Cantilever Beam Natural Frequency Prediction Using Artificial Neural Networks

Sallehuddin Mohamed Haris(✉) and Hamed Mohammadi

Department of Mechanical and Materials Engineering,
Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia
sallehmh@ukm.edu.my, hamed8929@gmail.com

Abstract. Available methods for calculating frequency in cantilever beams have much complexity. In this study we present a new method for calculating natural frequencies in cantilever beams. For this purpose, we use the finite element method (FEM), dynamic analysis and artificial neural network (ANN) techniques to calculate the natural frequency. Finite element software was used to analyze 100 samples of cantilever beams, and the results were used as training and testing data sets in artificial neural networks. For the ANN. the multilayer feed-forward network and back-propagation algorithms were used. We made use of different transfer functions and built 45 different networks in order to find the best network performance. Mean squared error (MSE) was used to evaluate the network performance. Finally, the natural frequencies which were predicted by the ANN techniques were compared to the natural frequencies calculated from theoretical formulation, as well as to those obtained from FEM methods. The results obtained show that the error was quite small.

Keywords: Natural frequency · Artificial neural networks · Mean squared error · Cantilever beams · Finite elements

1 Introduction

Nowadays, consideration of vibration in the design of structures is a significant issue. We can see much difference in the structure design methods used today compared to those used in the past. These days, in structure design, vibration control has become more important than deflection control. The American Institute of Steel Construction (AISC) and the Canadian Standards Association (CSA) have presented some formulations for the determination of the natural frequency in building structures. Other researchers have also presented methods to calculate the natural frequency in beams and plates, but these methods tend to be complex, making their use in vibration control of structures difficult and time consuming, with increased probability of errors. In this study we propose a method, using dynamic analysis, finite element analysis and artificial neural networks for the calculation of the natural frequency in cantilever beams.

There have been many studies on determining the natural frequency in beams and on vibration control in structures. Hamdan and Jubran [1], calculated the natural frequency of a cantilever beam with an intermediate concentrated load, where the effects

Y. Bi et al. (eds.), *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*,
Lecture Notes in Networks and Systems 16, DOI 10.1007/978-3-319-56991-8_33

of concentrated mass on natural frequency, and the effects of natural frequency on the beam vibration were shown.

Bokaian [2], presented the effects of a constant axial tensile force on the natural frequencies and mode shapes of a uniform single-span beam, having different combinations of end conditions. Liu and Haung [3], used the Laplace transform method to study the free vibration of a beam with one concentrated mass at the tip of the beam and another at an intermediate point. Liu and Yeh [4], used the Rayleigh-Ritz method to study the free vibration of restrained non-uniform and cantilever beams with intermediate loads. Goel [5], studied the free vibration of a cantilever beam carrying a concentrated mass at an arbitrary intermediate location. Kounadis [6], studied the free and forced vibrations of a restrained cantilever beam with attached masses. Kojima et al. [7], derived the equations of motion and presented solutions satisfying all boundary conditions at the beam ends. They then used this to investigate the forced response of a cantilever beam having a tip mass and a magnetic vibration absorber at the middle of the beam. Al-Ansari [8], studied two methods for calculating the natural frequency in a stepped cantilever beam; one by using finite element modeling, and the other, using the Rayleigh method. To improve the accuracy in using the Rayleigh method, a new method for calculating the equivalent moment of inertia of the stepped beam was used. He used four different types of beam cross-sections, which were circular, square, rectangular with step in the width only and rectangular with step in the height only. The results using the Rayleigh method was then compared with those from the FEM analysis. Mutra [9], presented an application of artificial neural networks (ANN) in predicting the natural frequency of laminated composite plates. The natural frequencies were calculated by FEM. For training and testing of the ANN model, a number of finite element analyses were carried out and the back propagation algorithm was used to create the network. Hornik et al. [10], showed that the number of hidden layers and neurons will affect the optimum results in each try of a network. He showed that increasing the number of hidden layers can improve the generalization capacity.

