University of Tartu

Faculty of Mathematics and Computer Science

Institute of Computer Science

Information Technology

Mairit Vikat

Identification of Non-Classical Boundary Conditions with the Aid of Artificial Neural Networks

Master's thesis (30 EAP)

Supervisor: PhD Helle Hein

Author: …… ……. May 2012

Supervisor: …… ……. May 2012

Allowed to defense

Professor: …….. ……. May 2012

Tartu 2012

Table of Contents

Introduction

The free vibration of elastically restrained beams is a subject of practical engineering interest that has been studied by various investigators over the years. One of the most popular theories dealing with the vibration of beams is the Euler-Bernoulli beam theory.

In the current thesis, the frequency equations for the Euler-Bernoulli beams with non-classical boundary conditions are considered. Two types of beams are studied: beams with elastic supports at the boundaries, and beams with intermediate elastic support.

The elastic supports of beams play an important role in structural performance. In the case of vibrating structures the frequencies of vibration depend on the stiffness parameters of elastic supports. The stiffness characteristics of elastic supports can change during the exploitation of structures due to environment conditions or damages, and as a result, they can significantly influence the performance of the structures. Therefore, it is important to identify these parameters online during the exploitation.

The calculation of the stiffness parameters of the support conditions from the governing equations of the vibrating beams is an inverse problem and cannot be done analytically. Therefore, some alternatives for reaching the goal of the task could be considered. One option is to use artificial neural networks.

Artificial neural networks (ANNs) are a simulation of biological neural networks which means that they are capable of learning by examples. Since ANNs are able to find relationships between input and output data, they can be trained to produce a desired output based on the input. In terms of the vibrating Euler-Bernoulli beams, this means that all of the frequency equations needed to solve a task do not need to be computed or measured anymore; instead, a smaller amount of obtained information is sufficient, and the rest of the necessary data could be predicted by the artificial neural network.

The main goal of the current thesis is to study the free vibration of the Euler-Bernoulli beams with non-classical boundary conditions and to analyze the efficiency of predicting the support coefficients based on the calculated training data provided to the neural networks. For both types of beams – beams with elastic supports at the boundaries and

beams with intermediate elastic support – several examples with different support conditions are investigated.

The thesis is arranged into three main chapters. Chapter 1 provides an overview of the Euler-Bernoulli theory and the Euler-Bernoulli beam model. The cases of possible boundary conditions are introduced, and two examples of finding general solutions to the beam models are provided.

In Chapter 2, the basics of artificial neural networks are presented. The section explains the basic idea of a perceptron, and gives an overview of the architectures and learning methods of artificial neural networks. The learning algorithm used in the practical implementation of the current thesis – called the error back-propagation – is characterized in detail.

The third chapter – the practical implementation of the thesis – focuses on two types of beams: cases of beams with elastic supports at the boundaries, and cases of beams with intermediate elastic support. For predicting the support coefficients of the beams, artificial neural networks are created and trained in the MATLAB environment. For each case, the acquired test results are compared to the expected results, and characterized based on predefined efficiency parameters.

1 Euler-Bernoulli beam theory

1.1 Overview of the Euler-Bernoulli beam theory

A beam is a common structural element in structural and mechanical engineering – some examples of beams are a bookshelf, the frame of a car, the plank of a seesaw [1]. Beams are capable of enduring vertical (sometimes also horizontal) load by bending [2].

The free vibration of beams is explored in attempt to simulate the dynamics of different structural and mechanical components [3].

One of the theories that deal with the vibration of beams is the Euler-Bernoulli beam theory. It is used for calculating the load-carrying and deflection characteristics of a beam. The Euler-Bernoulli beam theory was first formed by Leonhard Euler and Daniel Bernoulli in the middle of the 18th century, but it became the cornerstone of engineering only after the construction of the Eiffel Tower and Ferris wheel in the late 19th century [4].

The basic assumptions of the Euler-Bernoulli beam theory are:

1. The length of the beam is significantly larger than the width and thickness of the beam.
2. The material of the beam is linear-elastic (strain is directly proportional to stress).
3. Planes perpendicular to the neutral axis remain perpendicular after deformation [5].

1.2 The Euler-Bernoulli beam model

In the present paper, the frequency equations for the Euler-Bernoulli beams with elastically retrained end and intermediate supports are examined. The restraints are provided by either a translational or rotational spring, or both.

Figure 1 represents a simple beam that is clamped at the left end and has a translational and a rotational spring at the right end.

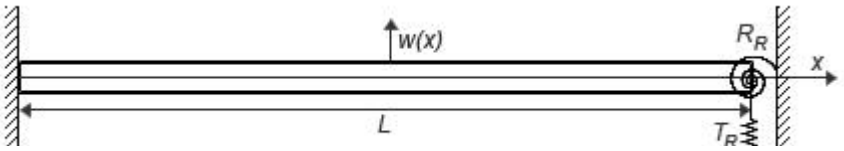

Figure 1. Beam clamped at left end, rotationally and translationally restrained at right end.

On Figure 1, $w(x)$ is the deflection, x is the location at distance x along the length of the beam L, R_R is the rotational spring constant at right end $(x = L)$ and T_R is the translational spring constant at right end $(x = L)$. For the free vibration of the beam, also EI – the flexural rigidity, A – the cross-sectional area of the beam, and ρ – density of the material need to be considered.

The free vibrations of a beam are described by the equation [6]

$$EI\frac{\partial^4 W(x,t)}{\partial x^4} + \rho A\frac{\partial^2 W(x,t)}{\partial t^2} = 0. \tag{1.1}$$

By dividing it with ρA, we get

$$\frac{EI}{\rho A}\cdot\frac{\partial^4 W(x,t)}{\partial x^4} + \frac{\partial^2 W(x,t)}{\partial t^2} = 0. \tag{1.2}$$

The function $W(x,t)$ depends on distance x and time t. For free vibrations the solution can be sought in the form

$$W(x,t) = w(x)\cdot sin(\omega t), \tag{1.3}$$

where ω is the natural frequency and $w(x)$ is the mode shape of the beam. Substituting (1.3) into (1.2) and eliminating the trivial solution $sin(\omega t) = 0$ we obtain

$$EI\cdot w^{IV}(x) - \rho A\omega^2 w(x) = 0, \tag{1.4}$$

where E is Young's modulus of elasticity, I is the moment of inertia, ρ is the material mass density, and A is the cross-sectional area of the beam.

