigenvalue and its associated eigenvector, corresponds to the direction and amount of maximum variability in the raw data (Li, 2011). The new dimension matrix is defined as Eq. (16), with P principal components.

$$[\overline{H_j}] = \frac{\sum_{i=1}^{M} h_{ij}(\omega)}{M}, \; j = 1,2,\ldots,N. \tag{11}$$

$$[S_j] = \sqrt{\frac{\sum_{i=1}^{M} (h_{ij}(\omega) - [\overline{H_j}])^2}{M}} \; . \tag{12}$$

$$\hat{h}_{ij}(\omega) = \frac{h_{ij}(\omega) - [\overline{H_j}]}{[S_j]\sqrt{M}}. \tag{13}$$

$$[C]_{NxN} = \left[\tilde{H}(\omega)\right]^T_{NxM} \left[\tilde{H}(\omega)\right]_{MxN}. \tag{14}$$

$$[C]\{\varphi_i\} = \lambda_i \{\varphi_i\}. \tag{15}$$

$$[A]_{MxP} = \left[\tilde{H}(\omega)\right]_{MxN} [\varphi]_{NxP} \tag{16}$$

3. RESULTS AND DISCUSSIONS

To reduce the dimensionality of the inputs a principal component analysis routine was made in Scilab and applied to the 172x3201 FRF matrix. It was decided to retain 98 % of variance about the raw data. The originals FRF´s for one healthy and damaged beam are shown in Fig. 4, for the first excitation point and in Fig. 5, for the second excitation point. A very carefully visual inspection reveals that there are differences between the FRF´s of the healthy and damaged specimens, especially for the fourth and fifth modes. Nevertheless, there is a significant difference in FRF´s patterns between each beam for the same cases of undamaged or damaged, that's means for all the heaths beams (or damaged) the frequencies for each mode are not exactly the same. The most likely causes are non-uniformity in the manufacturing process for each specimen, some of which may have more void volumes than others, or even low quality during FRF´s measurements. Table 1 shows the frequencies for all the FRF´s acquired. The total of 172 FRF´s, 48 are healthy FRF´s, 44 are damaged with crack size of 2 mm (D_2), 42 are damaged with crack size of 4 mm (D_4) and 38 are with crack size of 8 mm (D_8).

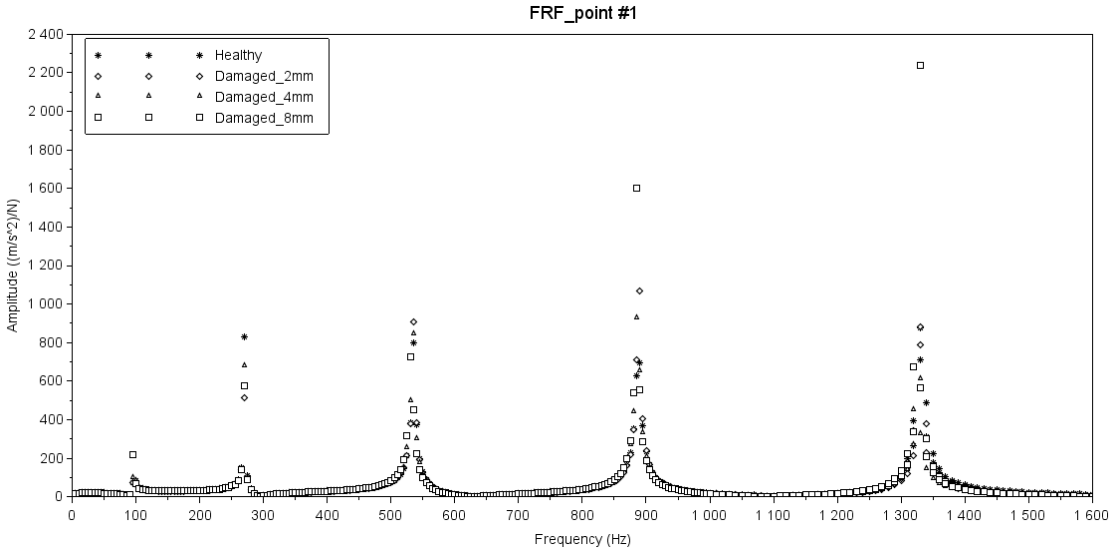

Figure 4. FRF for healthy and damaged beam (excitation applied in point #1).