2 Dynamic Analysis

To determine the natural frequency in cantilever beams, we first used dynamic analysis to find the governing equations. We define Eq. (1) as the governing differential equation for the cantilever beam shown in Fig. 1.

$$\frac{\partial^2}{\partial x^2}\left[EI(x)\frac{\partial^2 v(x,t)}{\partial x^2}\right]-\frac{\partial}{\partial x}\left[N(x)\frac{\partial v(x,t)}{\partial t}\right]+m(x)\frac{\partial^2 v(x,t)}{\partial t^2}=f(x,t) \tag{1}$$

In Eq. (1), E is the elastic modulus, I is the moment of inertia and m is the mass per unit length of the beam. If we ignore the effects of the lateral forces and assume that the moment of inertia is constant along the beam, Eq. (1) can be written as Eq. (2).

$$\left[EI\frac{\partial^4 v(x,t)}{\partial x^4}\right]+\left[m(x)\frac{\partial^2 v(x,t)}{\partial t^2}\right]=f(x,t) \tag{2}$$

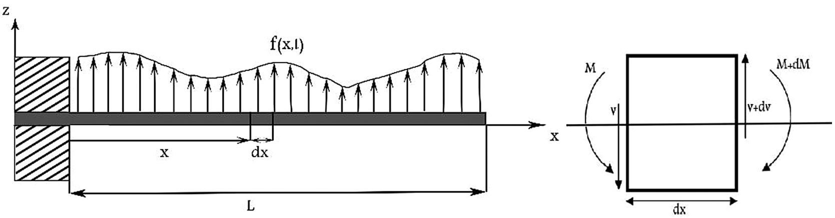

Fig. 1. Cantilever beam with non-uniform loading

$$Equvalently, \frac{EI}{\rho A}\frac{\partial^4 v(x,t)}{\partial x^4} + \frac{\partial^2 v(x,t)}{\partial t^2} = f(x,t) \tag{3}$$

Here, ρ is density and A is the cross section area of the beam. To solve Eq. (3) in the free vibration mode, we used the method of separation of variables of the differential equations, assuming homogeneous mass and constant flexural rigidity throughout the beam. Hence, Eq. (3) can be written as Eq. (4), with $v(x,t) = B(x)T(t)$.

$$\frac{c^2}{B(x)}\frac{d^4B(x)}{dx^4} = -\frac{1}{T(t)}\frac{d^2T(t)}{dt^2},\ c = \sqrt{\frac{EI}{\rho A}} \tag{4}$$

Where $B(x)$ is the displacement function and $T(t)$ is the time function. Then, the solution for displacement is

$$B(x) = C1cos(x\lambda) + C2sin(x\lambda) + \mathrm{C}3cosh(x\lambda) + C4sinh(x\lambda) \tag{5}$$

where

$$\lambda = \left(\frac{\rho A}{EI}\omega^2\right)^{\frac{1}{4}} \tag{6}$$

and C_1, C_2, C_3 and C_4 are constants that can be found from the boundary conditions. For a cantilever beam, at the fixed end, the displacement and slope are zero, but at the free end, the moment and shear are zero. Therefore,

$$x = 0;\ B(0) = 0\ \frac{dB(x)}{dx} = 0 \tag{7}$$

$$x = L;\ \frac{d^2X}{dx^2} = 0,\ \frac{d^3X}{dx^3} = 0 \tag{8}$$

These boundary conditions give $C_2 = -C_4$ and $C_1 = -C_3$.

$$\cos(\beta L)\cosh(\beta L) = -1 \tag{9}$$

Solving Eq. (5) for C_2 and C_1, gives

Then, the roots of Eq. (9) are

$$\beta_n L = \frac{(2n-1)\pi}{2} \tag{10}$$

Now Eq. (4) can be solved for the time function, giving

$$T(t) = k_1 \sin\left[\left(\beta_n^2 \sqrt{\frac{EI}{\rho A}}\right) t\right] + k_2 \cos\left[\left(\beta_n^2 \sqrt{\frac{EI}{\rho A}} t\right)\right] \tag{11}$$

The frequency in rad/s same can be written a

$$\omega = \left(\frac{\beta_n}{L}\right)^2 \sqrt{\frac{EI}{\rho A}} \tag{12}$$

Converting Eq. (12) into Hz gives:

$$f_n = \frac{\omega_n}{2\pi} = \frac{\beta_n^2}{(2\pi L)^2} \sqrt{\frac{EI}{\rho A}} = \frac{\beta_n^2}{(2\pi)^2} \sqrt{\frac{EI}{\rho A L^4}} \tag{13}$$

Now, using $n = 1$ gives the natural frequency as

$$f = \beta \sqrt{\frac{I}{PL^4}} \tag{14}$$

In Eq. (14), f is the natural frequency, I is the moment of inertia, P is dead load per length of beam, L is the length of the beam, and β is a constant that, in this article, will be determined through the use of neural networks. In this study, the beam material is steel, whose elasticity modulus, E is taken to be 2×10^6 kg/cm^2.