Introducing the non-dimensional quantities

$$\xi = \frac{x}{L}, k^4 = \frac{\rho A \omega^2 L^4}{EI}, \tag{1.5}$$

the equation of mode shapes can be presented as

$$w^{IV} - \mathrm{k}^4 w = 0. \tag{1.6}$$

The general solution of the equation (1.6) is as follows:

$$w = c_1 \sin k\xi + c_2 \cos k\xi + c_3 \sinh k\xi + c_4 \cosh k\xi. \tag{1.7}$$

Here, c_1, c_2, c_3, c_4 are the integration constants, k is the natural frequency parameter we are looking for, and $w = w(\xi),\ \xi \in [0, 1]$.

1.2.1 Model of a beam with intermediate support

A beam with an intermediate support can be represented in a similar way. Figure 2 shows a beam that is clamped at left end, has a rotational and a translational spring at right end, and a translational spring as an intermediate support.

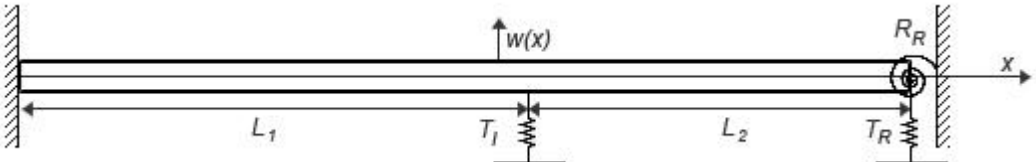

Figure 2. Beam clamped at left end, translationally supported in the center, rotationally and translationally supported at right end.

In case of a beam with intermediate support, the beam is divided into two parts with two different coordinate systems. The length of the left part of the beam is L_1, whereas the length of the right part of the beam is L_2. The general solution of the left part of the beam can be expressed with the following equation:

$$w_1 = c_1 \sin k_1\xi_1 + c_2 \cos k_1\xi_1 + c_3 \sinh k_1\xi_1 + c_4 \cosh k_1\xi_1. \tag{1.8}$$

In this case,

$$\xi_1 = \frac{x_1}{L_1}, w_1 = \frac{w}{L_1}, x_1 \in [0, L_1]. \tag{1.9}$$

The frequency parameter we are looking for can be expressed as follows:

$$k_1^4 = \frac{\rho A \omega^2 L_1^4}{EI} = \left(\frac{L_1}{L}\right)^4 \Omega^2, \tag{1.10}$$

where

$$\Omega^2 = \frac{\rho A \omega^2 L^4}{EI}. \tag{1.11}$$

The equation for the general solution of the right part of the beam is

$$w_2 = c_5 \sin k_2\xi_2 + c_6 \cos k_2\xi_2 + c_7 \sinh k_2\xi_2 + c_8 \cosh k_2\xi_2, \tag{1.12}$$

where

$$\xi_2 = \frac{x_2}{L_2}, w_2 = \frac{w}{L_2}, x_2 \in [0, L_2]. \tag{1.13}$$

The frequency parameter in this case is

$$k_2{}^4 = \frac{\rho A \omega^2 L_2^4}{EI} = \left(\frac{L_2}{L}\right)^4 \Omega^2. \tag{1.14}$$

Here, Ω^2 is the same as in equation (1.11).

1.3 Cases of boundary conditions

In the present thesis, the following boundary conditions of a beam are considered on the left end:

1. Clamped:

$$\begin{cases} w(0) = 0 \\ w'(0) = 0 \end{cases}$$

2. Simply supported:

$$\begin{cases} w(0) = 0 \\ w''(0) = 0 \end{cases}$$

3. Free:

$$\begin{cases} w''(0) = 0 \\ w'''(0) = 0 \end{cases}$$

4. Guided/sliding:

$$\begin{cases} w'(0) = 0 \\ w'''(0) = 0 \end{cases}$$

5. Sliding with translational spring:

$$\begin{cases} w'''(0) + K_{TL} w(0) = 0 \\ w'(0) = 0 \end{cases}$$

6. Free with translational spring:

$$\begin{cases} w'''(0) + K_{TL} w(0) = 0 \\ w''(0) = 0 \end{cases}$$

7. Translational and rotational spring:

$$\begin{cases} w'''(0) + K_{TL} w(0) = 0 \\ w''(0) - K_{RL} w'(0) = 0 \end{cases}$$

On the right end, the equations for the boundary conditions are analogous:

1. Clamped:

$$\begin{cases} w(1) = 0 \\ w'(1) = 0 \end{cases}$$

2. Simply supported:

$$\begin{cases} w(1) = 0 \\ w''(1) = 0 \end{cases}$$

3. Free:

$$\begin{cases} w''(1) = 0 \\ w'''(1) = 0 \end{cases}$$

4. Guided/sliding:

$$\begin{cases} w'(1) = 0 \\ w'''(1) = 0 \end{cases}$$

5. Sliding with translational spring:

$$\begin{cases} w'''(1) + K_{TR}w(1) = 0 \\ w'(1) = 0 \end{cases}$$

6. Free with translational spring:

$$\begin{cases} w'''(1) - K_{TR}w(1) = 0 \\ w''(1) = 0 \end{cases}$$

7. Translational and rotational spring:

$$\begin{cases} w'''(1) - K_{TR}w(1) = 0 \\ w''(1) + K_{RR}w'(1) = 0 \end{cases}$$

In the boundary constraints the following dimensionless translational and rotational stiffness coefficients at the ends of the beam have been introduced:

$$K_{TL} = \frac{T_L L^3}{EI}, K_{TR} = \frac{T_R L^3}{EI}, K_{RL} = \frac{R_L L}{EI}, K_{RR} = \frac{R_R L}{EI}.$$

For the following part – finding the constants in the general solution to the mode shape – the first, second and third derivations of *w(x)* are needed:

$$w = c_1 \sin k\xi + c_2 \cos k\xi + c_3 \sinh k\xi + c_4 \cosh k\xi\,, \tag{1.15}$$

$$w' = k(c_1 \cos k\xi - c_2 \sin k\xi + c_3 \cosh k\xi + c_4 \sinh k\xi), \tag{1.16}$$

$$\mathrm{w}'' = \mathrm{k}^2(-c_1 \sin k\xi - c_2 \cos k\xi + c_3 \sinh k\xi + c_4 \cosh k\xi), \tag{1.17}$$

$$w''' = k^3(-c_1 \cos k\xi + c_2 \sin k\xi + c_3 \cosh k\xi + c_4 \sinh k\xi). \tag{1.18}$$

