Computer Assisted Mechanics and Engineering Sciences, **18**: 191–203, 2011.

Application of neural networks for structure updating

Bartosz Miller
Department of Structural Mechanics, Rzeszów University of Technology
W. Pola 2, 35-959 Rzeszów, Poland
e-mail: bartosz.miller@prz.edu.pl

The paper presents the application of Artificial Neural Networks (ANNs) for finite element (FE) models updating. The investigated structures are beams and frames, their models are updated by ANNs with input vectors composed of dynamic characteristics of structures measured on laboratory models. The ANNs (multi layer feed-forward networks and Bayesian neural networks) are trained on numerical data disturbed by an artificial noise. The responses of the structures are measured on laboratory models. The updating procedure is also applied in identification of defects or additional masses attached to the structure.

Keywords: artificial neural networks, updating, dynamics, vibrations, identification.

1. Introduction

1.1. Problem description

The progress that takes place nowadays in both the development of new computational tools and construction of new, powerful computer hardware makes it possible to build very complicated dynamic models. The time required to simulate the response of a structure to static or dynamic excitations is being reduced. Unfortunately, even those very complicated models, with the number of degrees of freedom (DOF) exceeding hundreds of thousands, are in some cases not able to handle the present-day requirements. In some particular cases it is necessary to employ models which are able to simulate or reproduce the behaviour of real structures with very high accuracy. In order to minimize the differences between the results of experimental measurements and numerical simulations some changes can be introduced to the numerical model. The introduction of those changes is called *updating* [4, 7, 8, 10, 12].

Updating is a complex task, in [4] it is described as "the improvement of an inaccurate model by using imprecise and incomplete measurements". The inaccuracy of a computational model may result from various sources including the assumptions and limitations of the employed modelling method and the assumption of the materials homogeneity. The imprecision and incompleteness of measurements results from the disturbances of the measured signal, measurement noise and impossibility to measure all degrees of freedom of the structure in the full frequency range (see [4]).

In some specific cases the aim of model updating is to obtain a model able to reproduce the results of measurements with a specified accuracy (or even without any error), this kind of updating is achieved by the introduction of some changes directly into the stiffness and mass matrices. The so-called *direct methods* require very precise experimental data in order to avoid the reproduction of measurement noise or errors. The values of updated parameters are calculated according to known relations of the value of the model error and the values of the parameters updated. However, derivation of those formulas is very time-consuming and in some cases direct methods require more calculations than other methods. The main disadvantage of direct methods is their inability to interpret physically the introduced parameter changes, so the updated models can not be applied to defect identification. Model errors are corrected by changing all elements of stiffness and mass matrices, and in the majority of cases sparse stiffness and mass matrices are replaced by dense ones. To avoid this problem some direct methods change only the initially non-zero elements, e.g. by computation of relative changes of the elements.

The model updating approach applied herein is based on careful selection of control parameters of a model (Young modulus, mass density, joint rigidity, support stiffness, etc.) and iterative "improvement" of the parameters. The parameters to be updated should be selected according to the following rules:

- from among all the considered model parameters they are estimated with the highest error,
- the measured structure responses are sensitive to these parameters changes.

The parameters changes introduced have continuous physical interpretation and the updated model can be applied to simulate the behaviour of the structure in different circumstances.

Since updating is based on the measurement data, the experiment is a crucial point of the whole updating procedure. Unfortunately, in the majority of cases it is impossible to measure the response of the structure to external excitation in all the points necessary from the point of view of applied updating technique, especially the measurements of rotational degrees of freedom cause many difficulties [3]. Moreover, a significant number of degrees of freedom may be inaccessible. The amount of the measurement data may be insufficient to perform the updating of the whole model, so the computational model must be reduced, or – not often – the measurement model is expanded. Both approaches enable updating of the computational model but they introduce additional errors. However, in some cases the amount of measurement data is so small that the updating of reduced model is still impossible because the deterministic methods require an adequate amount of input data.

The accuracy of the updated model is determined by the penalty function depending on the eigenfrequencies and eigenvectors of both computational and experimental models. A search for the minimum of penalty function is carried out iteratively adopting a non-linear penalty function (most often sum square error function), which is in each stage approximated by the linear function. The main problems that arise during the iterative updating are connected with the necessity of comparison of the results of numerical simulations and experimental measurements.

The correlation of eigenfrequencies is assessed herein by the Root Mean Square Error (RMSE^n), Maximal Relative Error (MRE^n) and/or Average Relative Error (ARE^n), all calculated for the first n eigenfrequencies:

$$\mathrm{RMSE}^n = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i^m - f_i^c)^2}\,, \tag{1}$$

$$\mathrm{MRE}^n = \max_{i=1,\ldots,n}\left|\frac{f_i^m - f_i^c}{f_i^m}\times 100\%\right|, \tag{2}$$

$$\mathrm{ARE}^n = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{f_i^m - f_i^c}{f_i^m}\times 100\%\right|, \tag{3}$$

where f_i^m is the i-th eigenfrequency obtained from measurements, f_i^c is the i-th eigenfrequency obtained from the FE model, n is the number of considered eigenfrequencies.

The paper presents updating of computational models on the basis of a limited number of measurement data. As a tool for the model updating, different types of Artificial Neural Networks (ANN) are applied, see [7, 8]. ANNs in [1, 5] are the so called *soft* computational methods as they were initially formulated due to inspiration from the behaviour of the human brain. Their architecture and principle of operation have been considerably changed lately, but the ANNs still have to be *trained* in order to make a right decision on the basis of incomplete and even faulty data. In recent years, the ANNs have been increasingly used in structural mechanics and civil engineering [14, 15, 17–19].