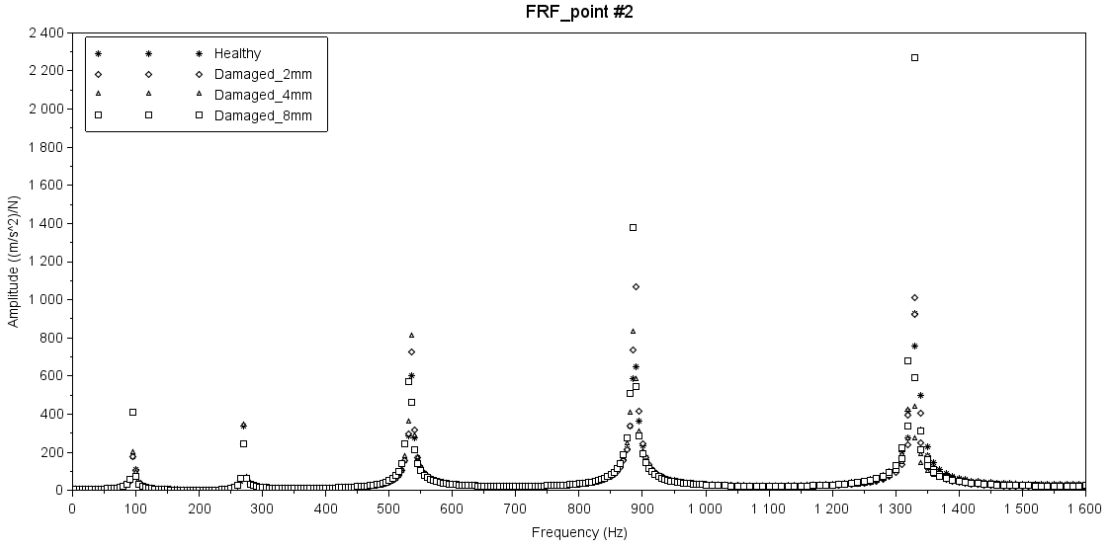

Figure 5. FRF for healthy and damaged beam (excitation applied in point #2).

Table 1. Frequencies range.

Case	Frequency (Hz)				
	Mode 1	**Mode 2**	**Mode 3**	**Mode 4**	**Mode 5**
Healthy	97-98	270-273	534.5-541	888-898	1327.5-1342.5
D_2	96-97.5	269.5-272.5	534.5-540.5	886.5-897	1327-1341
D_4	96.5-97.5	269.5-273	533.5-540	886.5-895.5	1323-1340,5
D_8	95.5-96.5	269.5-272.5	531-538	885-894.5	1324.5-1339

The PCA's curves are shown in Fig. 6, for healthy beam (with excitation applied in position 1 and 2) and the damaged beams (with excitation applied in position 1 and 2).

The neural network consists of an input layer which contains the PCA-compressed FRF´s with the 10 first PC´s, the hidden layer, which process the data and the output layer compounds with four neurons, which are classified as follow: [1 0 0 0] for undamaged, [0 1 0 0] for damaged with 2 mm, [0 0 1 0] for damaged with 4 mm and [0 0 0 1] for damaged with 8 mm. The inputs data were all normalized to have zero mean. The ANN was implemented in Julia language (Bezanson, *et al.*, 2014) using a steepest descent with backtracking line search algorithm and automatic differentiation to calculate the gradients. The classification was considered correct when one of the output neurons returned a value higher than 0.90.