The equation and derivations on the left end of the beam are respectively the equation and the derivations on the position $x = 0$:

$$w(0) = c_2 + c_4 \tag{1.19}$$

$$w'(0) = k(c_1 + c_3) \tag{1.20}$$

$$w''(0) = k^2(-\mathrm{c}_2 + \mathrm{c}_4) \tag{1.21}$$

$$w'''(0) = k^3(-\mathrm{c}_1 + \mathrm{c}_3) \tag{1.22}$$

The equation and derivations on the right end are on the position $x = 1$:

$$w(1) = c_1 \sin k + c_2 \cos \mathrm{k} + c_3 \sinh k + c_4 \cosh k \tag{1.23}$$

$$w'(1) = k(c_1 \cos k - c_2 \sin k + c_3 \cosh k + c_4 \sinh k) \tag{1.24}$$

$$w''(1) = k^2(-c_1 \sin k - c_2 \cos k + c_3 \sinh k + c_4 \cosh k) \tag{1.25}$$

$$w'''(1) = k^3(-c_1 \cos k + c_2 \sin k + c_3 \cosh k + c_4 \sinh k) \tag{1.26}$$

1.3.1 Beam clamped at the left end and having translational and rotational spring at the right end

As an example, we are going to find a general solution for the beam which is clamped at left end and has translational and rotational support at right end.

To find the frequencies, we need to construct a four by four matrix where the first two rows each contain the four coefficients from the equations of the left end of the beam and the following two rows each contain the four coefficients from the equations of the right end of the beam.

According to the boundary condition of the left end – clamped – the equations of the left end of the beam, using equations (1.19) and (1.20) are as follow:

$$\begin{cases} w(0) = \mathrm{c}_2 + \mathrm{c}_4 = 0 & (1.27) \\ w'(0) = k(c_1 + c_3) = 0 & (1.28) \end{cases}$$

On the right end, supported with a translational and rotational spring, we use equations (1.23)-(1.26) and get

$$\begin{cases} w'''(1) - K_{TR}w(1) = k^3(-c_1\cos k + c_2\sin k + c_3\cosh k + c_4\sinh k) - \\ \qquad -K_{TR}(\mathrm{c}_1\sin\mathrm{k} + \mathrm{c}_2\cos\mathrm{k} + \mathrm{c}_3\sinh\mathrm{k} + \mathrm{c}_4\cosh\mathrm{k}) = 0 & (1.29) \\ w''(1) + K_{RR}w'(1) = k^2(-\mathrm{c}_1\sin\mathrm{k} - \mathrm{c}_2\cos\mathrm{k} + \mathrm{c}_3\sinh\mathrm{k} + \mathrm{c}_4\cosh\mathrm{k}) + \\ \qquad +K_{RR}(c_1\cos k - c_2\sin k + c_3\cosh k + c_4\sinh k) = 0 & (1.30) \end{cases}$$

Using the equations (1.27)-(1.30) we create a matrix D with the coefficients from these equations:

$$D = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}, \qquad (1.31)$$

where the coefficients denoted with a_{ij} are respectively

$$a_{31} = -k^3\cos k - K_{TR}\sin k, \quad a_{32} = k^3\sin k - K_{TR}\cos k,$$

$$a_{33} = k^3\cosh k - K_{TR}\sinh k, \quad a_{34} = k^3\sinh k - K_{TR}\cosh k,$$

$$a_{41} = -k^2 \sin k + K_{RR} \cos k, \quad a_{42} = -k^2 \cos k - K_{RR} \sin k,$$

$$a_{43} = k^2 \sinh k + K_{RR} \cosh k, \quad a_{44} = k^2 cosh + K_{RR} \sinh k$$

The determinant of the matrix *D* has to be equal to zero for a non-trivial solution to exist in this homogeneous system [7].

1.3.2 Beam with intermediate support

Similarly to the previous example, the coefficients for the general solution for the beam with intermediate support can be found. In this example, we consider a beam that has sliding with translational spring at left end, a translational and a rotational spring as an intermediate support, and is simply supported at the right end.

Similarly to simple supported beam, a matrix with the coefficients of the general solution has to be constructed, but in this case the dimensions of the matrix are eight by eight.

The first two rows of the matrix contain coefficients from the equations of the left end of the beam; the next two rows contain coefficients from equations of the right end of the beam. The following four rows contain coefficients from the equations of the intermediate part of the beam.

The boundary condition at the left end of the beam is sliding with translational spring. Using the equations (1.19), (1.20) and (1.22) we get the equations:

$$\begin{cases} w_1'''(0) + K_{TL}\left(\frac{L_1}{L}\right)^3 w_1(0) = {k_1}^3(-c_1 + c_3) + K_{TL}\left(\frac{L_1}{L}\right)^3 (c_2 + c_4) = 0 & (1.32) \\ w_1'(0) = k_1(c_1 + c_3) = 0 & (1.33) \end{cases}$$

By using equations (1.23) and (1.25) we get the equations for the simply supported right end:

$$\begin{cases} w(1) = c_5 \sin k_2 + c_6 \cos k_2 + c_7 \sinh k_2 + c_8 \cosh k_2 = 0 & (1.34) \\ w''(1) = {k_2}^2(-c_5 \sin k_2 - c_6 \cos k_2 + c_7 \sinh k_2 + c_8 \cosh k_2) = 0 & (1.35) \end{cases}$$

For the equations on the intermediate part of the beam we get the following equations:

$$\begin{cases} w_1(1) - \dfrac{L_2}{L_1} w_2(0) = 0 & (1.36) \\ w_1'(1) - w_2'(0) = 0 & (1.37) \\ K_R \dfrac{L_2}{L} w_1'(1) + \dfrac{L_2}{L_1} w_1''(1) - w_2''(0) = 0 & (1.38) \\ K_T \dfrac{L_1}{L} \left(\dfrac{L_2}{L}\right)^2 w_1(1) - \left(\dfrac{L_2}{L_1}\right)^2 w_1'''(1) + w_2'''(0) = 0 & (1.39) \end{cases}$$

where K_R is the rotational constant in the intermediate point and K_T is the translational constant in the intermediate point.