1.2. Standard artificial neural networks

The type of standard ANN adopted in this study is called Multi Layer Perceptron (MLP) [5]. Its architecture adopted herein can be described as $I - H - O$, where I is the number of inputs, H the number of neurons in the hidden layer, O their number in the output layer. Each neuron performs a linear transformation (by means of coefficients called weights and biases, all gathered here in vector $\mathbf{w}$) followed by a non-linear (e.g. sigmoidal) transformation before signal transmission to the subsequent layer. Therefore, the dimension of vector $\mathbf{w}$ can be specified by the number of network parameters $I \times H + H \times O + H + O$.

The calibration of the MLP to operation is called "training" or "learning" and is based here on P vector pairs called patterns composed of a vector of selected model parameters $\boldsymbol{\gamma}$ and corresponding vector of FE model response $\mathbf{r}$ disturbed by an artificial noise mimicking measurements errors. The computation of the optimal vector $\mathbf{w}_{opt}$ can be formulated as follows:

$$\mathbf{w}_{opt} = \arg\min_{\mathbf{w}} \{\Omega(\mathbf{w})\}, \tag{4}$$

where

$$\Omega(\mathbf{w}) = \frac{1}{2P}\sum_{j=1}^{P}\sqrt{(\boldsymbol{\gamma}_j - \mathbf{y}_j)^T(\boldsymbol{\gamma}_j - \mathbf{y}_j)} \tag{5}$$

and $\mathbf{y}_j = \mathrm{MLP}(\mathbf{r}_j, \mathbf{w})$ is the approximation of vector $\boldsymbol{\gamma}_j$ given by MLP. In order to formulate well-posed problem of network weights optimization the number of patterns P at disposal should exceed the number of network parameters collected in vector $\mathbf{w}$.

The optimal vector $\mathbf{w}_{opt}$ can be computed by means of mathematical programming procedures, in the paper the Levenberg-Marquardt algorithm (see [5]) is applied.

The architecture of MLP (namely, the number of inputs I, hidden neurons H and output neurons O) is partially determined by the problem to be solved, since the number of inputs and outputs must equal the number of the parameters in the input and output vectors, respectively. The number of hidden neurons is obtained as a result of a cross-correlation procedure, where for fixed inputs I and outputs O the number of hidden neurons varies from a reasonably small value (in the paper $H_{\min} = 2$) to a reasonably high value (in the paper $H_{\max} = 30$). For each value of H the network is trained, the network error Ω (e.g. standard mean square error of the identification) is computed and the optimal value of H is selected as the smallest H for which its increase gives stabilization or even increase of the network error.

The accuracy of trained MLP (the one with optimal parameters) is verified using a second set of patterns ("testing" patterns), which should not be involved in the learning process. If the errors defined as the difference between targets $\mathbf{t}_j$ and $\mathbf{y}_{j,\mathrm{MLP}}(\mathbf{x}_j, \mathbf{w}*)$ (say mean square) are at an acceptable level, the network can be used routinely in the operational phase for real-life problems of structural diagnosis.

1.3. Bayesian neural networks

Besides traditional MLP also a very promising, new type of neural networks, namely Bayesian Neural Networks (BNN) [1, 11], are employed. BNNs are probabilistic networks based on *Bayes' theorem*:

$$p(\mathbf{w}|\mathbf{t}, \alpha, \beta) = \frac{p(\mathbf{t}|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\alpha, \beta)}, \tag{6}$$

where the denominator in Eq. 6, called *evidence* (or *marginal likelihood*), is defined as the following integral:

$$E = p(\mathbf{t}|\alpha,\beta) = \int_{\mathcal{R}^W} p(\mathbf{t}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)d\mathbf{w}. \tag{7}$$

The maximum of the logarithm of evidence with respect to vector $\mathbf{w}$ is evaluated

$$\overset{\max}{\mathbf{w}}|\ln E = \overset{\max}{\mathbf{w}}|\ln p(\mathbf{t}|\alpha,\beta) \tag{8}$$

and a new criterion called *Maximum Marginal Likelihood* (*MML*) is formulated, see references in [16] and Appendix in [6]. This criterion can be applied for design of network architecture instead of the cross-validation method [14, 16].

In BNNs the overfitting phenomenon is controlled by a regularization term $\frac{\alpha}{2}\sum_{i=1}^{W}\mathbf{w}_i^2$ introduced into the network cost function:

$$F(\mathbf{w}) = \frac{\beta}{2}\frac{1}{P}\sum_{j=1}^{P}\sqrt{(\gamma_j-\mathbf{y}_j)^T(\gamma_j-\mathbf{y}_j)} + \frac{\alpha}{2}\sum_{i=1}^{W}\mathbf{w}_i^2\,. \tag{9}$$

In the so-called *Semi*-Bayesian Neural Networks (SBNNs) *hyperparameters* α and β can be computed during network learning in the iterative way, while in *True*-Bayesian Neural Networks (TBNNs) they have to be *marginalized* (the integrals over weight vector components $\mathbf{w}_i$ have to be computed). In the paper both types of BNNs are used.

In the BNN Netlab package [11] commonly used the recommended procedure of SBNNs training is Scaled Conjugate Gradient (SCG) optimization.

1.4. Neural network updating procedure

The paper presents the updating of some computational models of engineering structures. The updating algorithm consists of the following steps:

1. **Direct analysis** of FE model is related to generating a set of patterns

   $$\mathcal{P}' = \left\{(\boldsymbol{\gamma},\mathbf{r})^{(p)}\,|\,p=1,\ldots,P\right\}, \tag{10}$$

   where $\boldsymbol{\gamma}$ – vector of control parameters, $\mathbf{r}$ – vector of FE model response (e.g. eigenfrequencies) as mapping $\boldsymbol{\gamma}\to\mathbf{r}$ for all the patterns p.

2. **Disturbance** of vectors $\mathbf{r}$ by an artificial random noise in order to mimic measurement errors.

