Computer Assisted Mechanics and Engineering Sciences, **18**: 217–228, 2011.


# Identification problems of Recurrent Cascade Neural Network application in predicting an additional mass location

Grzegorz Piątkowski, Zenon Waszczyszyn
*Rzeszów University of Technology*
*Department of Structural Mechanics*
*Al. Powstańców Warszawy 12, 35-935 Rzeszów, Poland*
*e-mail: grzegorz.piatkowski@prz.edu.pl, zewasz@prz.edu.pl*

The paper is a development of research originated in [8]. The identification problem deals with searching the location of a small mass attached to a steel plate. The corresponding inverse problem is based on measurement of dynamic plate responses on a laboratory model of the plate, taking into account only the bending plate eigenfrequencies. In the inverse analysis the Recurrent Cascade Neural Network was applied, developed in [3]. Much attention is paid to recognition of identification possibilities of RCNN. The testing process is in fact an unsupervised learning, which can lead to unstable and inaccurate recurrence procedure. That is why the verification testing process was carried out adopting the barrier bound approach. These problems are discussed in the present paper.

**Keywords:** laboratory model of plate, plate eigenfrequencies, Recurrent Cascade Neural Network (RCNN), supervised and unsupervised learning, verification testing, barrier bound.

## 1. Introduction

The paper is a continuation of investigation reported in Piątkowski's Ph.D. dissertation [8]. The problem discussed involves identification of location of a small mass attached to a bending elastic plate, see Fig. 1a. The plate dynamic characteristics can be tuned owing to change of the mass placement. This effect can be explored in formulation of new non-destructive methods of construction damage detection, see [1].

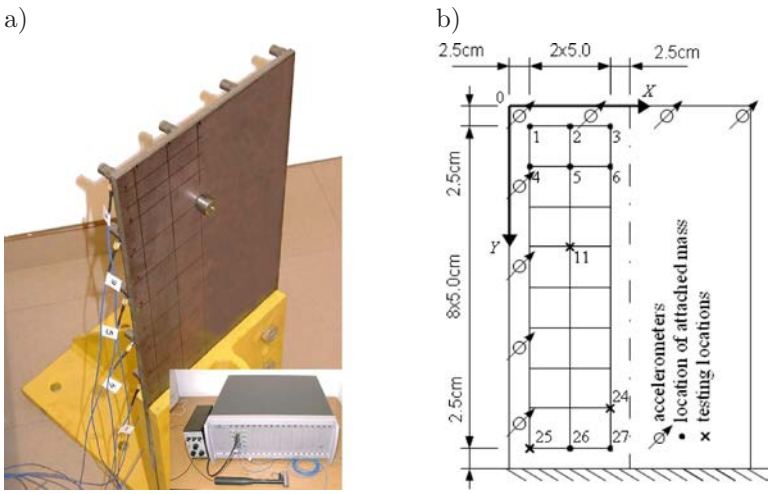


**Fig. 1.** a) Laboratory model for testing steel plate, b) scheme of tested plate.

On the basis of laboratory tests the plate responses were measured using an experimental laboratory set shown in Fig. 1a. The responses corresponded to the plate natural eigenfrequencies, where related to the bending forms of plate vibrations. They were measured for 27 locations of the mass attachments at the nodes of the square mesh corners, placed on a half of the symmetric plate, see Fig. 1b.

In order to identify an unknown location of the mass the Recurrent Cascade Neural Network (RCNN), described in [3], was applied. This network is a generalisation of the commonly used Cascade Neural Network, in which only the first step of recurrent process is carried out, cf. [11]. Due to continuation of this process, the recurrence of RCNN enabled us to improve significantly the accuracy of the network learning and change the character of the testing process, see [3].

Similarly as in [8, 9], the displaced coordinates $\{y_1, y_2\}^m$ for $m = 1, \ldots, 27$ measurement points of the additional mass attachment, were adopted as identified control parameters. The vector of measured plate eigenfrequencies $\{\boldsymbol{\omega}_j\}^m$ were the plate dynamic responses, corresponded to $j = 1, \ldots, J$ plate measured eigenfrequencies. Thus, the inverse problem $\{\boldsymbol{\omega}_j\}^m \rightarrow \mathbf{y}^m$ was analysed without introduction of any numerical model for the approximation of responses. Similarly as in [8], the amount of input data was increased adding the artificial white noise to the measured eigenfrequencies.

It was stated that the recurrent process, applied to the training patterns, runs similarly to that performed in the Jordan neural networks [7]. The learning process fully corresponds to the standard training of neural networks. A great difference is in the training process, which turns out to be related to unsupervised learning, recurrence iteration process. Contrary to results obtained in [3], where the computer simulation data were adopted, the problem investigated in the present paper is based on noisy data measured on the laboratory model. This caused difficulties with the stability and convergence of the recurrence testing process. These questions are discussed in the subsequent sections of the present paper.

## 2. Testing of plate laboratory model

A rectangular steel plate was clamped in the rigid stand, cf. [8, 9] and see Fig. 1a. The plate was made of an elastic steel strip of Young's modulus $E = 208$ GPa and thickness $h = 10 \pm 1$ mm. The plate weight was 10.76 kg and the additional mass of weight 109 g was attached at the corners of the quadratic $50 \times 50$ mm mesh at $M = 3 \times 9 = 27$ points (mesh corners) at the half of symmetric plate, see Fig. 1b. The experimental equipment was composed of eight PCB acceleration sensors, LMS SCADAD III analyser and LMS CADA-X software. Accelerograms were obtained for 27 locations of the small mass attachment with the measurement resolution of 0.5 Hz.