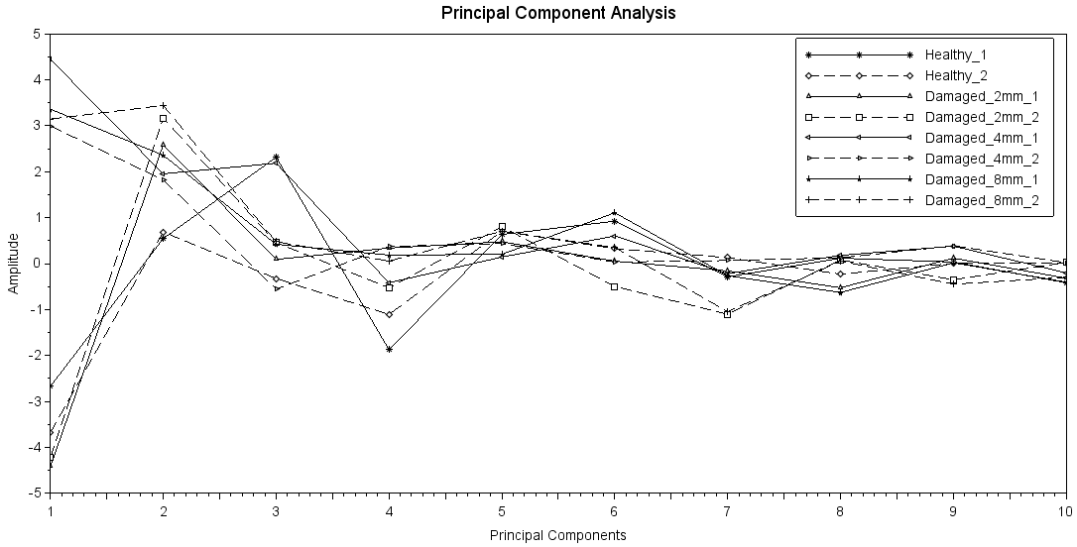

Figure 6. PCA's curves for healthy and damaged beam (excitation applied in point #1 and #2)

3.1 Damage detection for healthy beam and damaged beam with 8mm crack size.

Considering the 86 FRF´s, healthy and damaged with 8 mm crack size, 70 were used for training and testing and 16 for validating the ANN, as are shown in Tab. 2. The topology used was 10 inputs in the first layer, 8 and 6 neurons in the first and second hidden layers, respectively, and 4 neurons in the output layer. The momentum term was fixed at 0.7. The behavior of the cost function and efficiency (for training and testing) *versus* iterations are shown in Fig. 7. After 200 iterations, the network is stable and well trained with training error about 0.00192. In validating phase the ANN classified all the data set with 100 % of assertiveness.

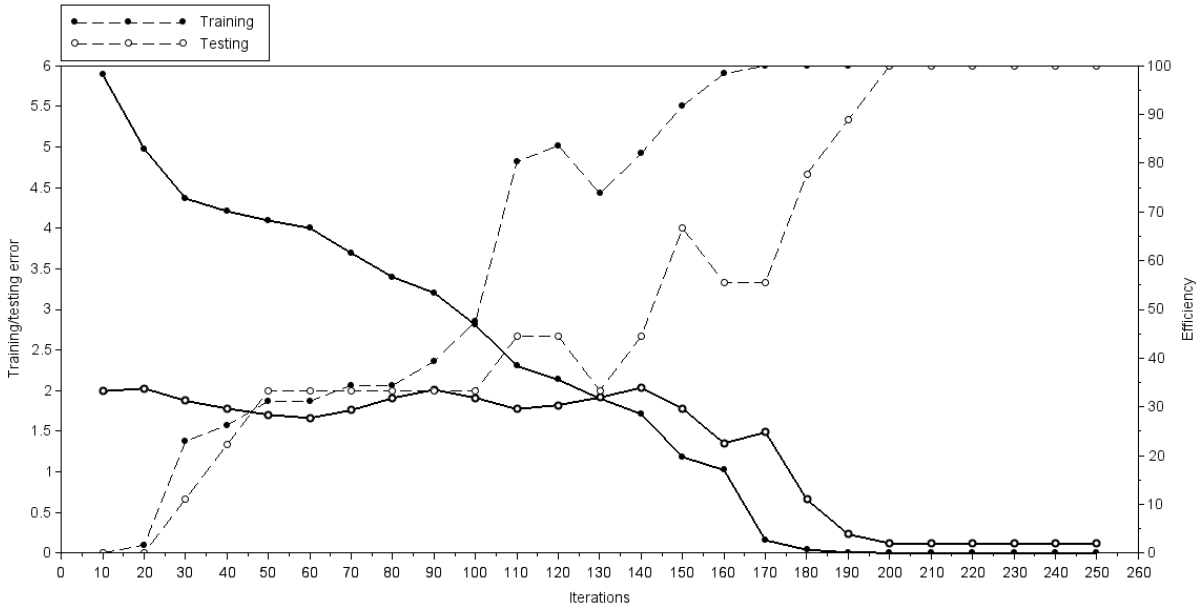

Figure 7. Errors functions (solid lines) and efficiency (broken lines) x iterations for training and testing.