Using equations (1.19)-(1.26) we get the following equations from (1.36)-(1.39):

$$\begin{cases} c_1 \sin k_1 + c_2 \cos k_1 + c_3 \sinh k_1 + c_4 \cosh k_1 - \dfrac{L_2}{L_1}(c_6 + c_8) = 0 & (1.40) \\ k_1(c_1 \cos k_1 - c_2 \sin k_1 + c_3 \cosh k_1 + c_4 \sinh k_1) - k_2(c_5 + c_7) = 0 & (1.41) \\ K_R \dfrac{L_2}{L} k_1(c_1 \cos k_1 - c_2 \sin k_1 + c_3 \cosh k_1 + c_4 \sinh k_1) + \\ + \dfrac{L_2}{L_1} {k_1}^2(-c_1 \sin k_1 - c_2 \cos k_1 + c_3 \sinh k_1 + c_4 \cosh k_1) - {k_2}^2(-c_6 + c_8) = 0 & (1.42) \\ K_T \dfrac{L_1}{L} \left(\dfrac{L_2}{L}\right)^2 (c_1 \sin k_1 + c_2 \cos k_1 + c_3 \sinh k_1 + c_4 \cosh k_1) - \\ - \left(\dfrac{L_2}{L_1}\right)^2 {k_1}^3(-c_1 \cos k_1 + c_2 \sin k_1 + c_3 \cosh k_1 + c_4 \sinh k_1) + {k_2}^3(-c_5 + c_7) = 0 & (1.43) \end{cases}$$

Using equations (1.40)-(1.43) we create an eight by eight matrix of the coefficients:

$$D = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & 0 & 0 & 0 \\ a_{21} & 0 & a_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{35} & a_{36} & a_{37} & a_{38} \\ 0 & 0 & 0 & 0 & a_{45} & a_{46} & a_{47} & a_{48} \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & a_{56} & 0 & a_{58} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & a_{67} & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & 0 & a_{76} & 0 & a_{78} \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & 0 & a_{87} & 0 \end{pmatrix}, \quad (1.44)$$

where a_{ij} are as follows:

$$a_{11} = -{k_1}^3, \quad a_{12} = K_{TL} \left(\frac{L_1}{L}\right)^3,$$

$$a_{13} = {k_1}^3, \quad a_{14} = K_{TL}\left(\frac{L_1}{L}\right)^3,$$

$$a_{21} = k_1, \quad a_{23} = k_1,$$

$$a_{35} = \sin k_2, \quad a_{36} = \cos k_2,$$

$$a_{37} = \sinh k_2, \quad a_{38} = \cosh k_2,$$

$$a_{45} = -\sin k_2, \quad a_{46} = -\cos k_2,$$

$$a_{47} = \sinh k_2, \quad a_{48} = \cosh k_2$$

$$a_{51} = \sin k_1, \quad a_{52} = \cos k_1,$$

$$a_{53} = \sinh k_1, \quad a_{54} = \cosh k_1,$$

$$a_{56} = \frac{L_2}{L_1}, \quad a_{58} = \frac{L_2}{L_1},$$

$$a_{61} = k_1 \cos k_1, \quad a_{62} = -k_1 \sin k_1,$$

$$a_{63} = k_1 \cosh k_1, \quad a_{64} = k_1 \sinh k_1,$$

$$a_{65} = -k_2, \quad a_{67} = -k_2,$$

$$a_{71} = K_R \frac{L_2}{L} k_1 \cos k_1 - \frac{L_2}{L_1} {k_1}^2 \sin k_1, \quad a_{72} = -K_R \frac{L_2}{L} k_1 \sin k_1 - \frac{L_2}{L_1} {k_1}^2 \cos k_1,$$

$$a_{73} = K_R \frac{L_2}{L} k_1 \cosh k_1 + \frac{L_2}{L_1} {k_1}^2 \sinh k_1, \quad a_{74} = K_R \frac{L_2}{L} k_1 \sinh k_1 + \frac{L_2}{L_1} {k_1}^2 \cosh k_1,$$

$$a_{76} = {k_2}^2, \quad a_{78} = -{k_2}^2,$$

$$a_{81} = K_T \frac{L_1}{L}\left(\frac{L_2}{L}\right)^2 \sin k_1 + \left(\frac{L_2}{L_1}\right)^2 {k_1}^3 \cos k_1, \quad a_{82} = K_T \frac{L_1}{L}\left(\frac{L_2}{L}\right)^2 \cos k_1 - \left(\frac{L_2}{L_1}\right)^2 {k_1}^3 \sin k_1,$$

$$a_{83} = K_T \frac{L_1}{L}\left(\frac{L_2}{L}\right)^2 \sinh k_1 - \left(\frac{L_2}{L_1}\right)^2 {k_1}^3 \cosh k_1, \quad a_{84} = K_T \frac{L_1}{L}\left(\frac{L_2}{L}\right)^2 \cosh k_1 - \left(\frac{L_2}{L_1}\right)^2 {k_1}^3 \sinh k_1,$$

$$a_{85} = -{k_2}^3, \quad a_{87} = {k_2}^3$$

The rest of the combinations of boundary conditions of the beams can be treated in a similar way.

As an example, some numerical values of the frequencies calculated for a beam clamped at left end and free at right end (called a cantilever beam), with translational and rotational springs at the intermediate support are depicted in Table 1. The calculated values of the rotational stiffness coefficients k_r = {10, 100, 1000} are outlined along the values reported by Lau [8], based on the fixed value of k_t = 10 of translational stiffness coefficient. The results are compared for modes three, four and five. The left column for each mode contains the results acquired by the author of the current thesis, and the right column contains results presented by Lau [8].

Table 1. Frequency parameters for a cantilever beam with translational and rotational intermediate support. Comparison to the results reported by Lau [8].

	mode = 3		**mode = 4**		**mode = 5**	
k_r	k_t **= 10**	k_t **= 10** [8]	k_t **= 10**	k_t **= 10** [8]	k_t **= 10**	k_t **= 10** [8]
10	8,023621	8,02362	11,002136	11,00212	14,205721	14,20572
100	8,423806	8,42381	11,008224	11,00821	14,355295	14,35528
1000	8,599826	8,59981	11,011199	11,01121	14,415894	14,41589

The values presented by Lau [8] and the values obtained with the present approach after rounding the values to five decimal points are almost equal. Only in a few cases, there is a slight difference of size 0.00001 or 0.00002.

2 Artificial neural networks

Artificial neural networks have been influenced by the natural networks of biological neurons in the (human) brain [9]. Data processing in the brain is carried out by highly interconnected neurons which send out electric impulses through the neural network [10]. Each neuron is a cell that uses biochemical reactions to receive, process, and transmit information. Among the neurons there are dendrites (treelike networks of nerve fibers connected to the cell body) and axons (a single long fiber extending from the cell body) which are connected to other neurons through synapses [11] (the links between one neuron's axon and another's dendrite [12]). The structure of a biological neural network in the brain is shown on Figure 3.