By means of Fourier Fast Transformation the first eight eigenfrequencies, corresponding to the bending forms of the plate vibrations, were selected for each additional mass location, see Fig. 2. Numbers 1-8 correspond to bending vibration form of the plate.

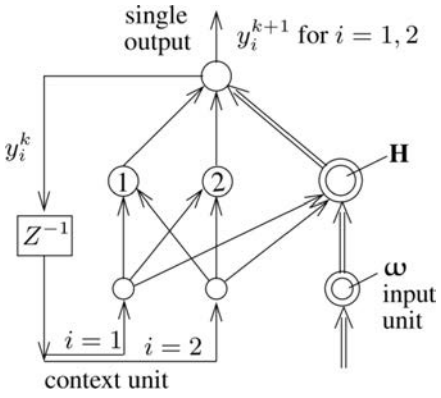


**Fig. 2.** Scheme of the Recurrent Cascade Neural Network (RCNN).

The measured plate responses are the eigenfrequency vectors $\boldsymbol{\omega}^m \in \mathcal{R}^J$ corresponding to the locations $\mathbf{X}^m \in \mathcal{R}^2$ of the attached small mass. This means that the following mapping is made for locations points $m = 1, \ldots, 27$, corresponding to the measurement process:

$$\mathbf{X}^m \in \mathcal{R}^2 \rightarrow \boldsymbol{\omega}^m \in \mathcal{R}^J, \tag{1}$$

where $J$ – number of eigenfrequency vectors taken into account at the mapping.

In Table 1 the bands of measured eigenfrequencies are given. From among them the first eigenfrequency $\omega_1$ was omitted and the next five eigenfrequencies were adopted, as these with the bands appropriate for the identification analysis. This means that in [8] the number of adopted eigenfrequencies was $J = 5$.

**Table 1.** Measured eigenfrequency bands of the bending forms of plate vibrations.

| Number of eigenfrequency | Interval values of eigenfrequencies | |
|---|---|---|
| | $\omega_{\min}$ | $\omega_{\max}$ |
| 1 | 35.5 | 37.0 |
| **2** | **116.0** | **125.0** |
| **3** | **236.0** | **256.0** |
| **4** | **430.5** | **445.0** |
| 5 | 588.5 | 636.0 |
| 6 | 645.5 | 673.0 |
| 7 | 829.5 | 862.0 |
| 8 | 930.5 | 984.0 |

On the basis of extended investigations both eigenfrequencies and corresponding vibration forms, the number of eigenfrequencies in the present paper was reduced to three that gave $J = 3$. The eigenfrequencies $\omega_2$, $\omega_3$ and $\omega_4$ were adopted in the identification analysis. This means that in the present paper the number of measured vectors of eigenfrequencies $P$ is $M = 27$ locations of the attached mass.

## 3. Recurrent cascade neural networks

### 3.1. Some general remarks on RCNN

As mentioned in the Introduction, RCNN is a kind of Jordan's neural network, cf. Fig. 3. It is composed of two parts. The input unit is the MLP (Multilayer Perceptron) network which has inputs corresponding to measured eigenfrequencies $\{\boldsymbol{\omega}\}^m$ . The other part of RCNN, called context unit, has the input/output self feedbacks $y_i^{(k)}/y_i^{(k+1)}$ for $i = 1, 2$.

The learning process proceeds under *supervised learning*. This means that the learning set of data $\mathcal{L}$ is composed of $M_{\mathrm{L}}$ known values of input/output pairs, randomly selected from the set of known data, which correspond to the measured set of patterns $\mathcal{P} = \{(\boldsymbol{\omega}, \mathbf{y})^m\}_{m=1}^M$. The patterns which were not selected compose the testing set $\mathcal{T}$ composed of $M_{\mathrm{T}} = M - M_{\mathrm{L}}$ patterns (random selection fulfils the following conditions: $\mathcal{P} = \mathcal{L} \cap \mathcal{T} = \varnothing$, $\mathcal{P} = \mathcal{L} \cup \mathcal{T}$). In case we use the randomly selected testing set $\mathcal{T}$, we perform the testing process for the known outputs, whose values are used in measuring of the testing process errors. Such an approach has a subjective character, see [4]. In the training process commonly used we compute predicted values of outputs without exploring their known outputs. Such outputs serve to verify computer results with those obtained on laboratory models or real structures.

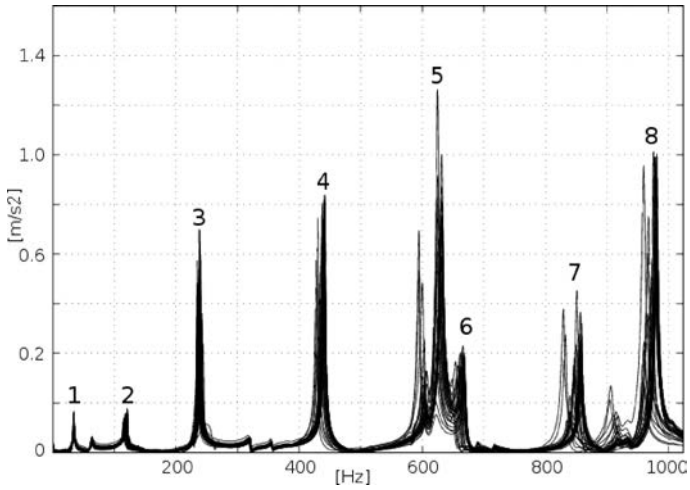


**Fig. 3.** Eigenfrequencies for the bending forms of a steel plate vibrations.

The training process discussed above does not correspond to recurrent computations carried out by RCNN because we have a set of discrete values of output vectors components $y_i^r(k)$. Because of the recurrence process, instead of standard testing we have to perform the *unsupervised learning* process.