Table 2. Data set used for ANN (intact x 8 mm crack size)

Data set	Intact	D_8	*Total*
Training set	34	27	*61*
Testing set	5	4	*9*
Validating set	9	7	*16*

3.2 Damage detection for healthy beam and damaged beams with 4 mm and 8 mm crack size.

Considering the 128 FRF´s, healthy and damages with 4 mm and 8 mm crack size, 113 were used for training and testing and 25 for validating the ANN, as shown in Tab. 3. The topology and parameters was the same used for health and damaged with 8 mm crack size. The behavior of the cost function and efficiency (for training and testing) *versus* iterations are shown in Fig. 8. After 1800 iterations the network is stable with training error about 0.1809. As can see during testing, the network showed efficiency around 77 %, after some more iterations the ANN doesn't improve the efficiency. In contrast, during validating phase the ANN classified 88 % of patterns. The output signals that had larger errors are shown in Tab. 4 and Tab. 5, where it is noted that the neural network erroneously classified three samples for testing phase and two samples for validating phase. It is also noted that the ANN was unable to classify one sample during validating.

Table 3. Data set used for ANN (intact x 4 mm x 8 mm crack size)

Data set	Intact	D_4	D_8	*Total*
Training set	34	29	27	*90*
Testing set	4	5	4	*13*
Validating set	10	8	7	*25*

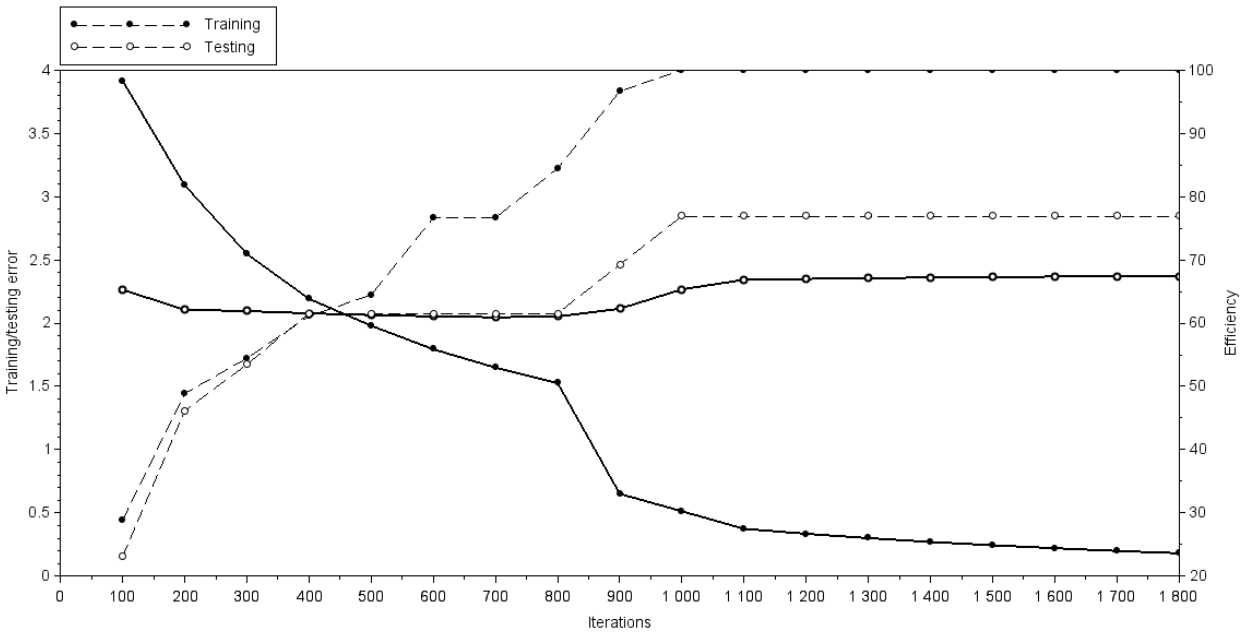

Figure 8. Errors functions (solid lines) and efficiency (broken lines) x iterations for training and testing.

Table 4. ANN pattern recognition misclassified - testing phase.

ANN pattern recognition	**Real pattern**
Damaged with 4 mm	Intact
Intact	Damaged with 4 mm
Damaged with 4 mm	Damaged with 8 mm

Table 5. ANN pattern recognition misclassified - validating phase.