3. **Inverse analysis** is related to the training and testing of a neural network using a set of patterns (10 but with inverse input and output vectors

   $$\mathcal{P} = \left\{(\mathbf{x}=\mathbf{r},\mathbf{t}=\gamma)^{(p)}\,|\,p=1,\ldots,P\right\}. \tag{11}$$

   The set $\mathcal{P}$ is split into training and testing sets $\mathcal{L}$ and $\mathcal{T}$, where $\mathcal{P}=\mathcal{L}\cup\mathcal{T}$, $\mathcal{L}\cap\mathcal{T}=\varnothing$, which are used for designing of a neural network.

4. **Calibration** of control parameters is performed by the trained neural network exploring responses $\mathbf{r}_{\text{exp}}$ measured on an empirical model

   $$\gamma_{\text{ident}} = \mathbf{y}_{\text{NN}}(\mathbf{r}_{\text{exp}}). \tag{12}$$

5. **Verification** of calibrated parameters by substitution of $\boldsymbol{\gamma}_{\text{ident}}$ into the FE model to compute its responses $\mathbf{r}_{\text{FEM}}$ and compare them with the measured responses $\mathbf{r}_{\text{exp}}$

$$\mathbf{r}_{\text{FEM}}\left(\boldsymbol{\gamma}_{\text{ident}}\right) - \mathbf{r}_{\text{exp}} = \varepsilon_{\mathbf{upd}}. \tag{13}$$

The procedure was initially tested on numerical data only (see [9, 18]). In this paper the updating method involving experimental data is presented.

All MLP simulations presented in the paper were performed using Matlab package with Neural Network Toolbox [2], SBNN computations were performed using Netlab Toolbox [11] and TBNN were performed using MCMCstuff toolbox [13].

2. Beam on two slings

The first investigated structure was a beam (see Fig. 1a) made of an aluminium alloy and suspended on two elastic slings. The known material properties of the aluminium alloy were volume mass density $\rho_V = 2743$ kg/m^3 (calculated as the ratio of the mass of the beam and its volume), Poison's ratio $\nu = 0.33$. Young's modulus was estimated by the updating method.

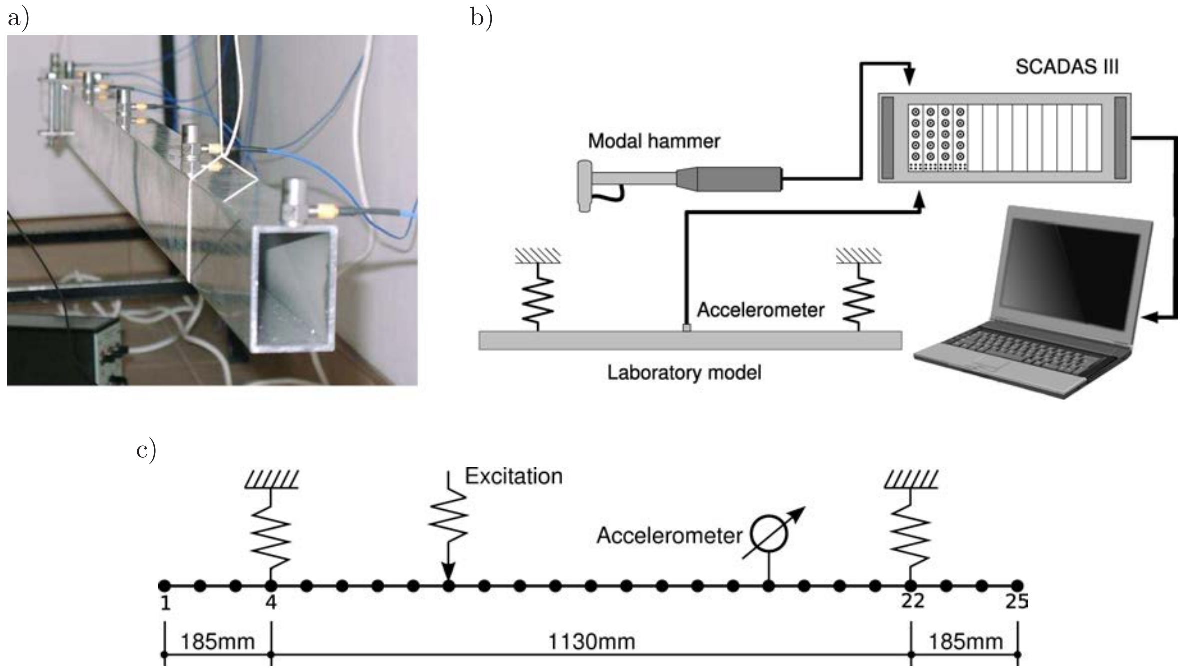

Fig. 1. The investigated beam: a) laboratory model, b) measurement set-up, c) FE model.

FE model, consisting of 24 beam elements, had 50 DOFs (see Fig. 1c). The influence of shearing forces was not negligible so the Timoshenko model of the beam was applied. The acquisition set-up applied is shown in Fig. 1b.

2.1. Updating of the initial model

The considered FE model (see Fig. 1c) had two free parameters: Young's modulus E and the shape correction factor k_s of Timoshenko beam. As the input information for the updating procedure the eigenfrequencies were used, the networks updating the dynamic model had five inputs (the first five eigenfrequencies $\mathbf{x} = \{f_1, f_2, f_3, f_4, f_5\}$).

The vibrations of the laboratory model were excited by an impact. The acquired signals were the reference signal measured by the force sensor built into the impact hammer and the response of the structure measured by an accelerometer (only one in this example, later on six accelerometers were used). Both the impulse excitation and the accelerometer were at nodes of the FE model.