Unsupervised learning is performed in cases we do not know the output values. Such a problem is related to the identification analysis. Looking at the learning process from the view point of the Hopfield type neural networks learning, the supervised learning corresponds to the stage of the network fundamental memory computation, see [2] when we compute attractors, which play the basic role in the next, recognition stage. Such a stage corresponds to the unsupervised leaning. From the mechanical point of view this process is related to searching of asymptotically stable points in the basin of attractors (searching for minimal values of network energy, cf. [2]).

The discussed analysis was numerically verified in [3], where the geometrical parameters of elastic circular, shallow arches were identified. The direct problem of computation of arch eigenfrequencies was carried out by the FE code ADINA. Then the inverse analysis was performed by the RCNN network. In [3] also artificial noise was introduced into the computed eigenfrequencies. The supervised learning gave satisfactory results after $K = 3$ recurrence cycles. The training process, supported on the un-supervised learning, gave convergent identified values of geometrical parameters for $K = 3$ recurrences, both for the perfect and noisy data. The results of the unsupervised learning were poorer than for the network recurrence learning, but the accuracy of the identification process could be estimated as quite good from the engineering point of view.

The analysis of the identification of the mass locations, concerning the problem reported in the present paper, turned out to be much more complex than identification of the geometrical parameters of arches. The number of needed recurrences reached $K \approx 10-20$ cycles of recurrence in the supervised learning process. This number of iteration steps gave stable but very inaccurate values of the unsupervised learning, performed for the measured noisy patterns. That is why an approximate approach was introduced. It can be called the *verification* testing process, related to introduction of a barrier bound as a penalty term. This corresponds to a subjective element of the testing process commonly used, induced by the recurrent character of RCNN. Such an approach is discussed below.

### 3.2. RCNN in the inverse analysis of additional mass location

In the problem considered we have a 2D space of control parameters. They correspond to the vector of computed displaced mass location vector $\mathbf{y}^m = \{y_1^m, y_2^m\}$. We consider an inverse regression problem, related to the parametric identification, which is related to the following mapping:

$$\boldsymbol{\omega}^m \in \mathcal{R}^3 \xrightarrow{\text{RCNN}} \mathbf{y}(\mathbf{x}^m, \mathbf{w}) \in \mathcal{R}^2, \tag{2}$$

where $\boldsymbol{\omega}^m_{(3\times1)}$ – vectors of eigenfrequencies measured at $m = 1, \ldots, M$ points of the additional mass locations, $\mathbf{y}_{(2\times1)}(\mathbf{x}^m, \mathbf{w})$ – vector of displaced mass locations at point $m$, $\mathbf{x}^m = \{X, Y\}^m$ – vector of coordinates of point $m$, $\mathbf{w} \in \mathcal{R}^W$ – vector of the neural network weights.

The RCNN is shown in compact form in Fig. 3. The extended form, developed in [3], is presented in Fig. 4. Figure 4a illustrates a neural network called a Multi-Layer Perceptron (MLP), cf. [2], of architecture MLP: 3-$H$-2 with a single hidden layer. This network corresponds to mapping (2).

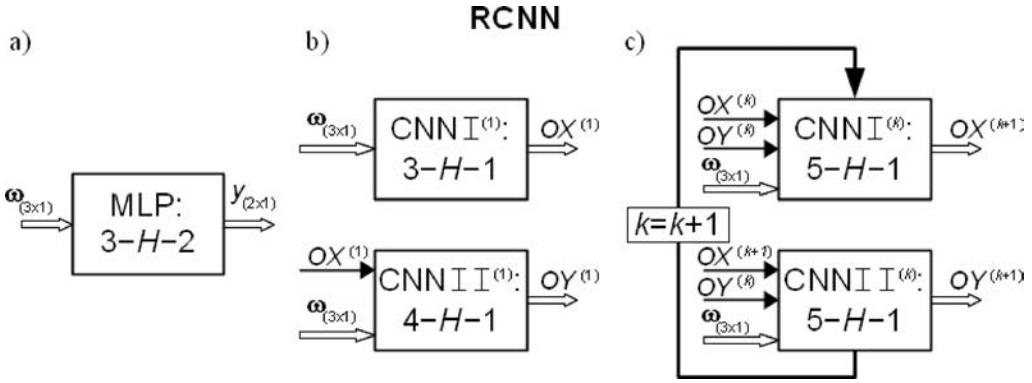


**Fig. 4.** Transition from a) Multi-Layer Perceptron (MLP) to: b) Cascade Neural Network (CNN) for $k = 1$, c) Recurrent Cascade Neural Network (RCNN) for $k > 1$.

Instead of MLP shown in Fig. 4a, a two-level RCNN is formulated for subsequent recursive cycles $k$. The standard CNN is applied for the first cycle $k = 1$, cf. [8], using two level networks with single outputs: I) CNN I$^{(1)}$: 3-$H$-1, II) CNN II$^{(1)}$: 4-$H$-1, see Fig. 4b. The networks CNN I$^{(k)}$ and CNN II$^{(k)}$ are of architecture 5-$H$-1 used for $k > 1$, see Fig. 4c.

### *3.2.1. Learning of RCNN*

The supervised learning of RCNN corresponds to recursive computation of the network weights. Following [8, 9] the data set was extended. Besides the measured eigenfrequency vector $\boldsymbol{\omega}^m_{(3\times1)}$, artificial noisy frequencies $\{\widetilde{\boldsymbol{\omega}}^m_n\}^N_{n=1} \in \mathcal{R}^3$ were added at each measurement point $m$. Thus, the noisy vector of eigenfrequencies has $1{+}N$ vectors of frequency, i.e. $\{\widetilde{\boldsymbol{\omega}}^m_n\}^{N+1}_{n=1} = \{\boldsymbol{\omega}^m + \widetilde{\boldsymbol{\omega}}^m_n\}^{N+1}_{n=1}$. The noisy vector input $\{\widetilde{\boldsymbol{\omega}}^m_n\}^{N+1}_{n=1}$ is related to the 2D output vector $\mathbf{y}^m = (y_1, y_2)^m$.

