Structural Engineering and Mechanics, Vol. 18, No. 3 (2004) 303-314
DOI: http://dx.doi.org/10.12989/sem.2004.18.3.303

Flexural and axial vibration analysis of beams with different support conditions using artificial neural networks

Ömer Civalek†

Akdeniz University, Engineering Faculty, Civil Engineering Department, Division of Mechanics, Antalya, Turkey

(Received February 13, 2003, Accepted February 26, 2004)

Abstract. An artificial neural network (ANN) application is presented for flexural and axial vibration analysis of elastic beams with various support conditions. The first three natural frequencies of beams are obtained using multi layer neural network based back-propagation error learning algorithm. The natural frequencies of beams are calculated for six different boundary conditions via direct solution of governing differential equations of beams and Rayleigh's approximate method. The training of the network has been made using these data only flexural vibration case. The trained neural network, however, had been tested for cantilever beam (C-F), and both end free (F-F) in case the axial vibration, and clamped-clamped (C-C), and Guided-Pinned (G-P) support condition in case the flexural vibrations which were not included in the training set. The results found by using artificial neural network are sufficiently close to the theoretical results. It has been demonstrated that the artificial neural network approach applied in this study is highly successful for the purposes of free vibration analysis of elastic beams.

Key words: artificial neural networks; natural frequencies; axial and flexural vibration; elastic beams.

1. Introduction

During the past ten years there has been a growing interest in algorithms, which rely on analogies to natural processes. The emergence of massively parallel computers made these algorithms of practical interest. These well-known algorithms and techniques in this class include artificial neural networks, genetic algorithms, fuzzy logic, evolution algorithms and simulated annealing. Although all these techniques have been adapted to the structural analysis, design and optimization problems, artificial neural networks (ANN) applications are much more general and found its application in every field of engineering problems. A neural network is a technique that seeks to build an intelligent program using models that simulate the working network of the neurons in the human brain. In contrast to conventional computers, which are programmed to perform specific tasks, most neural networks must be trained. They learn new associations, patterns, and functional dependencies. Artificial neural networks have been used widely in the field of structural analysis in recent years. Pioneering studies by Adeli and Yeh (1989), Adeli and Park (1995) Adeli and Hung (1995), and

† Assistant Professor

Ghaboussi *et al*. (1991) showed that neural networks can be applied successfully to the analysis of structural engineering problems. The first structural engineering application of ANNs is known the study of Adeli and Yeh (1989). Several other researchers have applied neural networks (Park and Adeli 1997), mostly back-propagation algorithms (Hani and Ghaboussi 1998, Wu *et al*. 1992, Chen *et al*. 1995). Authors have been suggested a fuzzy based neural network, hybrid artificial intelligent techniques and applied the analysis of plates and shells (Civalek 1998a, 1998b, 1998c, 1999a, 1999b, Ülker and Civalek 2001). Yun *et al*. (2001, 1998a) recently developed an ANN approach for joint damage assessment of structures. Furthermore, Yun and his co-workers (Yun *et al*. 1997, 2000) investigated the substructural identification using neural networks. Recently, Ghaboussi and Lin (1998) developed a new methodology for generating artificial earthquake accelerograms using neural networks. Stochastic neural networks approach for generating multiple spectrum compatible accelerograms is also given by Lin and Ghaboussi (2001). In the present study, natural frequencies of elastic beams are obtained using the ANN that adopts the back-propagation algorithm to train the network.

2. Artificial Neural Networks (ANN)

The general structure of a generic neuron is shown Fig. 1. A typical cell has three major regions: the cell body, which is also called the soma, the axon and the dendrites. The axon of a typical neuron makes a few thousand synapses with other neurons. Dendrites receive information from neurons through axons. The axon-dendrite contact organ is called a synapse. The synapse is where the neuron introduces its signal to the next neuron. McCullough and Pitts (1943) proposed a simple model of a biological neuron as a binary threshold unit as shown below (Fig. 2). Specially, the