The output vector was adopted as follows: $\mathbf{y} = \{E, k_s\}$ (Young's modulus and the shear correction factor), the obtained values of Young's modulus $E = 70.77$ GPa and the shear correction factor $k_s = 0.5642$ were applied to the dynamic model in order to obtain the updated model. The differences between eigenfrequencies obtained from the updated model and from the laboratory measurements were less than 1%.

2.2. Updating of the beam with an additional mass

A mass of 273.2 g was attached to the beam. In the FE model the additional mass was modelled using a new finite element in the place corresponding to the mass location.

In order to obtain precise measurements of eigenfrequencies for different locations of the mass, five more accelerometers (altogether six) were attached to the beam. The input vector consisted of the first five eigenfrequencies: $\mathbf{x} = \{f_1, f_2, f_3, f_4, f_5\}$. The output vector $\mathbf{y} = \{\rho_L^e, k_s^m\}$ consisted of the values of linear mass density of the element describing the additional mass ρ_L^e and the element shear correction factor k_s^m. The results obtained from 5-7-2 MLP are listed in Table 1, the errors of updated model are: $\mathrm{RMSE}^5 = 4.5Hz$, $\mathrm{MRE}^5 = 0.75\%$ and $\mathrm{ARE}^5 = 0.40\%$.

Table 1. The errors of eigenfrequencies prediction (beam with and additional mass and six accelerometers).

Eigenfrequencies	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	f_4 [Hz]	f_5 [Hz]
Measured	168.13	436.22	796.28	1283.9	1891.7
Computed	166.88	434.85	795.39	1293.55	1889.73
Errors [%]	0.74	0.31	0.11	-0.75	0.10

The accuracy of the updated model (with ρ_L^e and k_s obtained from MLP) was tested using the data obtained from the measurements of the beam with the mass attached in a different place (the 11^{th} node of FE model), the errors obtained from the model were: $\Delta f_1 = 1.42\%$, $\Delta f_2 = 1.20\%$, $\Delta f_3 = 0.85\%$, $\Delta f_4 = -0.47\%$, $\Delta f_5 = -0.49\%$, and $\mathrm{RMSE}^5 = 6.35Hz$, $\mathrm{MRE}^5 = 1.42\%$ and $\mathrm{ARE}^5 = 0.89\%$.

The MLPs of architecture 5-h-1 were also trained to locate the additional mass. The learning and testing patterns were obtained from the numerical simulations performed for an additional mass located in turn in all nodes of the FE model between the elastic link and the symmetry axis of the beam. The trained MLPs were verified by the data obtained from the laboratory measurements of the beam with the mass attached in 6^{th} or 11^{th} node (see Fig. 1c). The best results were obtained from 5-13-1 MLP which was able to correctly locate the mass in the 6^{th} node, the mass located in the 11^{th} was located with the error not exceeding the length of one element of the FE model.

2.3. Updating of impaired beam

The defects were introduced in stages on the length corresponding with the length of three final elements:

i) half of the thickness of both flanges removed on the length of one final element, two elements undamaged,

ii) half of the thickness of both flanges removed on the length of two final elements, one element undamaged,

iii) half of the thickness of both flanges removed on the length of all three elements,

iv) both flanges completely removed on the length of one element (see Fig. 2), two elements with half of the thickness of both flanges removed,

v) both flanges completely removed on the length of two elements, one element with half of the thickness of both flanges removed,

vi) both flanges completely removed on the length of all three elements.

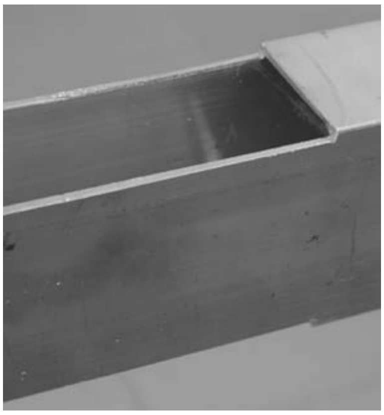

Fig. 2. The defect, both flanges completely removed.

In case i) half of the thickness of both flanges was removed between nodes 19 and 20 (see Fig. 1). The parameters that were updated were the beam volume mass density ρ_V^* (including the masses of six accelerometers and the influence of connecting cables) and the shear correction factor of the impaired cross-section k_s^α. The depth of the defect and its width were measured and considered to be known, the values of Young's modulus E and shear correction factor k_s of the original cross-section were taken from the previous updating cases.

The input vector consisted of five eigenfrequencies ($\mathbf{x} = \{f_1, f_2, f_3, f_4, f_5\}$) disturbed by an artificial noise, the output vector consisted of the volume mass density and the shear correction factor of the impaired cross-section ($\mathbf{y} = \{\rho_V^*, k_s^\alpha\}$). The results of identification were verified using the data obtained from the measurements of the beam with defects in cases ii) and iii) (development of the defect of the same kind), for the results see Table 2.

Table 2. The errors of eigenfrequencies prediction (beam with defect).

Defect case	Δf_1 [%]	Δf_2 [%]	Δf_3 [%]	Δf_4 [%]	Δf_5 [%]	RMSE^5 [Hz]	MRE^5 [%]	ARE^5 [%]
i)	0.7	0.0	−0.6	0.0	0.0	2.24	0.7	0.26
ii)	0.5	−0.3	−0.3	0.7	1.4	13.04	1.4	0.64
iii)	0.4	−0.3	0.5	1.6	15.71	15.71	1.6	0.83
iv)	2.4	−0.9	0.1	1.6	1.3	14.31	2.4	1.24
v)	2.4	−0.1	1.2	1.6	−0.4	11.00	2.4	1.15
vi)	−1.1	−1.2	0.6	0.6	−1.1	9.74	1.2	0.92

In case vi) (both flanges were removed on the length corresponding with the length of three finite elements of the dynamic model, see Fig. 2) the shear correction factor k_s^ζ of the cross-section with completely removed flanges was identified. Cases iv) and v), when both types of damage occurred, were used to verify the updated model. The relative errors of eigenfrequencies prediction using the updated FE model are given in Table 2.