According to the algorithm developed in [3], the RCNN is formulated by introduction of the self feedback with respect to two coordinates $y_1$ and $y_2$, see Fig. 3. This means that for the recurrences $k > 1$ the networks are of architecture 5-$H$-1, see Fig. 4c. Figure 4 is valid for each point $m$ of the attached mass location. Due to self feedback the context input has the time delay character.

The learning process, carried out in the context unit of RCNN, corresponds to the prototype stage in the Hopfield NN, cf. [2]. This means that due to the training process, attractors are generated at each measurement point included into the network learning process. The accuracy of the RCNN learning was estimated by the Root Mean Square Error ($RMSEL_i$) for single outputs $y_i$:

$$RMSEL_i = \sqrt{\frac{1}{M_{\mathrm{L}} \times (N+1)} \sum_{m=1}^{M_{\mathrm{L}}} \sum_{p=1}^{(1+N)\times M_{\mathrm{L}}} (y_i^{p,m} - t_i^m)^2}, \tag{3}$$

where $i = 1, 2$ – the number of the displaced coordinates of a point of the mass location, cf. Fig. 5a, $M_L$ – the number of learning measurement points $m$, $N$ – the number of noisy patterns $p$ related to the mass location point $m$, $t_1^m = X^m$ and $t_2^m = Y^m$ – coordinates of the target output.

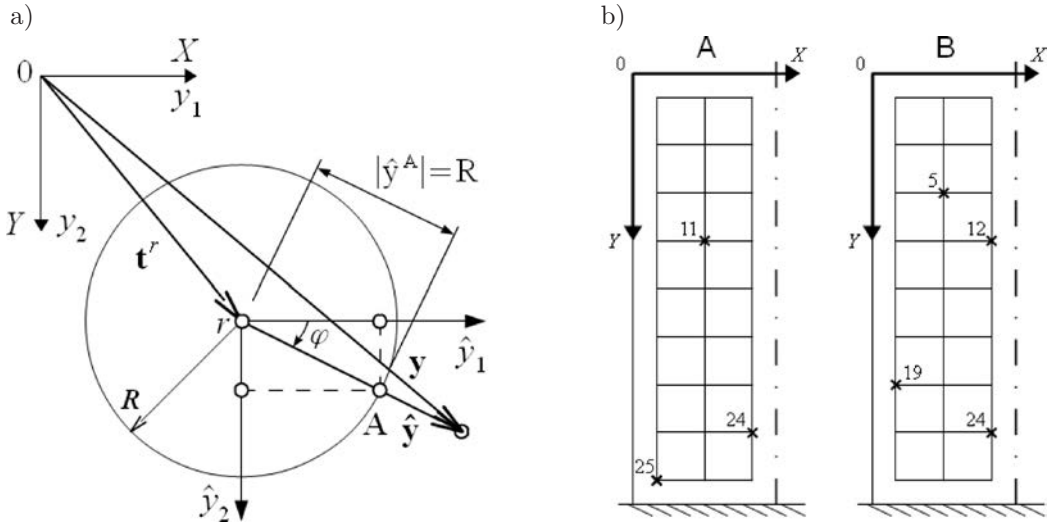


**Fig. 5.** a) Circular barrier bound for diminishing of the output vector $\mathbf{y}^{(k+1)}$ length, b) location of verification points $r$ for Example A and Example B.

The algorithm discussed above needs stopping criteria. They correspond to the number of cycles $K_{adm}$ , which enables fulfilling an adopted admissible error *AdmRMSEL*, see [3]. The other criterion is related to a fixed value of cycles, for instance $K = 20$, which was adopted as a stop criterion in the present paper.

### *3.2.2. Identification of mass location for given measured eigenfrequencies*

The approach commonly used has a subjective character, (see [4]), which is related to testing patterns $\mathcal{T}$, randomly selected from the set or patterns $\mathcal{P}$ with known values of outputs, cf. previous Subsec. 3.1. Then for $M_{\mathrm{T}}$ measurement points, the predictive outputs $\mathbf{y}^r$ should be computed by means of unsupervised learning for $r = 1, \ldots, M_{\mathrm{T}}$ patterns not included into the network training.

In case of RCNN the above mentioned approach can not be general and efficient, since it can lead either to bad predictions or the recursive iteration process can be divergent. Such numerical effects are caused by self feedbacks, which can cause the unsupervised learning very unstable for measured noisy patterns.

In the present paper we adopted the approach developed in [8]. The total number of $M = 27$ points, where the plate eigenfrequencies vectors $\boldsymbol{\omega}^m_{(3x1)}$ are measured, coincide with the locations of subsequently attached additional mass. The set $\mathcal{P}$ composed of $M$ patterns was randomly split into $M_{\mathrm{L}}$ and $M_{\mathrm{T}}$ learning and testing (identified) points, respectively. Thus, the testing error commonly used has the following form:

$$RMSET_{i,(k+1)} = \sqrt{\frac{1}{M_{\mathrm{T}}} \sum_{r=1}^{M_{\mathrm{T}}} (y^r_{i,(k+1)} - t^r_i)^2}. \tag{4}$$

A number of numerical experiments pointed out that unsupervised learning cannot be applied directly in the testing analysis corresponding to the adopted error measure (4). As previously mentioned, the noisy data obtained during measurements cause a great inaccuracy of computed vales of outputs. That is why the barrier boundary of circle shape was introduced as a penalty term to reduce the growth of the output vector length, see Fig. 5a. The centre of the barrier bound is placed at a fixed point $r$, see Fig. 5a.