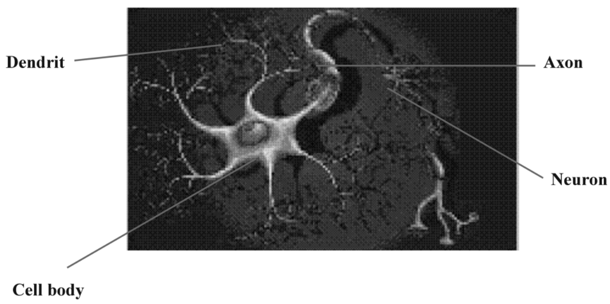


Fig. 1 A typical biological neuron and its elements

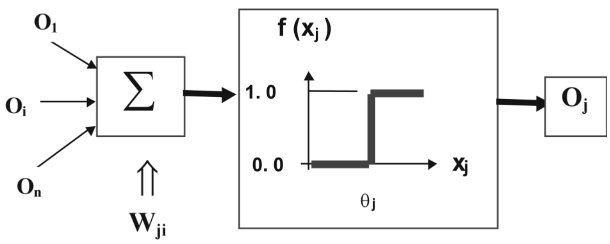


Fig. 2 McCullough - Pitts model of a biological neuron

model neuron computes a weighted sum of its inputs from other units given by,

$$Net_j = \sum_{i=1}^{N} O_i W_{ji} \tag{1}$$

where Net_j is the total input of the jth neuron, O_i is the input for this node, W_{ji} is the weighting coefficient, N is the number of the inputs for this node.

3. Back-propagation neural networks

Among the various neural network paradigms available, back-propagation networks are by far the most widely utilized for its relatively simple mathematical proofs and efficient generalization capabilities. The back-propagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a multi-layer feed-forward perceptron and the desired output. Although back-propagation is a trained algorithm, it is generally used and well known as a type of ANNs. For the most commonly used activation functions, the value of the derivative can be expressed in terms of the value of the function. One of the most typical functions is the binary sigmoid function, which has range of (0, 1) and is defined as

$$F(x_j) = \frac{1}{1 + \exp(-x_j)} \tag{2}$$

where x_j is the weighted summation of the total input. Back-propagation networks are typically trained using the generalized delta rule, application of which involves the calculation of the network output, a comparison of this output with desired output, the calculation of an error, and a backward propagation of this error in order to correct future outputs. In this process, each neuron updates the weights of its input connections in such a way that the error associated with its own output activation is decreased. Like a perceptron, a back-propagation network typically starts out with a random set of weights.

The Back-propagation algorithm is summarized below:

Weight Initialization: Set all weights and node thresholds to small random numbers. Note that the node threshold is the negative of the weight form the bias unit.

Calculation of Activation: 1-The activation level of an input unit is determined by the instance presented to the network.

2-The activation level O_j of a hidden and output layer is determined by

$$O_j = F(\sum W_{ji} O_i - \theta_j) \tag{3}$$

where W_{ji} is the weight from an input O_i, θ_j is the node threshold, F is a sigmoid function that was defined above in Eq. (2).

Weight Training: 1- Start at the output units and work backward to the hidden layers recursively. Adjust weights by

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji} \tag{4}$$

where $W_{ji}(t)$ is the weight from unit i to unit j at time t (or t th iteration) and ΔW_{ji} is the weight adjustment.

2-The weight change is computed by

$$\Delta W_{ji} = \alpha \delta_j O_i \tag{5}$$

where α is a trial-independent learning rate ($0 < \alpha < 1$) and δ_j is the error gradient at unit j. Convergence is sometimes faster by adding a momentum term (Rumelhart *et al*. 1986);

$$W_{ji}(t+1) = W_{ji}(t) + \alpha \delta_j O_i + \beta [W_{ji}(t) - W_{ji}(t-1)] \tag{6}$$

where β is known momentum term.