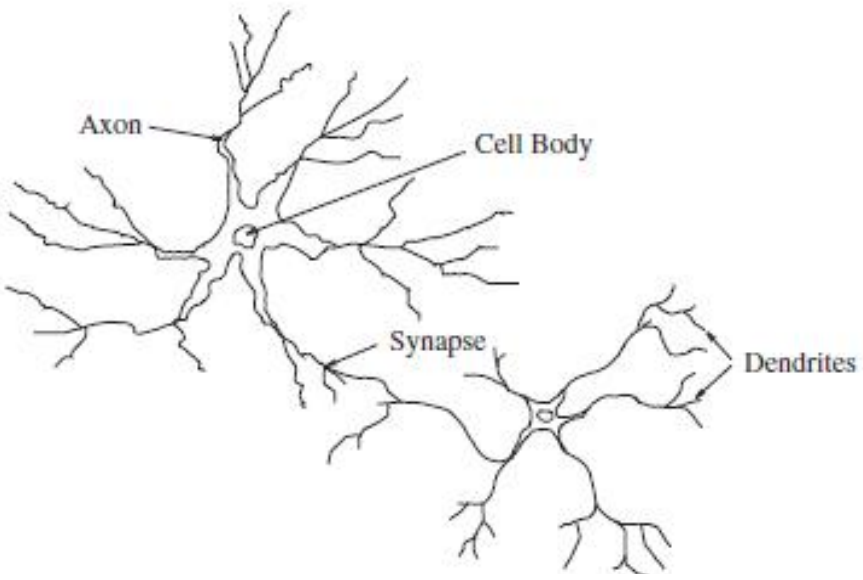

Figure 3. Structure of biological neural network [13].

The transmission of impulses from one neuron to another is a complex chemical process [11] in which signals are sent to other neurons along the axon and received through the dendrites, and as a result, certain chemicals are released [12]. Since the information is stored in the connection strengths between neurons in the brain [13], the synapses manage the work of the brain and are in charge of human memory [14].

2.1 Overview of artificial neural networks

Artificial neural networks (ANNs) can be created by simulating biological nervous systems on a computer [10]. ANNs consist of simple computational units called artificial neurons, or simply neurons. The synapses are represented by connection weights that adjust the effect of input signals [11].

2.1.1 History

In the history of artificial neural networks research, there have been three periods of considerable activity [12]. The first was in the 1940s after McCulloch and Pitts had introduced simplified artificial neurons [11], the second took place in the 1960s with Rosenblatt's perceptron convergence theorem, and Minsky and Papert's work showing the limitations of a simple perceptron. As a result of Minsky and Papert's work the enthusiasm of most researchers in the computer science community was diminished. In the 1980s, the interest in artificial neural networks began to rise again and artificial neural networks have become the subject of more and more attraction ever since [12].

2.1.2 Computational model

A neuron receives n signals $x_1, \ldots, x_n$ from other neurons through synapses:

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

Each input is multiplied with a weight coefficient (called synaptic weight) which can be either positive or negative. The synaptic weights can be denoted with a W:

$$W = [w_1 \quad \cdots \quad w_n].$$

The values of inputs $x_1, \ldots, x_n$ are multiplied with the corresponding weight coefficients to get the weighted sum [14]:

$$NET = W \cdot X = [w_1 \quad \cdots \quad w_n] \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = w_1 x_1 + \cdots + w_n x_n.$$

If the total value *NET* is above a threshold β, the neuron fires an impulse that is carried out to other neurons; and if the sum of incoming signals is below the threshold, the neuron stays inactive [13].

2.1.3 Perceptron

The first architecture of an ANN, called perceptron, was introduced in 1958 by Rosenblatt. Perceptron involves three types of neuron layers – input layer, hidden layer and output layer. The neurons (nodes) on the input layer allocate the input signals to the processing layers [13]. The first processing layer is the hidden layer (called hidden because of no external connections – they only receive input from other processing units and generate output to other processing units) which performs a significant role in the neural network since it captures the pattern in the input data and carries out a complex mapping between the input and output neurons [15]. The nodes on the second processing layer – the output layer – send out the received signals to the outer world. In a classic perceptron, only the connection strengths between the hidden nodes and the output nodes are modifiable; the connection strengths between the input nodes and the hidden nodes have to be preset before the training [13]. The neuron layers of a perceptron are illustrated on Figure 4.

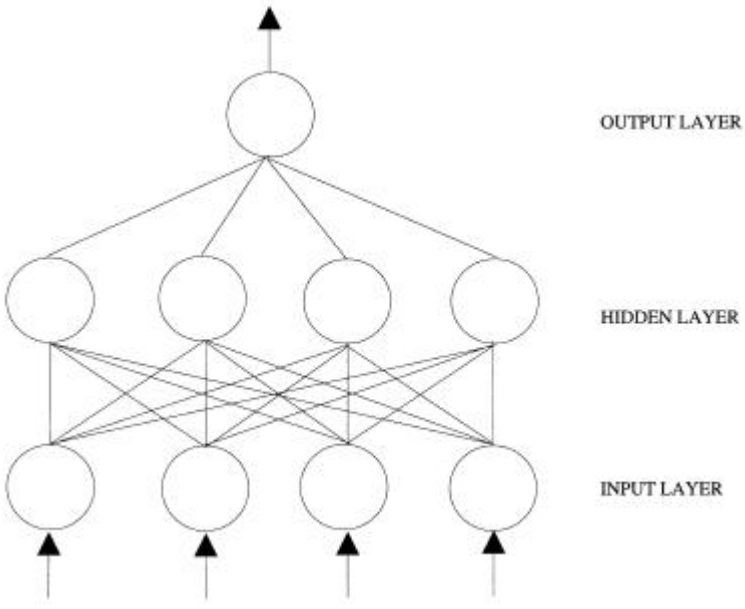

Figure 4. Three-layered perceptron structure [15].

At first, there was a keen interest in perceptrons, but soon problems began to arise – it became evident that with the growth of the scope of the task set complexity of the network grew exponentially. At first, the prospect was that real problems could be solved without maximal complexity, but Minsky and Papert demonstrated that the solutions to many basic problems require a full number of neurons. One solution seemed to be to add new layers of weights to the perceptron and train them, but no algorithm could be found that would allow that. That is why after the research of Minsky and Papert the attraction to ANNs abated for a couple of decades [13].

The reason for not finding a working training method for the three-layered perceptron came out to be using the hard threshold as an activation function of the neurons [13]. The activation function (also called transfer function) is a mathematical formula that takes the input signal of the neuron and calculates the output [9]. To obtain a training method that would work on the perceptron, a continuous function (for example, a sigmoidal function or a hyperbolic tangent function) should be used instead of the hard threshold [13]. The most common activation functions used in ANNs are

- sigmoid (logistic) function: $f(x) = \frac{1}{1+e^{-x}}$,
- hyperbolic tangent function: $f(x) = \tanh x$,
- sine or cosine function: $f(x) = \sin x$ or $f(x) = \cos x$ [15].