ANN pattern recognition	**Real pattern**
Damaged with 4 mm	Intact
Intact	Damaged with 8 mm
Don't classify	Damaged with 8 mm

3.3 Damage detection for healthy beam and damaged beams with 2 mm, 4 mm and 8 mm crack size.

Considering the 172 FRF´s, 141 were used for training and testing and 31 for validating the ANN, as shown in Tab. 6. The topology used was 10 inputs in the first layer, 7 and 6 neurons in the first and second hidden layers, respectively and 4 neurons in the output layer. The momentum term was fixed at 0.8. The behavior of the cost function and efficiency (for training and testing) *versus* iterations are shown in Fig.9. As can be seen during testing, the network showed efficiency around 66%, and after some iterations the error become stable. The maximum efficiency for training set was 93.5 %. During validating phase, the ANN correctly 77% of patterns. The output signals that had larger errors are shown in Tab. 7 and Tab. 8, where it is noted that the neural network erroneously classified one sample for testing phase and three samples for validating phase.

Table 6. Data set used for ANN (intact x 2 mm x 4 mm x 8 mm crack size)

Data set	Intact	D_2	D_4	D_8	*Total*
Training set	34	31	31	27	*123*
Testing set	4	5	5	4	*18*
Validating set	10	8	6	7	*31*

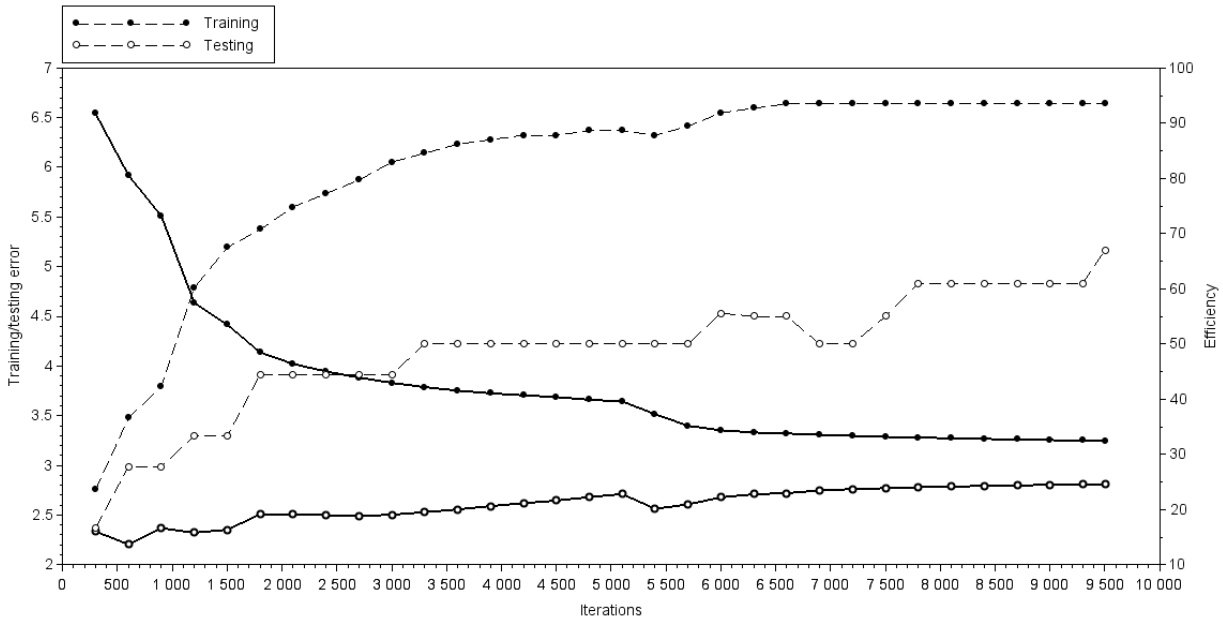

Figure 9. Errors functions (solid lines) and efficiency (broken lines) x iterations for training and testing.

Table 7. ANN pattern recognition misclassified - testing phase.

ANN pattern recognition	**Real pattern**
Damaged with 2 mm	Intact
Don't classify	Damaged with 2 mm
Don't classify	Damaged with 4 mm
Don't classify	Damaged with 4 mm
Don't classify	Damaged with 4 mm
Don't classify	Damaged with 4 mm

Table 8. ANN pattern recognition misclassified - validating phase.