3-The error gradient is given by: For the output units;

$$\delta_j = O_j(1 - O_j)(T_j - O_j) \tag{7}$$

where T_j is desired (target) output activation and O_j is the actual output activation at output unit j. For the hidden units;

$$\delta_j = O_j(1 - O_j) \sum_k \delta_k W_{kj} \tag{8}$$

where δ_k is the error gradient at unit k to which a connection points from hidden units j. Repeat iterations until convergence in terms of the selected error criterion. The well-known error criterion (Adeli and Hung 1995, Fausett 1994, Zurada 1992) is defined as average error and given by;

$$E_a = \frac{1}{2P} \sum_p \sum_j (T_j - O_j)^2 \tag{9}$$

where P is the total number of instances.

4. Dynamic analysis of beams

Significant advancement has been made in the past two decades in the development and application of the theory to dynamic analysis of structures. This has been brought about for primarily two reasons: a better understanding of the physical principles involved and the ever-increasing advances in computer technology providing faster executing times, grater accuracy and large storage capacity at a smaller cost. Analysis and design of engineering structures such as; bridge, dam, tall building, nuclear power plant constructed subjected to dynamic loads involve consideration of time-dependent inertial forces. The resistance to displacement exhibited by a structure may include forces that are functions of the displacement and the velocity. As a consequence, the governing equations of motion of the dynamic system are generally linear and nonlinear partial differential equations that are extremely difficult to solve in mathematical terms (Paz 1997). The dynamic analysis of beams for free or forced vibration has been analyzed by various numerical techniques, such as finite difference, finite element, and Rayleigh-Ritz, by this time. General introduction and detailed solution of the beam vibration problems can be found in the related references (Meirovitch 1986, Thomsan and Dahleh 1988, Paz 1997, Craig 1981).

4.1 Lateral (flexural) vibration

Consider the lateral vibration of a thin uniform beam. The governing differential equation of motion for free vibrations (Paz 1997):

$$EI\frac{\partial^4 y}{\partial x^4}(x, t) + \rho A\frac{\partial^2 y}{\partial t^2}(x, t) = 0 \tag{10}$$

where EI is the flexural rigidity of beams is a constant, ρ is the material density, and A is the cross-sectional area of the beam. One method of solving this equation is by separation of variables. It can be assumed that the solution of the governing equation of beams may be expressed as the product of a function of position $\Psi(x)$ and a function of time $\beta(t)$ that is,

$$y(x, t) = \psi(x)\beta(t) \tag{11}$$

where $\psi(x)$ is the function of x and $\beta(t)$ is a function of time t. By substituting Eq. (11) into Eq. (10) and separating variables,

$$\frac{EI}{m}\left[\frac{\psi^{IV}(x)}{\psi(x)}\right] = -\frac{\ddot{\beta}(t)}{\beta(t)} \tag{12}$$

where dot denotes time derivatives and roman denote x derivatives, and $m = \rho A$. The two sides of the differential equation above will be equal to each other only if both are equal to the same constant. Thus,

$$\psi^{IV}(x) - \frac{m}{EI}\Omega^2\psi(x) = 0 \tag{13a}$$

and

$$\ddot{\beta}(t) + \Omega^2\beta(t) = 0 \tag{13b}$$

At this point, we assume a solution of the Eq. (13b) of the form

$$\beta(t) = \sin\omega t \tag{14}$$

$$\Omega^2 = \omega^2 \tag{15}$$

Finally, by using the result above, Eq. (13a) is written as

$$\frac{d^4\psi(x)}{dx^4} - \lambda^4\psi(x) = 0 \tag{16}$$

where

$$\lambda^4 = \frac{m\omega^2}{EI}$$

The solution of Eq. (16) is

$$\psi(x) = A\cosh(\lambda x) + B\sinh(\lambda x) + C\cos(\lambda x) + D\sin(\lambda x) \tag{17}$$

By substituting the Eq. (14) and Eq. (17) into Eq. (11), the general solution is given by (Clark 1972):

$$y(x, t) = [A\cosh(\lambda x) + B\sinh(\lambda x) + C\cos(\lambda x) + D\sin(\lambda x)]\sin(\omega t) \tag{18}$$

where A, B, C, and D are coefficients which depend on the boundary conditions of the beam. Another approximate method that is widely used in the vibration analysis of beam is the Rayleigh's method. In this method, a deflection function is chosen which satisfy the beam boundary conditions. For the $\varphi_k(x)$ displacement function, the frequencies of the beam is defined as

$$\omega_k = \left[\frac{\int_0^L EI(x)[\varphi_k''(x)]^2 dx}{\int_0^L m(x)[\varphi_k(x)]^2 dx}\right]^{1/2} \tag{19}$$

where $\varphi_k(x)$ is the kth mode shape function.