The most frequently used one of these is the sigmoid function, which is also used in the practical implementation of the current thesis (described in Chapter 3).

2.1.4 Architectures

Based on their architecture, ANNs can be classified into two categories:

- feed-forward networks;
- recurrent (feedback) networks [12].

In a feed-forward network, sending a signal from one neuron to another can only occur one way – in a feed-forward manner; this means no backward connections (loops) exist [11]. Feed-forward networks produce only one set of output values from a given input since their response to a new input pattern is independent of the previous state [12].

Contrary to feed-forward networks, in recurrent networks the signals can be sent either way between the neurons because of loops in the network [16]. When a new input pattern is introduced, the neuron outputs are computed and due to feedback connections the weights of each connection can be modified, leading the network to enter a new state [12].

2.1.5 Learning methods

One of the major advantages of ANNs over traditional systems is that instead of following rules specified by humans, they learn the rules automatically from representative examples – training patterns. The learning (also called training) process in ANNs means adjusting the connection weights of artificial neurons to achieve efficient results when performing a specific task [12]. Since the computation of a neuron varies depending on the weights, we can accomplish the desired output by adjusting the weights of the neuron. As the network may contain a large number of neurons, it is difficult to find the necessary weights by hand. To simplify the process, there are certain training algorithms for ANNs that can adjust the weights of the neurons to receive better results [17].

The learning paradigms in ANNs can be divided into two major categories: supervised learning and unsupervised learning [16]. The difference between the two methods is that supervised learning involves an external teacher that controls the learning process in the network by providing the desired responses for each output node, whereas unsupervised learning has no external teacher [11].

In supervised learning, an external teacher provides the input vector with training examples to the input layer together with a set of expected outcomes from the output layer. To regulate the connection weight changes in the ANN, the errors between the desired and the actual result of each node in the output layer are found [11]. Supervised learning methods usually work off-line which means that learning and operating are carried out separately, not at the same time like in on-line learning [9]. One of the most common training algorithms of supervised learning, also used in the practical implementation part of the present thesis, is back-propagation algorithm. An overview of this algorithm is given in Chapter 2.2 of the thesis.

Contrary to supervised learning, in unsupervised learning there is no previously known set of categories into which the patterns are supposed to be classified. Instead, the ANN is trained to respond to patterns within the input vector and has to develop its own representation of the input [11]. Unsupervised learning is also called self-organization since it self-organizes the provided data into similar classes. Examples of unsupervised learning algorithms are Hebbian learning and competitive learning [16].

Neural networks can be trained in two modes: online and offline (batch mode). Online learning means that the network learns and operates at the same time [9] – the weights of each input sample are calculated and modified after each sample [11]. In the batch training mode the learning phase and the operating phase are separated – the weight changes are calculated after each input sample, but they are accumulated until the end of one pass through the whole training set, called an epoch. After each epoch, the contributions of the nodes are added up and the weights are adjusted with the compound value [11].

2.1.6 Applications

Artificial neural networks can be used in various fields because of their adaptive nature of learning by examples and the ability to treat complicated problems with ease. In addition to modeling real neural networks, for example to study the behavior of animals [17], these characteristics make ANNs widely usable in the area of classification and prediction (pattern recognition, forecasting) where understanding of the problem to be solved is insufficient but the training data is available [9].

2.2 Error back-propagation

Back-propagation is the most widely used supervised learning algorithm in feed-forward multi-layer neural networks [16]. The algorithm was first introduced by Bryson and Ho in 1969 and independently rediscovered by Werbos in 1974, by Parker in the middle of the 1980s and by Rumelhart and Williams in 1985 [18]. The back-propagation algorithm had a substantial influence in the reappearance of neural networks in the middle of the 1980s after the decrease of interest in them in the 1960s [16].

The goal of training a back-propagation neural network is to retrieve a desired output when feeding a certain input to the network [17]. This is done by measuring the error between the acquired result and the desired result and reducing this error to a minimum – the smaller the error, the better the network (a perfect network would have an error of size zero) [10].

In back-propagation neural networks, the neurons are arranged into layers – the training data is received by the input layer and the output is delivered by the neurons on the output layer. The training vector of expected results on the output layer is provided by the external teacher. There can be one or more hidden layers [17] – the more hidden layers there are, the more complicated the network gets [16].

The first step in training the network is setting all connection weights in the ANN to small random numbers [10]. The example patterns are passed forward from the input layer to the output layer, producing an output pattern based on the random connection weights [16]. The next step is measuring the error – the difference between the received outcome and the expected outcome. In the back-propagation step these errors are passed back through the neural network and the connection weights are modified, based on the calculated contribution of each hidden node and determination of the necessary adjustments [16]. The adjustment of weights is repeated many times to successively reduce the error until it no longer changes [10]. As a result, the neural network has been trained to learn from examples.

3 Practical implementation of ANNs

The practical implementation of the current thesis includes creating ANNs, training and visualizing them and comparing the results. Predictions are made about the support coefficients based on the natural frequencies of the vibrating beams. An overview of the results received is provided in two sections – Chapter 3.1 shows the results of vibrating beams having elastic supports at the boundaries and Chapter 3.2 outlines the results of beams with intermediate elastic support.

The ANNs for the predictions are created in MATLAB programming environment. MATLAB is extensively used for solving technical computational problems. It can be used to perform numerical calculations, develop different algorithms, analyze and visualize data [19]. Extending MATLAB with Neural Network Toolbox adds functions and graphical tools for designing, training, simulating and visualizing artificial neural networks [20].

To evaluate and compare the effectiveness of the artificial neural networks created, the mean absolute error (*MAE*), the variance account for (*VAF*) and the coefficient of determination (R^2) are calculated.

The mean absolute error is a measurement of difference between the predicted values and the acquired values. The mean absolute error over all the patterns is expressed by the following equation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|n_{t_i} - n_{p_i}\right|, \tag{3.1}$$

where n is the number of patterns in the test set, n_{t_i} is the measured value and n_{p_i} is the predicted value [21].

The coefficient of determination is used to measure the reliability of the prediction of future outcomes, based on related examples. This value can be calculated using the equation

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(n_{t_i} - n_{p_i})^2}{\sum_{i=1}^{n}(n_{t_i} - n_m)^2}, \tag{3.2}$$

where $n_m$is the mean of target values n_t [22]:

$$n_m = \frac{1}{n}\sum_{i=1}^{n} n_{t_i}. \tag{3.3}$$

The variance account for is expressed with the following equation:

$$VAF = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(n_{t_i} - n_{p_i} - r)^2}{\frac{1}{n}\sum_{i=1}^{n}(n_{t_i} - n_m)^2}, \tag{3.4}$$

where $n_m$is the mean of target values (equation (3.3)) and r is the mean variance between the expected value and the predicted value:

$$r = \frac{1}{n}\sum_{i=1}^{n}(n_{t_i} - n_{p_i})^2. \tag{3.5}$$

The values of VAF and R^2 are between 0 and 1. In ideal situations, MAE would be equal to 0, and VAF and R^2 would be equal to 1.