The errors of the prediction of eigenfrequencies of the beam with different defects are smallest in the cases, from which the data were used during the MLP training (cases i) and vi)). In other cases the errors were higher, the maximal MRE error was obtained in cases iv) and v) (with both types of damage occurring) and equalled 2.4%. In the updating on the basis of experimental data this error is rather small and the model should be considered as properly updated in the range of the first five eigenfrequencies.

3. Two-storey frame

The investigated structure was two-storey frames (see Fig. 3) made from aluminium alloy (volume mass density $\rho = 2743$ kg/m^3, Young's modulus $E = 70.77$ GPa, Poison's ratio $\nu = 0.3$). The height of both frames was 40 cm, the width was 46.9 cm, the beams and columns were of a rectangular cross-section 2.6 cm by 0.6 cm. The columns were bolted to steel plates mounted in the ground. Due to different heights of the plates the rigidity of both footings was different and the effective length of the columns was between 30 cm (the length of the columns above the steel connection plates) and 40 cm (overall length of the columns).

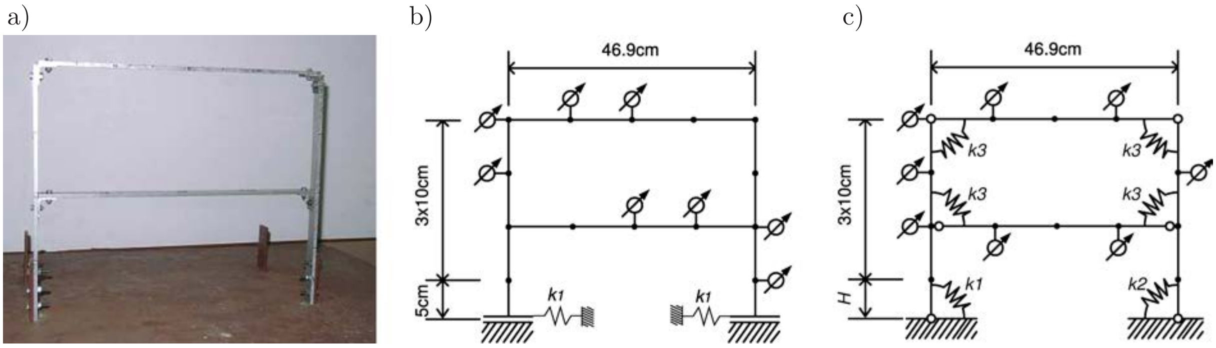

Fig. 3. Portal frame: a) laboratory model, b) FRAME3, with springs k_1 in both footings , c) FRAME4, with flexible joints k_1, k_2, k_3 and the height of the model H.

The first model, called FRAME3, had two parameters for updating, namely the horizontal stiffnesses of both columns footings k_1 and k_2. The updating of both stiffnesses was performed by the 4-h-2 neural network, where the input and output vectors were defined as: $\mathbf{x} = \{f_1, f_2, f_3, f_4\}$, $\mathbf{y} = \{k_1, k_2\}$. The relative errors of eigenfrequencies obtained from the updated model are as follows: $\Delta f_1 = 5.45\%$, $\Delta f_2 = 4.28\%$, $\Delta f_3 = 1.04\%$ and $\Delta f_4 = 1.63\%$.

The errors of eigenfrequencies prediction by the updated model were as follows: RMSE^4 = 2.43 Hz, MRE^4 = 5.45% and ARE^4 = 3.10%. The accuracy of prediction of higher eigenfrequencies and eigenmodes were rather poor. After trying a number of different FE models the cause of the difficulties was found as: the footings of both columns in the laboratory model were made rather imprecisely. The laboratory model was built again (with all the geometrical and material parameters the same as for the previous model) and a new FE model of the portal frame (see Fig. 3c) was constructed.

All the eigenfrequencies obtained from FE simulations were disturbed by an artificial random noise with mean $\mu = 0.0$ and standard deviation $\sigma = 0.1$ Hz (0.1 Hz corresponds with the measurements accuracy possible to obtain using the equipment at our disposal) and then trimmed again to 0.1 Hz.

A number of simulations involving MLP updating of FRAME4 model with four parameters were performed. The input vector was composed of model eigenfrequencies: $\mathbf{x} = \{f_1, f_2, \ldots, f_n\}$ where $n = 1, 2, \ldots, 7$ and it had N elements, the output vector in all cases consisted of parameters being updated: $\mathbf{y} = \{k_1, k_2, k_3, H\}$ (see Fig. 3c). The architecture of MLP applied was N-h-4, where $h = 2, 3, \ldots, 30$. Altogether 7×29 different combinations of input vector definition and the number of hidden neurones were tested. In each case MLP was trained on numerical patterns to update the FRAME4 model parameters and after training the parameters were identified using the measured eigenfrequencies as MLP inputs. In each case ARE^7 errors of the first seven eigenfrequencies were calculated (even for $N < 7$, when less than seven eigenfrequencies were involved during the updating procedure). The mean values of ARE^7 over different h and for constant n were computed, and are shown in Fig. 4.