4.2 Axial vibration

In the case of axial vibration, the governing equation of motion is given by (Craig 1981)

$$EA\frac{\partial^2 y}{\partial x^2}(x, t) - \rho A\frac{\partial^2 y}{\partial t^2}(x, t) = 0 \tag{20}$$

As similar to the lateral vibration, the solution of the above Eq. (13) is given by

$$y(x, t) = \psi(x)\beta(t) = [A_1\cos(\lambda x) + A_2\sin(\lambda x)]\cos(\omega t) \tag{21}$$

where where A_1 and A_2 are coefficients which depend on the boundary conditions of the beam and $\lambda = \omega^2\rho/E$. In the present study, the training set had been obtained by using the Rayleigh's and analytical method.

5. Training and architecture of the developed network

The architecture of the neural network used in this study is shown in Fig. 3. It is widely known that one hidden layer is general sufficient for back-propagation networks. A single hidden layer

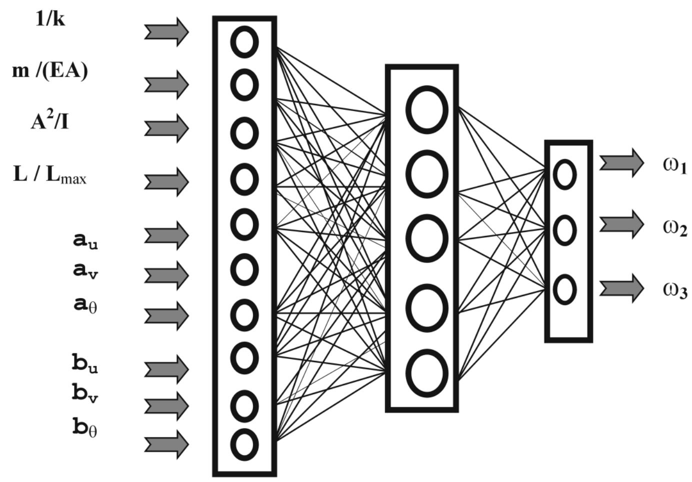


Fig. 3 Architecture of the developed multi-layer network

network is also easier to train. The network is feed-forward, multi-layer neural network which have ten input nodes in the input layer, and three output nodes in the output layer. One hidden layer with five nodes, however, had been used in the developed network. In the input layer; six neurons are used for support conditions that was known as the axial, vertical, and rotational displacement capability of each support. These values may be either 1 or 0, and given in Table 5 as similar the matrix-displacement method, and four neurons to defined the material and geometrical properties of the beam. For the unknown frequencies, three nodes are used in the output layer. Totally 13 different parameters are defined in the input layer as indicated; k, m, I, E, L, L_{max}, A, a_u, a_v, a_θ, and b_u, b_v, b_θ. These are known respectively; the stiffness of the beam, mass per unit length, moment of inertia, modulus of elasticity, length, maximum length in the training set, cross- sectional area of the beam and support conditions. The output variables were the first three natural frequencies of the beam as indicated ω_1, ω_2, ω_3. The stiffness coefficients can be easily found in the literature (Thomsan and Dahleh 1998). Some of these are shown in Table 1. Support conditions are defined as form the binary code and its given in Table 5.