The barrier bound of circle shape was adopted as a penalty term for reducing the growth of the output vector length $|\mathbf{y}^{(k+1)}|$, see Fig. 5a. The computation of the improved vector of the length

$|\widehat{\mathbf{y}}^{(k+1)}| = R$ can be made after both components of this vector are known, i.e. after the second step of each recurrent cycle $k$. The improvement condition can be written in the form:

$$\text{if} \quad |\widehat{\mathbf{y}}^{(k+1)}| \geq R \quad \text{then} \quad |\widehat{\mathbf{y}}^{(k+1)}| = R \quad \text{and} \quad (\widehat{y}_1 = R\cos\varphi, \;\; \widehat{y}_2 = R\sin\varphi),$$

where $R$ – radius of the circle barrier bound, $\widehat{\mathbf{y}}^{(k+1)}$ – improved vector related to the centre point $r$, see Fig. 5a.

In case the centre of the circle barrier bound coincides with the fixed location $r$ of the additional mass $m$ we obtain the approach adopted in [8]. The unsupervised learning, with such a barrier bound with the known location of its centre, can be called a *testing verification process* or *verification process*, for short. This process, used in [8], can verify the accuracy of the coincidence of the points mass location corresponding to the measured and verification processes.

## 4. Verification of neural inverse analysis

### 4.1. Formulation of patterns and design of RCNNs

The additional mass was placed subsequently at $M = 3 \times 9 = 27$ locations marked in Fig. 1b as measurements points $m = 1, \ldots, 27$. Around each of them twelve noisy points were randomly generated assuming artificial white noise with Gaussian distribution of zero mean and standard deviation $\sigma$. On the basis of the investigations carried out in [8], we assumed noisy values of frequencies $\{\widetilde{\omega}_j\}_{j=2}^{4}$ generated in [1] for $\sigma = 0.001$.

Thus, the input data set is completed of $P = 3 \times 9 = 27$ measured vectors of eigenfrequencies (patterns) and $P_N = 12 \times 27 = 324$ noisy patterns. Altogether $\widetilde{P} = (1 + 12) \times 27 = 351$ input patterns were adopted. They correspond to $2M$ output coordinates treated as components of the target vector $\mathbf{t}_{(54\times1)} = \{X^m, Y^m\}_{m=1}^{27}$.

For the design of RCNN at the first cycle $k = 1$ a single output CNN networks of architecture 4-$H$-1 and 5-$H$-1 were designed. In order to estimate the optimal value of hidden neurons $H_{opt}$ the data set $\widetilde{P}$ was randomly split with respect to the number of measurement points to have learning and testing data sets corresponding to the learning and vverification sets. Next, using the MLP with two outputs, the cross-validation method was applied in [8], which gave the estimation $H_{opt} = 5$. The same number of hidden neurons was estimated as the optimal number of hidden neurons for the network of architecture 5-$H$-1 for the next recurrent cycles in RCNN$^{(k)}$ for $k > 1$.

The networks mentioned above were composed of bipolar sigmoid hidden neurons and the linear output. The computations were carried out by means of the package called Neural Networks Toolbox for use with MATLAB [5], applying the Levenberg-Marquardt learning method.

The computation was carried out applying dimensionless variables: $\overline{\omega}_j = \omega_j/\omega_{\max} \in (0, 1.0]$, $\overline{X} = X/12.5 \in (0.0, 1.0]$ and $\overline{Y} = Y/42.5 \in (0.0, 1.0]$. In the discussion of results and in the figures and tables physical quantities were used.

### 4.2. Two random selections of verification mass locations

In order to compare the results obtained by RCNN, the verification measurement points were adopted as those randomly selected points analysed in [8].

- Example A: Three verification points $r = m = 11, 24, 25$ selected randomly in [8], see Fig. 5a.
- Example B: Four verification points $r = m = 5, 12, 19, 25$ selected in this paper and shown in Fig. 5b.

**Example A:**

The learning and identification (testing) error curves $RMSES_i$ for the learning and testing type of unsupervised learning and verification processes (marked by $S = L, U, V$) for the coordinates $i = y_1, y_2$ are shown for the barrier bound parameter $R = 2.5$ cm, see Fig. 6.

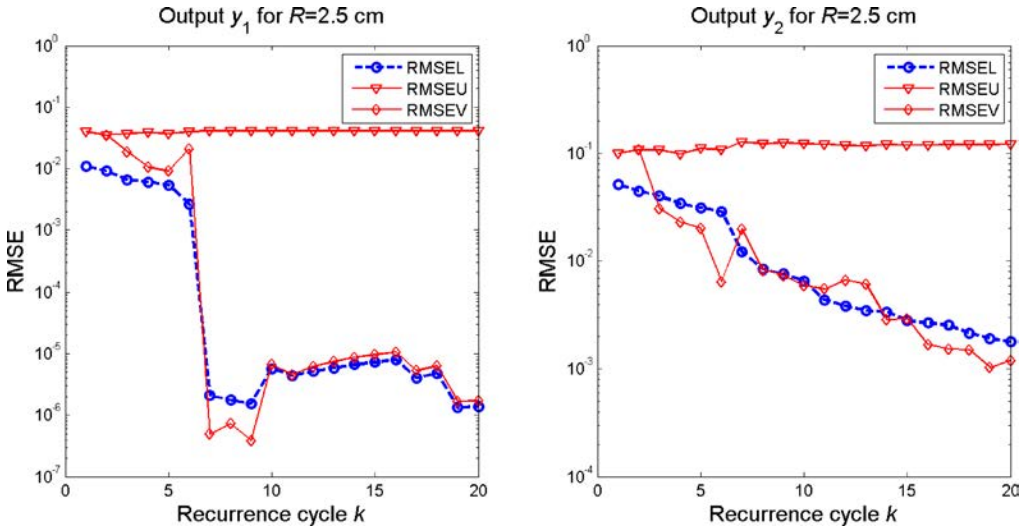


**Fig. 6.** Error curves $RMSES_i(k)$ for Example A, where: $i = y_1, y_2$, and for the supervised learning $S = L$, unsupervised learning $S = U$ and for verification $S = V$ processes, i.e. $RMSEL_i(k)$, $RMSEU_i(k)$ and $RMSEV_i(k)$, respectrively.