All computations of the neural networks are performed on a computer with AMD Athlon™ II X4 640 Processor (3.00 GHz) and 8GB of installed RAM.

3.1 Cases of beams with elastic supports at the boundaries

The objective of this section is to provide an overview of the test results received by training the neural networks to predict the translational or rotational spring coefficients on one end of vibrating beams. The analysis includes the comparison of *MAE*, *VAF*, R^2 and training time based on the number of the natural frequencies (three, four, five, six, or nine). The number of neurons in the hidden layer of the neural network in all cases is equal to the number of patterns in the training set.

The total data available is allocated into two sets – one used for training the neural network and the other used for testing. To prevent the neural network from "knowing"

the answers rather than “learning” them, the two sets do not have an intersection. The training set consists of 500 patterns and the test set consists of 50 patterns.

The cases of boundary conditions of the beams studied are:

1. clamped at left end and sliding with translational spring at right end,
2. clamped at left end and free with translational spring at right end,
3. clamped at left end and with a translational and a rotational spring at right end,
4. simply supported at left end and sliding with translational spring at right end,
5. simply supported at left end and free with translational spring at right end,
6. simply supported at left end and with a translational and a rotational spring at right end, and
7. translational and rotational springs at both ends.

Each of these cases is analyzed in detail in the following subsections.

The stiffness coefficients of the left end are fixed in all cases and the coefficients of the right end are varied. For a beam with translational and rotational springs at both ends the spring coefficients of left and right end are symmetric – the rotational coefficient of both ends is fixed and the predictions are made about the varying translational coefficient parameter.

3.1.1 Clamped – sliding with translational spring

For a beam clamped at left end and sliding with translational spring at right end, the accuracy parameters of prediction of the rotational right end support condition coefficient are displayed in Table 2. The parameters are pointed out for each number of frequencies.

Table 2. Efficiency results of beam clamped at left end and sliding with translational spring at right end.

frequencies	R^2	VAF	$\frac{\text{MAE}}{\max(\text{expected})}$ (%)	Training time (s)
3	0,999825	0,916434	0,103546	8,676860
4	0,999981	0,998978	0,076899	10,082575
5	0,999993	0,999858	0,050281	11,818164
6	0,999995	0,999941	0,049185	11,743470
9	0,999999	0,999997	0,020143	44,427214

The results show that the efficiency is the best when using four, five, six or nine frequencies as the input of the neural network and slightly lower when using three frequencies. This indicated that adding more frequencies to the input does not noticeably improve the quality of the predictions.

3.1.2 Clamped – free with translational spring

Table 3 presents the accuracy parameters of the test results of a beam clamped at left end and free with translational spring at right end.

Table 3. Accuracy of predictions for beam clamped at left end and free with translational spring at right end.

frequencies	R^2	VAF	$\frac{\text{MAE}}{\max(\text{expected})}$ (%)	Training time (s)
3	0,994821	–	0,360189	18,862208
4	0,999652	0,707948	0,003869	20,038095
5	0,999987	0,999576	0,035884	16,045238
6	0,999944	0,992450	0,101045	14,899730
9	0,999980	0,999050	0,094972	20,194107

The *VAF* results are close to perfect when using five, six or nine frequencies ($VAF > 0.99$). In case of four frequencies the result is 0.7 and in case of three frequencies the value of

VAF did not fit in the range [0, 1] and is therefore discarded from the table. The unqualified *VAF* values means that there are some patterns in the training or test sets which could be considered as "noise". When a training set contains too distinct or too similar values, the neural network may rather "remember" them instead of "learning", and is therefore unable to produce reasonable results on unseen data [23]. The results of training the neural network illustrated on Figure 5 indicate that there is a pattern in the training set that the neural network is incapable of learning.

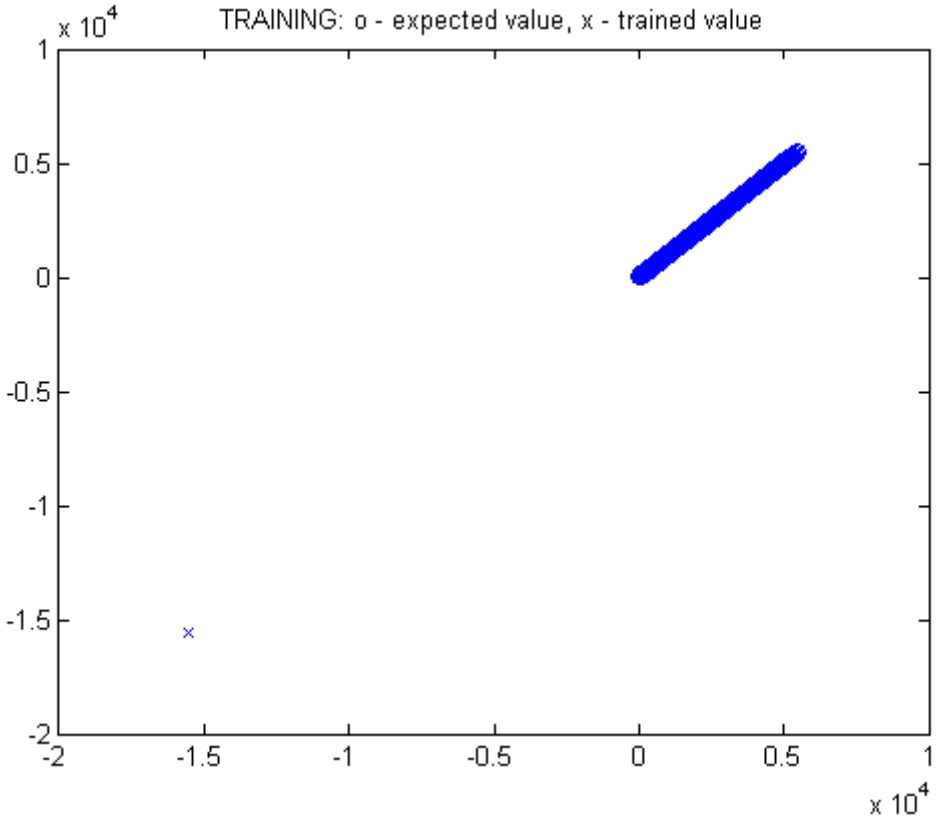

Figure 5. Training results of a beam clamped at left end and free with translational spring at right end.