The selection of number of nodes in the hidden layer is an important factor in the architecture of the network. The number of neurons of the hidden layer is recommended to be at least greater than square root of the sum of the number of neurons in the input and output layer (Eberhart and Dobbins 1990), or usually selected as the mean value of the number of the input and output nodes plus the input nodes. More complicated networks use dynamic node growing in the hidden layer. We used the first criterion in this study. In this paper, the network with one hidden layer having five nodes is good enough to model the title problem. The training set contains input and output (target) vectors. The input and output data had been normalized so that the maximum value is 1. Normalization of the data can often be as simple as either dividing the values by the maximum value or by subtracting the minimum value and then dividing the values by the range, which is the maximum value minus the minimum value. In this study, we use the first of them for normalization procedure.

Table 1 Some of stiffness coefficients for the network training

Beam support and vibration type	Stiffness
Axial vibration	$k = \frac{EA}{L}$
Cantilever beam (flexural vibration)	$k = \frac{3EI}{L^3}$
Simply supported beam (flexural vibration)	$k = \frac{48EI}{L^3}$
Both end Clamped (flexural vibration)	$k = \frac{192EI}{L^3}$
Clamped - hinged (flexural vibration)	$k = \frac{768EI}{7L^3}$

6. Numerical examples

Flexural and axial vibrations of beams with different boundary conditions are considered. The neural network architecture for the vibration analysis had been trained only flexural vibration conditions. A total of 36 training data sets were presented to the developed neural network. In the testing phase 7 samples were used. All these data are obtained via Rayleigh's method and analytical solutions of the governing equations of flexural vibration of beams with six different boundary conditions. A summary of these data is shown in Table 2. The number of samples usually depends on the characteristics of the problem. In some cases, a large amount of samples does not guarantee that a network can learn better than with smaller samples. The minimum and maximum values for the input data are given in this table. The 10-6-3 and 10-7-3 configuration was also trained for a random distribution of 40 training sets. For these cases, the errors were increased. These configurations are not suitable. The final architecture denoted as 10-5-3 with 10 nodes in input layers, 5 nodes in hidden layer, and 3 nodes in output layer. The neural network was converged at the end of the about 24000 iterations for training data set. It took approximately 6 minutes for training on a standard PC using an artificial neural networks program. This program has developed by the author using C+ programming language (Ülker and Civalek 2001). After the training of the network the momentum term and learning rate values are obtained as 0.82 and 0.86. At the end of the training, the neural network has been tested for cantilever beam (C-F) and both end free (F-F) in the axial vibration conditions. The results found by using ANN are sufficiently close to the exact results. The absolute error of the network is obtained as 2.7% in this case. The obtained results are given in Table 3. In case the flexural vibration, the trained network has been tested for the both end clamped (C-C) and guided-pinned (G-P) support conditions. The results are given in the Table 4 for the first three frequencies. The absolute error of the network is obtained as 1.68% for ω_3 in this case. The variation of the average error with the number of iterations was shown in Fig. 4 for the various momentum rates. The best solution is obtained for the momentum rate of 0.82. Initially, the momentum and learning rate values were 0.3.

Table 2 Intervals of the sampling data for training

Parameter	Minimum	Maximum
m (kg·s²/cm)	0.1	350
L (cm)	40	800
A (cm^2)	24	15000
I (cm^4)	72	45000000
k (kg/cm)	0.3	175
E (kg/cm^2)	280	24000000

Table 3 Comparison of frequencies $(\omega_k)^a$ for axial vibration case

	Free-Free (Case-9) (L = 150 cm)			Clamped-Free (Case-2) (L = 100 cm)		
	Exact results (Paz 1997)	ANN results	Absolute error (%)	Exact results (Paz 1997)	ANN results	Absolute error (%)
ω_1	0.02094	0.020855	0.40	0.01570	0.01548	1.40
ω_2	0.04188	0.04178	0.24	0.04712	0.04584	2.70
ω_3	0.06283	0.06182	1.60	0.07853	0.07824	0.37