3 Prediction of Natural Frequency Using Artificial Neural Networks

Figure 2, shows the network architecture of the ANN used in this study. The input layer consists of 3 items; moment of inertia (I), length of beam (L) and dead load along the beam (P), while the output layer is the natural frequency of the beam (f). For the hidden layers, one and two layer architectures were used.

The transfer functions used were Tangent-sigmoid (Tansis) and Purelin for the hidden layer, and Purelin, Log-sigmoid (Logsis) and Tangent-sigmoid for the output layer.

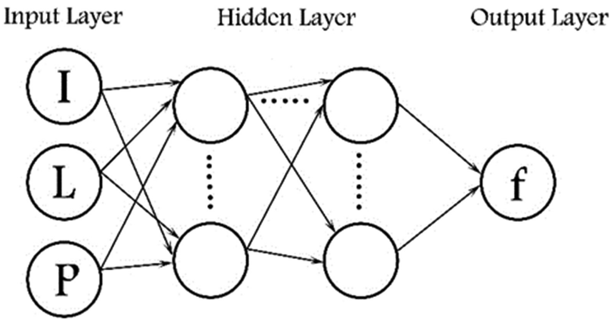

Fig. 2. Network architecture

Carpenter and Barthelemy [11] presented Eq. (15) as a guide for finding the minimum number of neurons in each hidden layer.

$$h_n = i_n + 1 \tag{15}$$

Here, h_n is the number of nodes in the hidden layers and i_n is the number of nodes in the input layer. Hence, in this study the minimum number of nodes in the hidden layer was four.

The next important step in creating a neural network is to determine the number of training data. Increasing the number of training data increases the total training time required, but a larger number of training data improves training performance.

Calculating the needed number of training data depends on the hidden layer nodes, input nodes and output layer nodes. The number of training data used in this work is given by Eq. (16).

$$Tn = h_n(i_n + 1) + o_n(h_n + 1) \tag{16}$$

Where T_n is the minimum number of training data, h_n is the number of hidden layer nodes, i_n is the number of input layer nodes and o_n is the number of output layer nodes. In this study we used 15 nodes for the hidden layer, 3 nodes for input layer (length of beam, moment inertia and dead load) and 1 output layer node (natural frequency of beam), giving

$$Tn = 15(3 + 1) + 1(15 + 1) = 76 \tag{17}$$

To improve results of the network, all data were normalized in the range of [0, 1],

$$X_n = X_i / X_{max} \tag{18}$$

Where X_n is the normalized data, X_i is the original data, and X_{max} is the maximum value of all data.

Evaluation of the network performance was based on the Mean Squared Error (MSE) as shown in Eq. (19), where R_s is the estimated value and R_a is the actual value.

Table 1. Details of the neural networks used in this study

Number of data	100 data: 80 training, 15 validation, 5 testing
Input layer	Length of beam, Moment of inertia, Load
Number of neurons in hidden layers	5, 10, 15, 20
Output layer	Natural frequency of beam
Net architecture	3-h_1-1 and 3-h_1-h_2-1
Network type	Multilayer feed-forward
Net algorithm	Back-propagation
Training function	Trainbr and Trainlm
Learning function	Learngd and Learngdm
Output transfer function	Tangent Sigmoid and Purelin
Hidden transfer function	Log-Sigmoid, Tangent Sigmoid, Purelin
Performance function	MSE

The best performance would be given by the lowest MSE value. Table 1 shows the details of the neural network used.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(R_s - R_a)^2 \tag{19}$$

In this study we generated 45 different networks, where the details of the networks are shown in Tables 2, 3 and 4. Table 2 shows the MSE and the coefficient of determination (R^2) results for the network with one hidden layer and Purelin and Tansis

Table 2. MSE and R^2 results - one hidden layer and Purelin as output transfer function