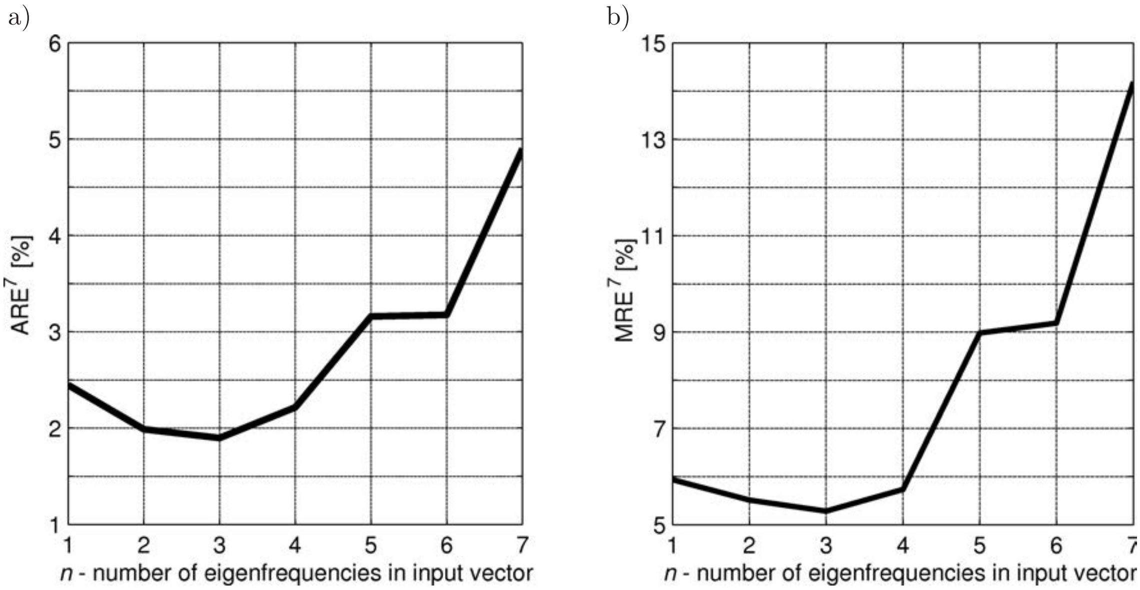

Fig. 4. Eigenfrequencies prediction error: a) ARE^7, b) MRE^7 versus the number of eigenfrequencies in the MLP input vector.

It is clearly visible that in order to minimize the error of prediction of the first seven eigenfrequencies the updating should be performed with only the first three eigenfrequencies involved. When the number of eigenfrequencies involved is higher, the model adjusts better to higher eigenfrequencies (mainly the fourth and the seventh) but the overall updating accuracy decreases.

Table 3 and Table 4 present the results of various types ANN updating parameters of FE model FRAME4. The networks MLP can be called *standard NNs*, vs Bayesian type NNs SBNN (Semi-Bayesian NN) and TBNN (True-BNN). All the investigated networks had their input vector composed of the first three eigenfrequencies: $\mathbf{x} = \{f_1, f_2, f_3\}$. The output vector was composed either of all the FE model free parameters (one network updated all the parameters simultaneously) or of only one parameter (four networks were required to update the FE model). In the first case the output vector definition was as follows: $y = \{k_1, k_2, k_3, H\}$, the network applied were MLPs, SBNN trained using Scaled Conjugate Gradients (SCG) algorithm.

Table 3. The updated FE model FRAME4 parameters and eigenfrequencies prediction accuracy.

	k_1	k_2	k_3	H	RMSE^7	MRE^7	ARE^7
	[Nm/rad]			[cm]	[Hz]	[%]	[%]
MLP	1004.3	1003.7	2497.1	4.3	8.36	4.94	1.70
SBNN, SCG	1919.1	2964.3	2606.1	2.9	7.94	4.93	1.75
single-output MLP	1582.4	3405.8	2668.3	2.6	7.80	6.04	1.99
TBNN	1923.6	2790.3	2846.5	2.9	8.15	4.98	1.93

Table 4. The eigenfrequencies obtained from updated two-storey frame FE models.

Measured value	f_1 [Hz] 32.25	f_2 [Hz] 95.50	f_3 [Hz] 110.30	f_4 [Hz] 126.30	f_5 [Hz] 265.30	f_6 [Hz] 294.80	f_7 [Hz] 422.00	RMSE^3 [Hz]	MRE^3 %	ARE^3 %
MLP	32.24 0.03%	95.49 0.01%	110.09 0.19%	121.14 4.09%	269.89 −1.73%	297.47 −0.91%	442.85 −4.94%	0.12	0.19	0.08
SBNN SCG	32.27 −0.06%	95.54 −0.04%	110.22 0.07%	120.07 4.93%	269.41 −1.55%	297.67 −0.97%	441.42 −4.60%	0.06	0.07	0.06
single output MLP	31.93 0.99%	95.55 −0.05%	110.26 0.04%	118.67 6.04%	269.03 −1.41%	297.78 −1.01%	440.56 −4.40%	0.19	0.99	0.36
TBNN	32.33 −0.25%	95.78 −0.29%	110.72 −0.38%	120.01 4.98%	269.65 −1.64%	298.59 −1.29%	441.80 −4.69%	0.30	0.38	0.31

In the latter case four output vectors were created: $y^1 = \{k_3\}$, $y^2 = \{H\}$, $y^3 = \{k_1\}$, $y^4 = \{k_2\}$. Four networks updated the FE model parameters in turn, in the following sequence: k_3, H, k_2 and k_1 (this sequence was determined by the dissipation of the parameters obtained by single four-output networks). In each step of this "cascade" approach a new set of patterns was created using the FE model with some parameters already updated in the previous steps. This approach involved TBNN (the TBNN software enforces the application of single output networks) and MLP (for comparison with the results obtained from TBNN).