The learning process is characterised by the monotonically decreasing learning error curves, clearly visible at the curve $RMSEL_i(k)$, for $i = y_2$, see Fig. 6b. In case of the coordinate $y_1$ the learning process is monotonically decreasing up to the cycle $k \approx 6$. Then errors are so small that the numerical instability occurs and only a decreasing trend can be deduced.

The unsupervised learning error curves $RMSEU_i(k)$, computed for points $r$, are nearly independent of the recurrence number $k$. They are strongly insensitive to the recurrence process. The validation error curves $RMSEV_i(k)$ have discontinuous gradients caused by the interference of the penalty barrier bound. The shape of these curves is similar to the learning curves. Other effects are caused by a small number of the testing patterns (three patterns in Example‘ A). Such a set is statistically non-representative so the error of the verification curve can only point out a decreasing trend.

The convergence of verification iterations is shown in Fig. 7, where the computed points of mass locations are presented vs. target values $t^m$, depicted for selected recurrence cycles $k = 1, 5, 10, 20$. It is visible a quick convergence of the recurrence process in case of the coordinate $y_1$. An acceptable value of $K \approx 10$ cycles is twice smaller than the number of recurrence cycles for the identification of $y_2$ This corresponds to larger movements of displaced points in the direction parallel to the axis $y_2$ then to $y_1$. The values of verification errors are listed in Table 2 for different values of the barrier bound parameter $R$ and $k = 20$.

Figure 8A illustrates the movements of verified points $r = 11, 24, 25$ in Example A. The initial movable position of the ends of vectors $y^{r,k+1)}$, see Fig. 5a, are marked in Figs. 8 by ▼ and the final position, related to $k = 20$ corresponds to ♦. These positions of the movement of mass location $s$ are joined by the broken line.

It is visible that for the points located far off the clamped edge of the tested plate, the bounds of movement are smaller than for the locations which are close to the lower, clamped boundary of the plate. For the higher value of barrier bound parameter $R$ the bands of location movements are wider. Nevertheless, for $R = 5.0$ cm (equal the site of the measurement point mesh) the validated positions of mass locations are quite accurate. In case of $R = 7.5$ cm the verified location of the