Train function	Learn function	Transfer function	MSE & R^2	The number of neurons in hidden layer							
				5		10		15		20	
				Train	Test	Train	Test	Train	Test	Train	Test
Trainlm	Learngd	Purelin	MSE	0.163	0.169	0.157	0.171	0.131	0.129	0.178	0.195
			R^2	0.841	0.845	0.848	0.855	0.869	0.872	0.837	0.835
		Tansis	MSE	0.165	0.188	0.15	0.154	0.156	0.164	0.177	0.168
			R^2	0.869	0.883	0.863	0.864	0.868	0.871	0.868	0.874
	Learngdm	Purelin	MSE	0.174	0.191	0.178	0.182	0.173	0.192	0.176	0.181
			R^2	0.875	0.839	0.842	0.834	0.866	0.868	0.843	0.853
		Tansis	MSE	0.185	0.159	0.161	0.172	0.148	0.176	0.153	0.144
			R^2	0.866	0.904	0.783	0.883	0.846	0.889	0.888	0.862
Trainbr	Learngd	Purelin	MSE	0.168	0.146	0.158	0.163	0.168	0.173	0.157	0.176
			R^2	0.870	0.890	0.848	0.856	0.863	0.836	0.856	0.867
		Tansis	MSE	0.110	0.121	0.098	0.087	0.073	0.075	0.119	0.121
			R^2	0.921	0.902	0.940	0.951	0.963	0.961	0.913	0.902
	Learngdm	Purelin	MSE	0.116	0.180	0.189	0.174	0.147	0.163	0.151	0.179
			R^2	0.912	0.885	0.876	0.843	0.883	0.877	0.838	0.866
		Tansis	MSE	0.154	0.174	0.189	0.136	0.168	0.148	0.167	0.152
			R^2	0.877	0.845	0.839	0.868	0.853	0.829	0.875	0.860

Table 3. MSE and R^2 results - one hidden layer and tansis as output transfer function

Neurons in hidden layer	Training and learning function: TRAINBR and LEARNGDM				Hidden layer transfer function
	R^2			Training MSE	
	Training	Validation	Testing		
10	0.989	0.992	0.971	0.0263	Purelin
	0.979	0.964	0.952	0.0313	Logsis
15	0.999	0.997	0.999	0.0161	Purelin
	0.982	0.976	0.948	0.0286	Logsis
20	0.915	0.931	0.909	0.0387	Purelin
	0.938	0.929	0.925	0.0351	Logsis

Table 4. MSE and R^2 results - two hidden layers and purelin as output transfer function

Neurons in hidden layer		Training and learning function: TRAINBR and LEARNGDM				Hidden layer transfer function	
		R^2			Training MSE		
Hidden layer 1	Hidden layer 2	Training	Validation	Testing		Hidden layer 1	Hidden layer 2
10	10	0.981	0.975	0.977	0.0272	Purelin	Logsis
		0.992	0.984	0.988	0.0247	Logsis	Purelin
		0.974	0.964	0.961	0.0301	Purelin	Purelin
5	5	0.999	0.999	0.999	0.0173	Tangsis	Tangsis
		0.994	0.991	0.990	0.0231	Tangsis	Logsis
15	10	0.941	0.932	0.920	0.0342	Purelin	Tangsis
		0.959	0.967	0.973	0.0328	Logsis	Logsis

as the output transfer functions. Here, the numbers of nodes used in the hidden layer were 5, 10, 15 and 20, the training functions used were Trainbr (Bayesian Regularization) and Trainlm (Levenberg Marquart), the learning functions were Learngd (Gradient descent) and Learngdm (Gradient descent with momentum). Table 3 gives the MSE and R^2 results for networks with one hidden layer and Tansis as the output transfer function, where fixed training and learning functions (Trainbr and Learngdm) were used. The hidden layer transfer function and number of neurons were varied to find the combination that gives the best network performance. Table 4 shows the results for networks with two hidden layers where the output transfer function was Purelin, and the fixed training and learning functions were Trainbr and Learngdm. Here, the hidden layers had 5, 10 or 15 neurons, and the hidden layer transfer functions were combinations of Tansis, Purelin and Lonsig.

The performance of the neural networks depends on many factors such as the training and testing parameters, and the range of data values. From all the created networks, we selected two that gave the best performances. The selection was based on the MSE and R^2 results from the whole range of networks. Table 5 shows the parameter values for the two selected networks, denoted Net 1 and Net 2.

Table 5. MSE, R^2 and other parameters of Net 1 and Net 2 - the best performance between all 45 networks

No.	Net architecture	Neurons and transfer function in hidden layer		Training function	Learning function	Output transfer function	R^2 (All)	MSE	
		1st	2nd					Train	Test
Net 1	3-15-1	15 Purelin	–	Trainbr	Learngdm	Tangsis	0.999	0.0161	0.0168
Net 2	3-5-5-1	5 Tangsis	5 Tangsis	Trainbr	Learngdm	Purelin	0.999	0.0173	0.0177

Figure 3 shows the R^2 results for Net 2. The horizontal axis is the target frequency obtained from FEM analysis and the vertical axis is the predicted frequency obtained from ANN. The regression analysis graph gives a straight line, which means low error