$^a\omega_k$ = Tabulated value $\cdot \sqrt{(EA)/m}$

Table 4 Comparison of frequencies $(\omega_k)^a$ for flexural vibration case

	Clamped-Clamped (Case-1) (L = 100 cm)			Guided-Pinned (Case-6) (L = 100 cm)		
	Exact results *(10^{-3}) (Paz 1997)	ANN results *(10^{-3})	Absolute error (%)	Exact results *(10^{-3}) (Hurty *et al*. 1964)	ANN results *(10^{-3})	Absolute error (%)
ω_1	0.4734	0.4698	0.76	0.1571	0.1565	0.38
ω_2	0.7853	0.7912	0.75	0.4712	0.4700	0.25
ω_3	1.0996	1.0985	0.1	0.7852	0.7733	1.68

$^a\omega_k$ = Tabulated value $\cdot \sqrt{(EA)/m}$

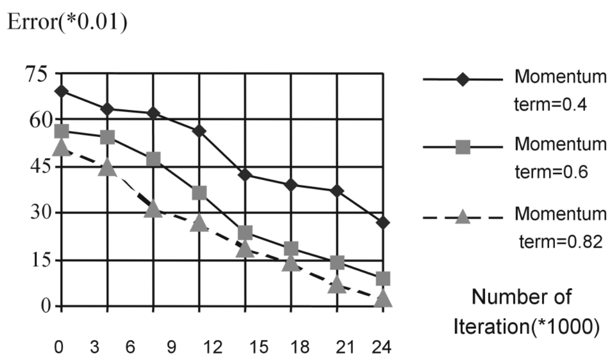
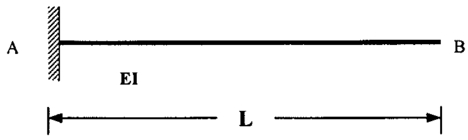
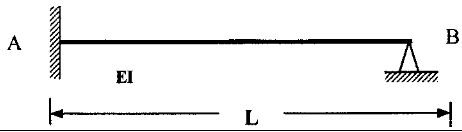
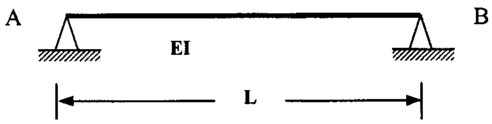
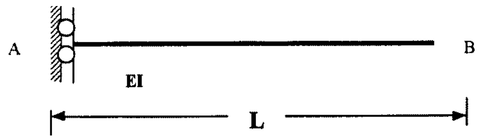
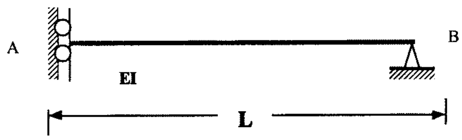
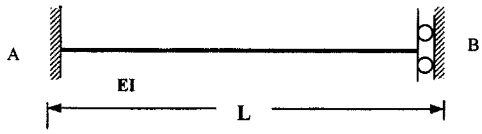
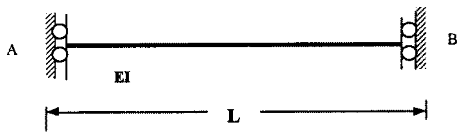
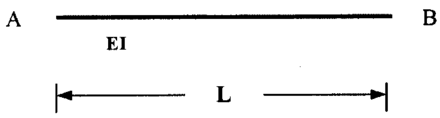
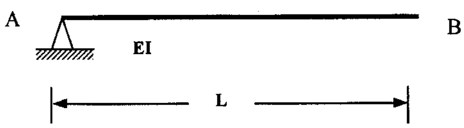


Fig. 4 Comparison of errors and iteration numbers with momentum term

Table 5 Support conditions of the beams

		Support A (a_u, a_v, a_θ)	Support B (b_u, b_v, b_θ)
Case-1 Clamped- Clamped (Tested for flexural vibration case)		(0,0,0)	(0,0,0)
Case-2 Clamped- Free (Tested for axial vibration case)		(0,0,0)	(1,1,1)
Case-3 Clamped- Pinned (Trained for flexural vibration case)		(0,0,0)	(0,0,1)
Case-4 Pinned- Pinned (Trained for flexural vibration case)		(0,0,1)	(0,0,1)
Case-5 Guided- Free (Trained for flexural vibration case)		(0,1,0)	(1,1,1)
Case-6 Guided- Pinned (Tested for flexural vibration case)		(0,1,0)	(0,0,1)
Case-7 Clamped- Guided (Trained for flexural vibration case)		(0,0,0)	(0,1,0)
Case-8 Guided- Guided (Trained for flexural vibration case)		(0,1,0)	(0,1,0)
Case-9 Free- Free (Tested for axial vibration case)		(1,1,1)	(1,1,1)
Case-10 Pinned- Free (Trained for flexural vibration case)		(0,0,1)	(1,1,1)