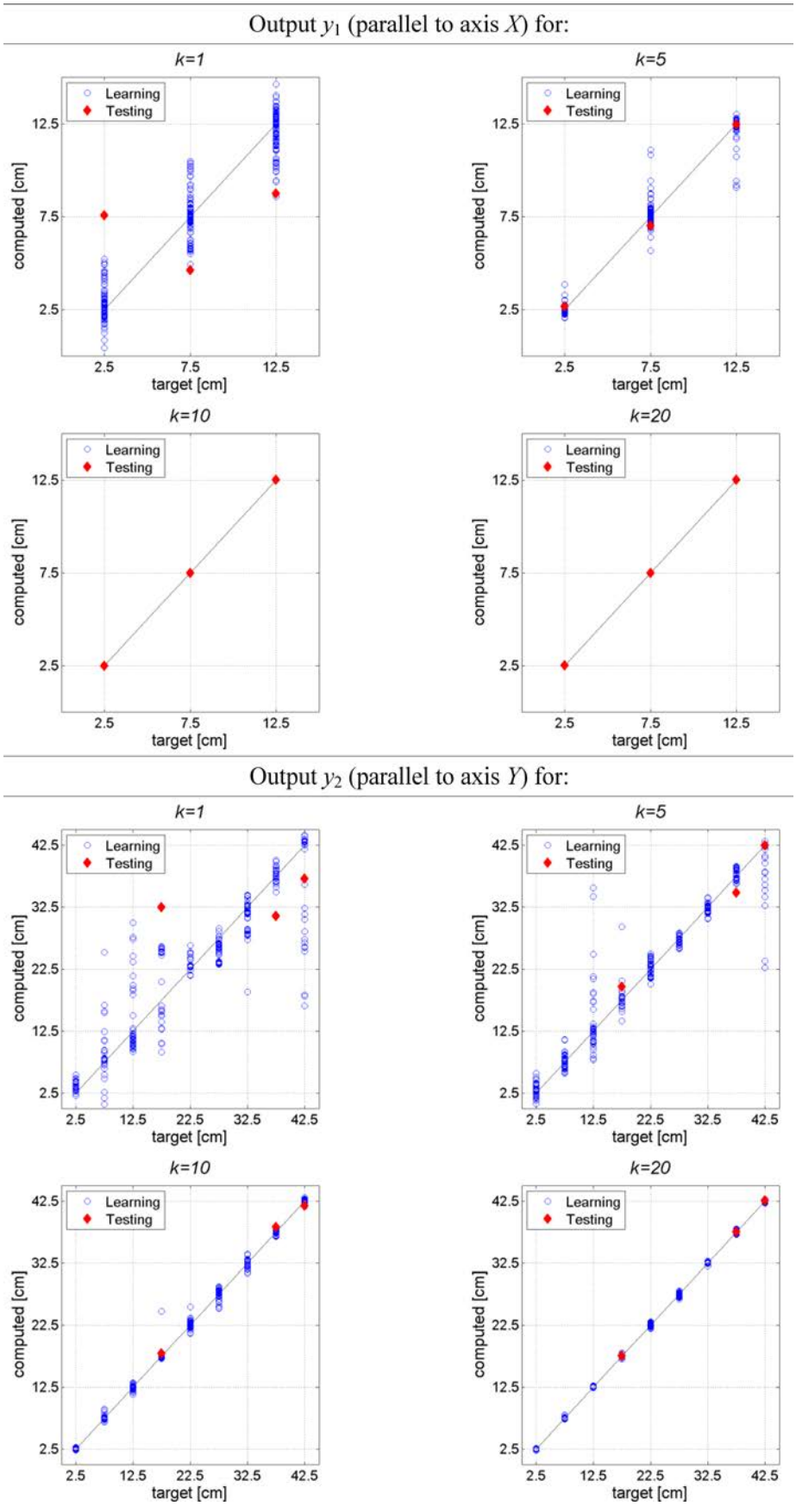


**Fig. 7.** Relations of computed outputs $y_i(\mathbf{x}^m; k)$ vs. target outputs $t_i(\mathbf{x}^m; k)$ for Example A.

**Table 2.** Displacements of points $m$ for barrier parameters $R$ for recurrence cycle $k = -20$.

| Example | $m$ | $(OX^m, OY^m)$ [cm] | $(\Delta OX^m, \Delta OY^m)$ for: | | | $(\Delta OX^m, \Delta OY^m)$ |
|---|---|---|---|---|---|---|
| | | | $R = 2.5$ cm | $R = 5.0$ cm | $R = 7.5$ cm | According to [1] |
| **A** | 11 | (7.50, 17.5) | (0.00, 0.02) | (0.00, 0.05) | (0.01, 0.16) | (0.00, 0.85) |
| | 24 | (12.5, 37.50) | (0.00, 0.06) | (0.00, 0.14) | (0.01, −0.44) | (−0.09, 1.13) |
| | 25 | (2.50, 42.50) | (0.0, 0.08) | (0.00, −0.74) | (5.00, 1.35) | (5.00, 3.21) |
| **B** | 5 | (7.50, 7.5) | (0.00, 0.54) | (0.00, −0.39) | (0.03, 0.52) | – |
| | 12 | (12.5, 17.50) | (0.00, −0,24) | (0.00, −0.49) | (0.00 0.65) | – |
| | 19 | (2.50, 32.50) | (0.0, 0.21) | (0.00, −0.13) | (0.00, −0.07) | – |
| | 24 | (12.5, 37.50) | (0.0, 0.12) | (0.00, 0.75) | (5.00, −0.59) | |

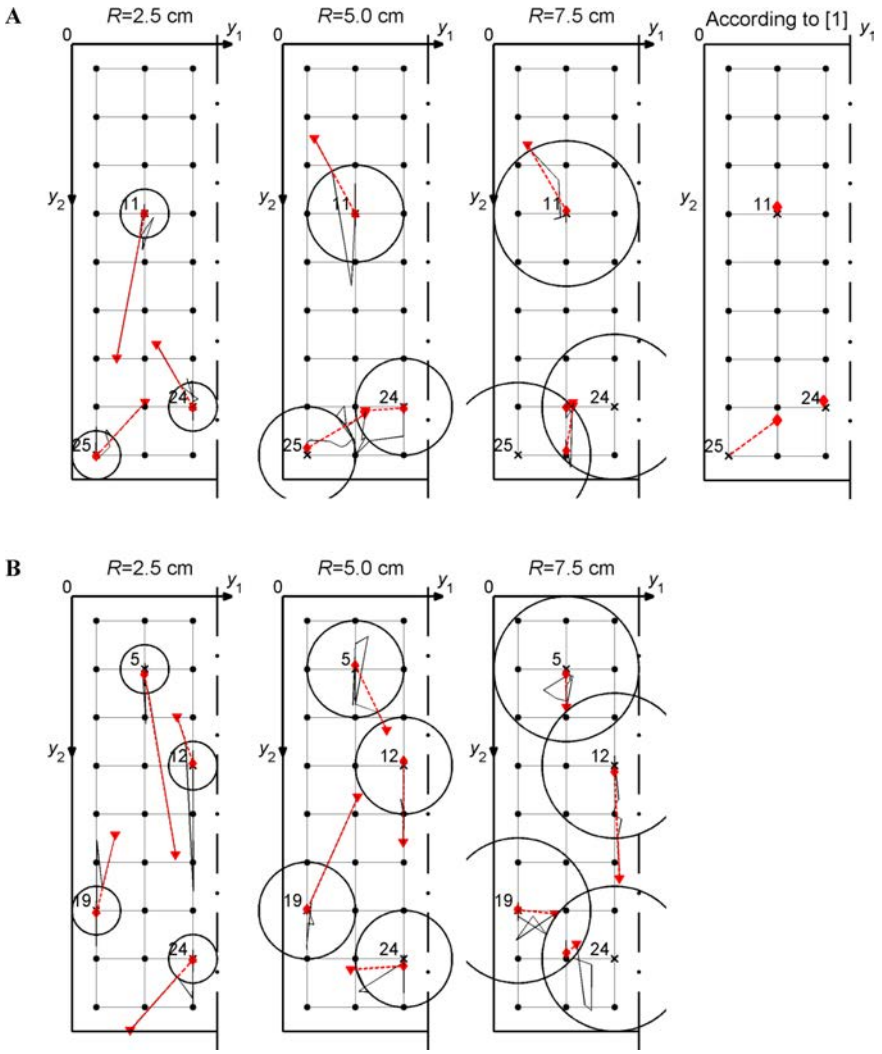


**Fig. 8.** Movements of location points ▼ and ♦ for Example A and B.

point 25 is false. After $k = 20$ recurrent steps this point moves towards the point 26, which is closer to the measurement point 26 than to the validation point 25.