The results presented in Table 3 and Table 4 show the differences between the measured and computed (by means of updated FE model FRAME4) eigenfrequencies. As stated earlier, the updating was performed taking into account only the first three eigenfrequencies, which is easy to notice in the final results: the relative errors calculated for f_1, f_2 and f_3 are below 1%, RMSE^3, MRE^3 and ARE^3 are close to zero (see Table 4). It is possible to minimize RMSE^n, MRE^n and ARE^n also for $n = 4, 5$ using the expanded input vector (e.g. $\mathbf{x} = \{f_1, f_2, f_3, f_4\}$), but this causes a significant increase of RMSE^7, MRE^7 and ARE^7 (see Fig. 4).

It is visible in Table 3 that the values of updating parameters and errors of approximation are comparable for the networks SBNN-SCG and TBNN. In cases of standard NNs application (networks MLP and MLP with single outputs) the values of the updating parameters are different from those corresponding to SBNN and TBNN.

4. Updating involving selected elements of eigenforms

In all the examples discussed above neural networks input vector consists of eigenfrequencies only. To make the procedure more general also selected elements of eigenforms should be incorporated into the input vector. Figure 5 presents preliminary results from such an approach in the identification of rotational stiffnesses in two-storey frame model FRAME4 (see Fig. 3c).

The input vector is defined as

$$\mathbf{x} = \left\{ f_1, \frac{v_1^1}{v_1^2}, \ f_2, \frac{v_2^1}{v_2^2} \right\}, \tag{14}$$

where v_j^i is i-th element of j-th eigenform. As one can notice only two first eigenfrequencies and two elements of two first eigenforms are taken into account. The output vector

$$\mathbf{y} = \{k_1, k_2, k_3\} \tag{15}$$

has been shrunk, in comparison to the previous example, by removing H which was assumed to equal 10 cm. The removal of parameter H from the input vector is connected with ambiguity of the results caused by the simultaneous identification of both H and k_1.

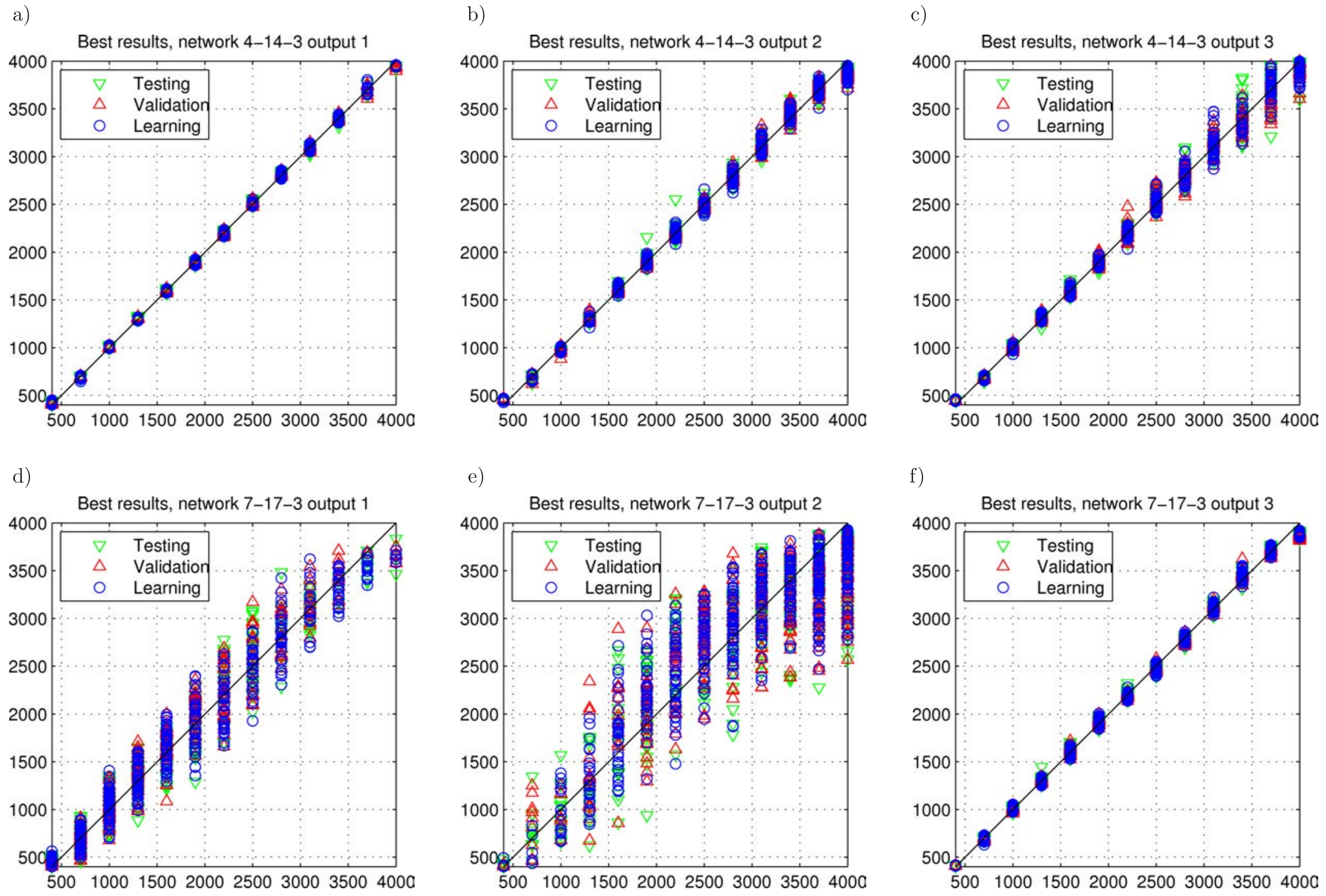

Fig. 5. Identification of k_1 (a), (d), k_2 (b), (e), k_3 (c), (f) either on the basis of eigenfrequencies and eigenvectors (a), (b), (c) or eigenfrequencies only (d), (e), (f).

Since the eigenvectors have not been measured yet the calculations were performed using numerical data only (with random noise $\pm(5-10)\%$).