7. Conclusions

The primary purpose of this study is to illustrate the use of a back-propagation neural network to predict the frequencies of the beams. The designed neural network is a three layer feed-forward neural network consisting of 10 nodes in input, 5 nodes in hidden, and 4 nodes in output layers and was trained by back-propagation algorithm. Neural networks are capable of describing, input-output functional relations, even when a mathematically explicit formula is unavailable. To create such mappings, it suffices to present a neural network with a set of known input-output pairs. During the training process, the interconnection weights of all neurons in the network are computed according to the applied learning rule. It should be noted that once the network was trained, the time required for a given set of input results was nearly instantaneous in PC.

The success of neural network implementation is dependent not just on the quality of the data used for training, but also on the type and structure of the neural network adopted, the method of training, and the way in which both input and output data are structured and interpreted. It has been demonstrated that the ANN approach applied in this study is highly successful for the purposes of frequency analysis of elastic beams. This network can also be used for frequency analysis of different type structures such as plates, shells or frames by doing an easy arrangement of the network topology. However, ANN algorithms cannot, of course, replace totally the conventional numerical techniques, such as finite elements, finite differences and boundary element methods.

References

Adeli, H. and Hung, S.L. (1995), *Machine Learning-Neural Networks, Genetic Algorithms and Fuzzy Systems*, John Wiley & Sons, Inc.

Adeli, H. and Yeh, C. (1989), "Perceptron learning in engineering design", *Microcom. Civil Eng.*, **4**, 247-256.

Adeli, H. and Park, H.S. (1995), "A neural dynamics model for structural optimization-theory", *Comput. Struct.*, **57**(3), 383-390.

Chen, H.M., Qi, G.Z., Yang, J.C.S. and Amini, F. (1995), "Neural networks for structural dynamic model identification", *J. Eng. Mech.*, ASCE, **121**(12), 1377-1381.

Civalek, Ö. (1998a), "Linear and nonlinear static-dynamic analysis of plates and shells by neuro-fuzzy technique", M.Sc. Thesis, University of Fýrat, (in Turkish), Elazığ.

Civalek, Ö. (1998b), "The analysis of rectangular plates via neuro-fuzzy technique, III", *National Computational Mechanic Conferences*, 517-525, November, Istanbul.

Civalek, Ö. (1998c), "The analysis of rectangular plates by neuro-fuzzy technique", *Third National Computational Mechanics Conferences*, 16-18, İ.T.Ü., Istanbul, November.

Civalek, Ö. (1999a), "The analysis of the rectangular plates without torsion via hybrid artificial intelligent technique", *Proc. of the Second International Symposium on Mathematical & Computational Applications*, Azerbaijan, September.

Civalek, Ö. (1999b), "The analysis of circular plates via neuro-fuzzy technique", *J. Eng. Science of Dokuzeylül University*, **1**(2), 13-31.

Clark, S.K. (1972), *Dynamics of Continuous Elements*, Prentice-Hall, New Jersey.

Craig, R.R. Jr., (1981), *Structural Dynamics, An Intoduction to Computer Methods*, John Wiley & Sons, New York.

Eberhart, R.C. and Dobbins, R.W. (1990), *Neural Network PC Tools*, Academic Press, San Diego, California.

Fausett, L. (1994), *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications*, Prentice-Hall, Inc., New Jersey.