ANN pattern recognition	**Real pattern**
Damaged with 2 mm	Intact
Intact	Damaged with 2 mm
Damaged with 2 mm	Intact
Don't classify	Damaged with 2 mm
Don't classify	Damaged with 2 mm
Don't classify	Damaged with 2 mm
Don't classify	Damaged with 4 mm

4. CONCLUSIONS

In this work, an ANN for damage detection in metallic beams is studied. A wide variety of network architectures and parameters were studied, their influences on the effectivity of the procedure are analyzed, and the outstanding solutions were present in this work. The results, especially when the crack size is 8 mm, show that the combination of vibration methods with reduction of dimensionality through PCA´s, and the use of ANN brings an adequate methodology for damage detection. Despite the overall efficiency of 82.5 % for cases with 4 mm and 8 mm crack size, the methodology was also satisfactory. Some improvements in the methodology should be made in case the damage size is very small, or even the application of other more precise techniques for dynamic measurements, as it is observed in the FRF's the differences between frequencies, for the case of damage with 2 mm, is almost imperceptible. Even a robust neural network may not be able to classify correctly, or even not to classify. When there is no clear distribution of the input data in a neural network, the problem with high variance becomes a major barrier in pattern recognition, the network overfitting quickly and instead of learning during iterations it ends up by memorizing the data showing low efficiency in the test and validation phase.

5. REFERENCES

Arora, J.S, 2004. *Introduction to optimum design*. UK, 2nd edition.

Bandara, R. P., Chan, T.H.T. and Thambiratnam, D. P., 2014. Structural damage detection method using frequency response functions. *Structural Health Monitoring.* Vol 13, Issue 4, 418 – 429.

Bezanson, J., Edelman, A., Karpinski, S. and Shah, V.B., 2014. Julia: A fresh approach to numerical computing. *CoRR*, abs/1411.1607. Disponível em: <http://arxiv.org/abs/1411.1607>. Acesso em: 15 maio. 2017.

Bishop, C. M., 2013. *Neural networks for pattern recognition*. Oxford University Press. UK.

Campbell, F.C., 2010. Structural composite materials. *ASM International*. EUA.

Fan, W. and Qiao, P., 2011. Vibration-based damage identification methods: a review and comparative study. *Structural Health Monitoring*. Vol 10, Issue 1: 83-111.

Fang, X, Luo, H and Tang, J., 2005. Structural damage detection using neural network with learning rate improvement. *Computers and Structures*, 83: 2150-2161.

Farrar, C. R., Worden K., 2013. *Structural health monitoring: A Machine Learning Perspective*. John Wiley & Sons.

Haykin, S., 2001. *Neural networks: a comprehensive foundation*. Translate by Paulo Engel. 2nd edition. Porto Alegre: Bookman.

He, J. and Fu. Z. F., 2001. Modal analysis. Butterworth-Heinemann.

Kedem, G., 1980. Automatic differentiation of computer programs. *ACM Transactions on Mathematical Software Journal*. Volume 6 Issue 2:150-165.

Li, J., Dackermann, U., Xu, Y.L. and Samali, B., 2011. Damage identification in civil engineering structures utilizing PCA-compressed residual frequency response functions and neural networks ensembles. Structural Control Health Monitoring. 18:207-226.

Pleune, T.T, Chopra, O.K., 2000. Using artificial neural networks to predict the fatigue life of carbon and low-alloy steels. *Nuclear Engineering and Design*. 197:1-12.

Puscasu, G and Codres B., 2011. Nonlinear system identification and control based on modular neural networks. *International Journal of Neural Systems*. Vol 21, 4: 319-33.

Rao, S.S., 2009. *Engineering optimization: theory and practice*. John Wiley & Sons. 4th edition. CA.

Samanta, B. Al-Balushi, K.R., 2003. Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mechanical Systems and Signal Processing*. 17(2), 317-328.

Silva, I.N, Spatti, D.H. and Flauzino, R., 2016. *Artificial neural networks*. 2nd edition. São Paulo: Artliber.

Sirca, G.F. and Adelo, H., 2012. System identification in structural engineering. *Scientia Iranica A*. 19(6):1355-1364.

Zago, J.A., 2016. Failure detection in rolling bearings through vibration analysis and artificial neural networks. Department of Mechanical Engineering. Santa Catarina State University.

Zang, C. and Imregun, M., 2001. Combined neural network and reduced FRF techniques for slight damage detection using measured response data. *Applied Mechanics* 71 525-536.

Zhang, Z and Friedrich, K., 2003. Artificial neural networks applied to polymer composites: a review. *Composites Science and Technology* 63:2029-2044.

6. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.