Figure 5 shows a comparison of results obtained using the input vector involving elements of eigenvectors (Fig. 5a through c) or only seven first eigenfrequencies (5d through f). The nearer the points in Fig. 5 are to the line $x = y$ the better, as one can notice the results obtained from the approach involving eigenvectors is significantly better.

5. Conclusions

The paper presents the application of artificial neural networks, both standard MLP and probabilistic BNN, in the updating of FE models of engineering structures. Examples of updating of FE models of beam, portal frame and two storey frame anre presented. In each case the results are verified by experimental measurements.

Owing to constant physical interpretation of parameters being updated the obtained FE models can be applied in a variety of problems, including identification of material properties of structure members, location and identification of the size of defects and experiment planning, since the proposed procedure can point at the optimal location of the accelerometer and external excitation point.

The updating procedure involving neural networks is able to deal with inaccurate data and solve the identification problem even on the basis of a very limited amount of input information, one of the examples shows the updating of four FE model parameters on the basis of only three eigenfreuencies.

Further studies should concentrate on two fields:

- the application of eigenforms in the updating procedure: the presented results show that this can increase the accuracy of the presented method,
- careful selection of FE model parameters being updated, including the sensitivity analysis.

Acknowledgements

Financial support by the Polish Ministry of Science and High Education, grant No. N N506 432636 *Connection of neural networks and Bayesian inference in the identification analysis of structural dynamics and geomechanics problems* is gratefully acknowledged.

Apparatus/Equipment purchased in the project No. POPW.01.03.00-18-012/09 from the Structural Funds, The Development of Eastern Poland Operational Programme co-financed by the European Union, the European Regional Development Fund.

Numerical experiments were conducted with the use of MATLAB application, purchased during the realization of project No. UDA-RPPK.01.03.00-18-003/10-00 "Construction, expansion and modernization of the scientific-research base at Rzeszów University of Technology" is co-financed by the European Union from the European Regional Development Fund within Regional Operational Programme for the Podkarpackie Region for the years 2007-2013, I. Competitive and innovative economy, 1.3 Regional innovation system.

References

[1] C.M. Bishop. *Pattern Recognition and Machine Learning*, Heidelberg: Springer, 2006.

[2] H. Demuth, M. Beale. *Neural Network Toolbox User's Guide, Version 3.0*, The MathWorks Inc., Natick MA, USA, 1998.

[3] D.J. Ewins. *Modal Testing: Theory and Practice*, Letchworth: Research Studies Press, 1984.

[4] M.I. Friswell, J.R. Mottershead. *Finite Element Model Updating in Structural Dynamics*, Kluwer Academic Publishers, Dordrecht, 1996.

[5] S. Haykin. *Neural Networks: a Comprehensive Foundation*, Prentice Hall International, Inc., 2nd ed., 1999.

[6] M. Kłos, Z. Waszczyszyn. *Modal analysis and modified cascade neural networks in identification of geometrical parameters of circular arches*, Computers&Structures. **89**(7–8): 581–589, 2011.

[7] R.I. Levin, N.A.J. Lieven. *Dynamic Finite Element Model Updating Using Neural Networks*, Journal of Sound and Vibration, **210**(5): 593–607, 1998.

[8] R.I. Levin, N.A.J. Lieven. *Dynamic Finite Element Model Updating Using Simulated Annealing and Genetic Algorithms*, Mechanical systems and signals processing, **12**(1): 91–120, 1998.

[9] B. Miller. *Updating of Mathematical Models of Engineering Structures (in Polish)*, PhD dissertation, Rzeszów University of Technology, Rzeszów, 2002.

[10] J.E. Mottershead, M.I. Friswell. *Model Updating in Structural Dynamics: a Survey*, Journal of Sound and Vibration, **167**(2): 347–375, 1993.

[11] I.T. Nabney. *NETLAB Algorithms for Pattern Recognition*, London-Berlin-Heidelberg: Springer, 4th print, 2004.

[12] H.G. Natke. *Problems of Model Updating Procedures: a Perspective Resumptions*, Mechanical systems and signals processing, **12**(1): 65–74, 1998.

[13] J. Vanhatalo, A. Vehtari. *MCMC Methods for MLP-network and Gaussian Process and Stuff – A documentation for Matlab Toolbox MCMCstuff*, Helsinki University of Technology, Espoo, Finland, 2006.

[14] Z. Waszczyszyn, L. Ziemiański. *Neural Networks in Mechanics of Structures and Materials – New Results and Prospects of Applications*, Computers&Structures, **79**: 2261–2276, 2001.

[15] Z. Waszczyszyn, L. Ziemiański. Parameter Identification of Materials and Structures. In: Mróz Z. Stavroulakis GE, ed. Neural Networks in the Identification Analysis of Structural Mechanics Problems. CISM Lecture Notes. Wien, New York: Springer, 2004.

[16] Z. Waszczyszyn, M. Słoński. *Maximum of marginal likelihood criterion instead of cross-validation for designing*, In: Rutkowski L. et al., [Ed.] Artificial Intelligence and Soft Computing ICAISC2008, Berlin-Heidelberg-New York: Springer, 2008.

[17] Z. Waszczyszyn editor. Advances of Soft Computing in Engineering. CISM Courses and Lectures, Udine, vol. 512. Wien-New York: Springer, 2009.

[18] L. Ziemiański, B. Miller. *Dynamic Model Updating Using Neural Networks*. Computer Assisted Mechanics & Engineering Sciences, **4**: 68–86, 2000.

[19] L. Ziemiański, B. Miller, G. Piatkowski. *Application of neurocomputing in the parametric identification using dynamic responses of structural elements-selected problems*. Journal Of Theoretical And Applied Mechanics, **42**: 667–693, 2004.