In Fig. 8A the verification performed for the same points as in [8] is shown. The identification obtained for $k = 1$ (standard CNN was used in [8]) is much poorer than that obtained by RCNN for $R = 5.0$ cm and $k = 20$. This especially concerns the point 25. The values of identified positions of three measurement mass locations $r = 11, 24, 25$ are given in Table 2.

**Example B:**

The verification of mass locations at four points $r = 5, 12, 19, 24$ was computed similarly as in Example A. The only results are shown in Fig. 8B and Table 2. They confirmed the conclusions expressed above, concerning the influence of the clamped edge boundary conditions of the tested plate and on the accuracy of verification. The same effects are related to the adopted values of barrier bound parameter $R$. The movement of points shown in Fig. 8A similar to the movements shown in Fig. 8B. The displacements of points related to the verified points $r = 5, 12, 19, 24$, computed for $k = 20$, are given in Table 2.

## 5. Application of sbnn

Having single outputs in RCNN the Semi-Bayesian Neural Network (SBNN) was also used. It was concluded in [10] that this kind of neural network can improve the accuracy of neural approximation. Application of SBNN in the considered problem of identification of location of a small mass attachment gave results comparable to those obtained by RCNN for the training process. In case of validation process basing on the SBNN application turned out to be divergent. This question needs further investigation.

## 6. General remarks and conclusions

1. The paper is a continuation of research originated in [8]. The investigated problem was based only on the measurements of a steel plate responses, which were eigenfrequencies corresponding to locations of a small mass attached subsequently to the plate points. In such a way the empirical data set EMP was composed of mass location points, distributed regularly over the plate.

2. The Recurrent Cascade Neural Network (RCNN), developed in [3], was applied for the analysis of a reverse problem. The problem was related to the validation of the location of the attached mass knowing only measured eigenfrequencies. Thus, a hybrid computational system EMP&RCNN was formulated. The system is of a low degree of fusion of two components EMP and RCNN, cf. [6]. Contrary to the commonly used approaches, no computational method was used for the approximation of measured responses.

3. The recurrence of RCNN enables significant improvement of the accuracy of supervised learning. The testing process should correspond to unsupervised learning. This learning process proved to be stable and quite accurate in case of computationally simulated patterns, see [3]. In the case of measured noisy patterns, used in the present paper, the testing process is not convergent with the identified patterns. Introduction of the barrier bound with immovable centre made the recurrence process convergent with the known locations of the mass locations. These processes, called verification process, do not well correspond to objective identification of tested patterns.

4. The results obtained for the verification process supported the introduction of the barrier bound, and can be treated as a step towards deeper knowledge of the identification features of RCNN. The current investigations are directed a the development of a new unsupervised learning process corresponding to the RCNN network structure.

5. Application of Semi-Bayesian Neural Network gave results of the supervised learning comparable to those obtained by RCNN. This was not the case of verification process which turned out to be divergent if SBNN was applied. This question needs further investigation.

## Acknowledgments

The authors would like to acknowledge financial support from the Polish Ministry of Science and Higher Education, Grant No N N506 4326 36, "Joining of artificial neural networks and Bayesian reasoning for the analysis of identification problems in structural dynamics and geotechnics".

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] K. Dems, Z. Mroz. Identification of damage in beam and plate structures using parameter-dependent frequency changes. *Eng. Comp.*, **18**: 96–120, 2001.

[2] S. Haykin. *Neural Networks – A Comprehensive Foundation*, 2nd ed. Prentice-Hall, Upper Saddle River, 1999.

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

[4] T. Masters. *Practical Neural Network Recipes in C++*. Academic Press, 1993.

[5] *Neural Network Toolbox for Use with MATLAB*, User's Guide, Version 4. The MathWorks, Inc., Natick, MA, 2000.

[6] E. Pabisek. *Hybrid Systems Integrating FEM and ANN for the Analysis of Selected Problems of Structural and Material Mechanics*, D.Sc. Dissertation, Monograph 369, Series of Structural Engineering. Cracow University of Technology. Cracow, 2008.

[7] D.T. Pham, X. Liu. Neural Networks for Identification and Control, 2nd ed. Prentice–Hall, 2000.

[8] G. Piątkowski. *Damage Detection in Structural Elements Applying Artificial Neural Networks* (in Polish). Ph.D. Dissertation, Rzeszów University of Technology, 2003.

[9] G. Piątkowski, L. Ziemiański. The solution of an inverse problem in plates by means of artificial neural networks. In: L. Rutkowski et al. [Eds.]. *Artificial Intelligence and Soft Computing – ICAISC 2004*, LNAI 3070, 1087–1092. Springer, 2004.

[10] Z. Waszczyszyn, M. Słoński. Selected problems of artificial neural networks development. In: Z. Waszczyszyn [Ed.], *Advances of Soft Computing in Engineering*, CISM Courses and Lectures, vol. 512, Springer, Wien New York, 2010.

[11] Z. Waszczyszyn, L. Ziemiański. Neural networks in the identification analysis of structural mechanics problems. In: Z. Mróz, G. Stavroulakis [Eds.]. *Parameter Identification of Materials and Structures.* CISM Lecture Notes, vol. 469, 265–340. Springer, Wien New York, 2005.