Ghaboussi, J., Garrett, Jr. and Wu, X. (1991), "Knowledge-based modeling of material behavior with neural

networks", *J. Struct. Eng.*, ASCE, **117**(1), 132-153.
Ghaboussi, J. and Lin, J.C-C. (1998), "New method of generating spectrum compatible accelerograms using neural networks", *Earthq. Eng. Struct. Dyn.*, **27**, 377-396.
Ghaboussi, J. (1992), "Potential applications of neuro-biological computational models in geotechnical engineering", *Numerical Models in Geomechanics*, Pande and Pietruszczak *(eds)*, 543-555.
Ghaboussi, J. (1993), "An overview of the potential applications of neural networks in civil engineering", in *Proc., ASCE Structures Cong. '93*, Irvine, California.
Hani, K.B. and Ghaboussi, J. (1998), "Neural networks for structural control of a benchmark problem, active tendon system", *Earthq. Eng. Struct. Dyn.*, **27**, 1225-1245.
Hurty, W.C. and Rubinstein, M.F. (1967), *Dynamics of Structures*, Prentice-Hall, New Delhi.
Lee, S.C., Park, S.K. and Lee, B.H. (2001), "Development of the approximate analytical model for the stubgirder system using neural networks", *Comput. Struct.*, **79**, 1013-1025.
Lin, J.C-C. and Ghaboussi, J. (2001), "Generating multiple spectrum compatible accelerograms using stochastic neural networks", *Earthq. Eng. Struct. Dyn.*, **30**, 1021-1042.
Meirovitch, L. (1986), *Elements of Vibration Analysis*, McGraw-Hill.
Park, H.S. and Adeli, H. (1997), "Distributed neural dynamics algorithms for optimization of large steel structures", *J. Struct. Eng.*, ASCE, **123**(7), 880-888.
Paz, M. (1997), *Structural Dynamics, Theory and Computation*, Champman & Hall.
Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986), "Learning internal representation by error propagation in parallel distributed processing: Explorations in the microstructures of cognition", M.I.T. Press, Cambridge.
Szewezyk, Z.P. and Hajela, P. (1994), "Damage detection in structures based on feature-sensitive neural networks", *J. Computing Civil Eng.*, ASCE, **8**(2), 163-178.
Szewezyk, Z.P. and Hajela, P. (1992), "Feature-sensitive neural networks in structural response estimation", *Proc. of Artificial Neural Networks in Eng.*, Ed. Dağli , C., Burke, L.I., Shin, Y.C., ASME, 15-18 November, Missouri.
Thomsan, W.T. and Dahleh, M.D. (1988), *Theory of Vibration with Applications*, Prentice Hall, New Jersey.
Ülker, M. and Civalek, Ö. (2001), "The analysis of circular cylindrical shells via hybrid artificial intelligent technique", *Turkish Chamber of Civil Eng.*, **12**(2), 2401-2417.
Yi, W.H., Lee, H.S., Lee, S.C. and Kim, H.S. (2000), "Development of artificial neural networks based hysteretic model", *Fourteenth Engineering Mechanics Conference*, American Society of Civil Engineers, May 21-24,Austin-Texas, U.S.A.
Yun, C.B. and Bahng, E.Y. (2001), "Joint damage assessment of framed structures using a neural networks technique", *Eng. Struct.*, **23**, 425-435.
Yun, C.B., Yi, J.H. and Bahng, E.Y. (2000), "Substructural identification using neural networks", *Comput. Struct.*, **77**(1), 41-52.
Yun, C.B., Yi, J.H. and Bahng, E.Y. (1998), "Joint damage estimation using neural networks", In: *Proceedings of 5th Pacific Structural Steel Conf.*, Seoul, Korea, Techno Press, 1211-1216.
Yun, C.B. Bahng, E.Y. and Lee, D.G. (1997), "Substructural identification using neural networks", *Proceedings of ICOSSAR '97, The 7th Int. Conf. on Structural Safety and Reliability*, Kyoto, Japan, 24-28. 11, 509-512.
Wu, X., Ghaboussi, J. and Garrett, J.H. (1992), "Use of neural networks in detection of structural damage", *Comput. Struct.*, **42**(4), 649-659.
Zurada, J.M. (1992), *Introduction to Artificial Neural Networks*, West Publishing Com.