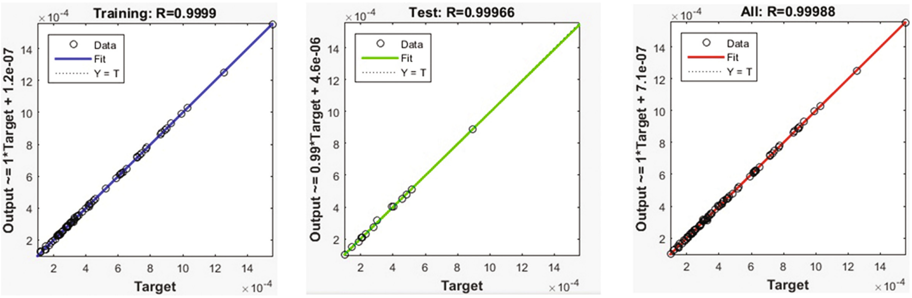

Fig. 3. Regression analysis for Net 2

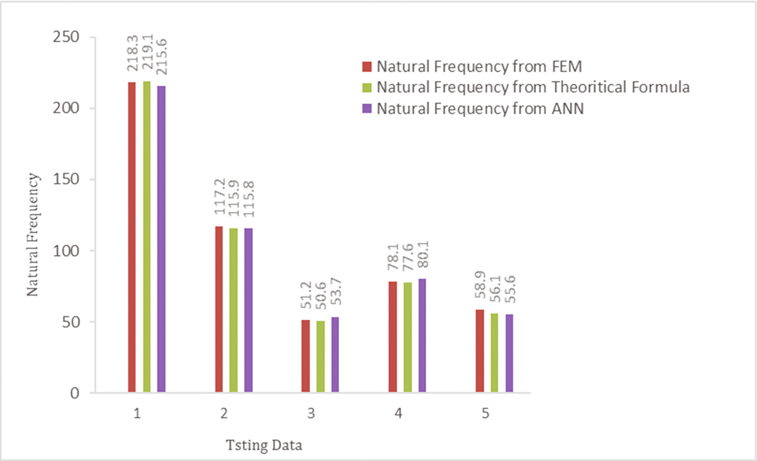

Fig. 4. Comparison of natural frequency from 5 test data between 3 methods

and good performance. A comparison of the predicted frequency from generated neural networks to the frequencies obtained from theory and FEM analysis is shown in Fig. 4. A close correlation is clearly evident.

4 Conclusion

There are many different methods for the calculation of the natural frequency of cantilever beams. However, these methods tend to be complex. Here, a simple method to predict the natural frequency using artificial neural networks has been presented. Both generated networks used in this study were found to be trustworthy and can be used as references for natural frequency calculation in steel cantilever beams of any length and any cross section, and subjected to any lateral load. While this work has concentrated on steel cantilever beams, it is envisaged that in future, the method can be extended to beams made from other materials as well.

References

1. Hamdan, M.N., Jubran, B.A.: Free and force vibrations of a restrained cantilever beam carrying a concentrated mass. J. King Abdulaziz Univ. **3**, 71–83 (2012)
2. Bokaian, A.: Natural frequencies of beams under tensile axial loads. Sound Vib. **142**, 481–498 (1990)
3. Liu, W.H., Haung, C.C.: Free vibration of beams with elastically restrained edges and intermediate concentrated masses. Sound Vib. **122**, 193–207 (1988)
4. Liu, W.H., Yeh, F.H.: Free vibration of restrained non uniform beam with intermediate masses. Sound Vib. **117**, 555–570 (1987)
5. GoeI, R.P.: Vibration of a beam carrying concentrated mass. J. Appl. Mech. **40**, 821–822 (1973)
6. Kounadis, A.N.: Dynamic response of cantilevers with attached masses. Eng. Mech. Div. **101**, 695–706 (1975)
7. Kojima, H., Nagaya, K., Niiyama, N., Nakai, K.: Vibration control of a beam structure using an electromagnetic damper with velocity feedback. Bull. JSME **29**, 2653–2659 (1986)
8. Al-Ansari, L.S.: Calculating static deflection and natural frequency of stepped cantilever beam using modified rayleigh method. Mech. Prod. Eng. Res. Dev. **3**, 113–124 (2013)
9. Reddy, M.R.S., Reddy, B.S., Reddy, V.N., Sreenivasulu, S.: Prediction of natural frequency of laminated composite plates using artificial neural networks. Sci. Res. Eng. **4**, 329–337 (2012)
10. Hornik, K., Stinchcombe, M., White, H.: Universal approximation of an unknown mapping and its derivative using multilayer feed forward networks. Neural Netw. **3**, 551–560 (1990)
11. Carpenter, W.C., Barthelemy, J.F.: Common misconceptions about neural networks as approximates. Comput. Civ. Eng. **8**, 345–358 (1994)