Due to the failure of learning some pattern in the training set, the test results are also distorted.

3.1.3 Clamped – translational and rotational spring

The parameters of test results of a beam clamped at left end and translationally and rotationally supported at right end are demonstrated in Table 4.

Table 4. Prediction of the support parameter at right end of a beam clamped at left end and translationally and rotationally restrained at right end.

frequencies	R^2	VAF	$\frac{\text{MAE}}{\max(\text{expected})}$ (%)	Training time (s)
3	0,998256	0,997710	0,446440	9,566218
4	0,999984	0,999984	0,071364	12,114586
5	0,999707	0,999693	0,117893	18,538891
6	0,998838	0,998388	0,265321	11,157478
9	0,999998	0,999997	0,035793	46,864194

The values of R^2 and *VAF* have a similar trend in the sense of the number of frequencies. The similarity of the trend of the results is illustrated on Figure 6.

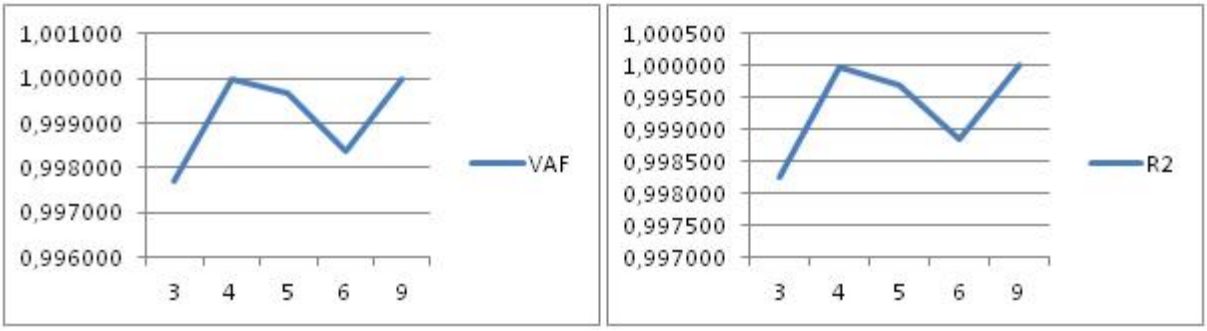

Figure 6. Comparison of the trends of *VAF* and R^2 in case of a beam clamped at left end and translationally and rotationally supported at right end.

As can be seen from Table 4 and Figure 6, the accuracies of *VAF* and R^2 are the highest when using four or nine frequencies, and the lowest with three frequencies.

3.1.4 Simply supported – sliding with translational spring

The accuracy parameters for a beam simply supported at left end and sliding with a translational spring at right end are outlined in Table 5.

Table 5. Prediction of the support parameter at right end of a beam simply supported at left end and sliding with translational spring at right end.

frequencies	R^2	VAF	$\frac{\mathbf{MAE}}{\mathbf{max(expected)}}$ (%)	Training time (s)
3	0,970940	–	0,748432	9,341914
4	0,999943	0,991377	0,110054	9,841985
5	0,999984	0,999304	0,099334	14,758124
6	0,999997	0,999977	0,032079	18,002236
9	0,999997	0,999968	0,046501	26,690434

The results are quite similar and stable when using four, five, six or nine frequencies. Only when using three frequencies, the outcomes are not as satisfactory – the variance account for does not fit in the expected region, and the ratio of mean absolute error and the maximum expected result is 0.75% (as opposed to the values of 0.03 to 0.11 in the other cases).

3.1.5 Simply supported – free with translational spring

Table 6 indicates the values of prediction of the right end coefficient in case of a beam simply supported at left end and free with translational spring at right end.

Table 6. Prediction of the support parameter at right end of a beam simply supported at left end and free with translational spring at right end.

frequencies	R^2	VAF	$\frac{\mathbf{MAE}}{\mathbf{max(expected)}}$ (%)	Training time (s)
3	0,656743	–	2,576485	42,847447
4	0,976297	–	0,881612	19,646157
5	0,766862	–	2,078920	12,000257
6	0,930134	–	1,174855	11,678558
9	0,998340	–	0,263396	19,624027

In all cases, the value of *VAF* is left out of the table since it was inadequate in the sense of the efficiency comparison. The anomaly of the values of *VAF* could become evident when there are some patterns in either training set that make the prediction of the neural network propose random values of the outcome on the test data. Figures 7 and 8 respectively show the training and test results of the beam simply supported at left end and free with translational spring at right end when using three input frequencies.

Figure 7. Training results of a beam simply supported at left end and free with translational spring at right end using three frequencies.

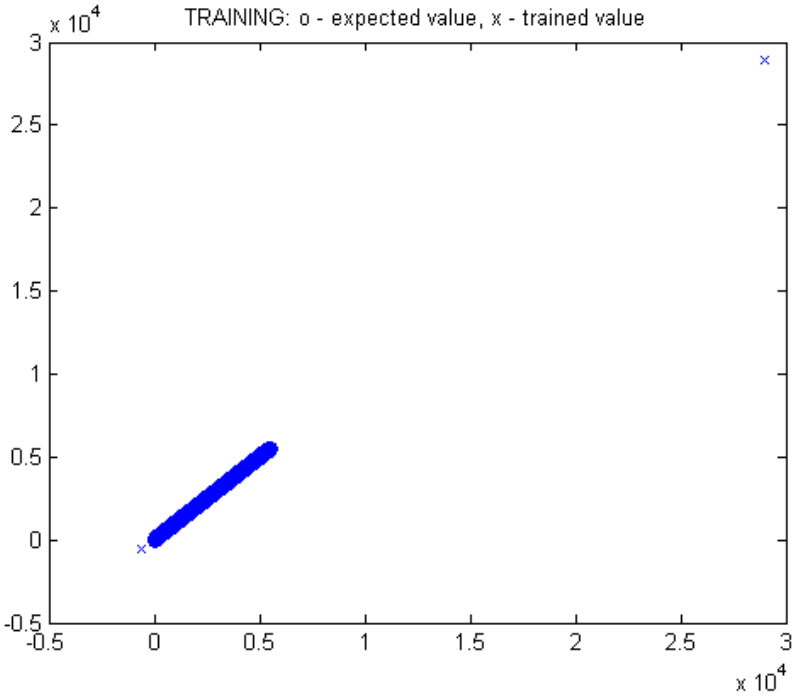

Figure 7 indicates that there exists a pattern in the training data that the neural network is not capable of learning. Thus, some value from the test set is not predicted correctly either, as can be seen on Figure 8.

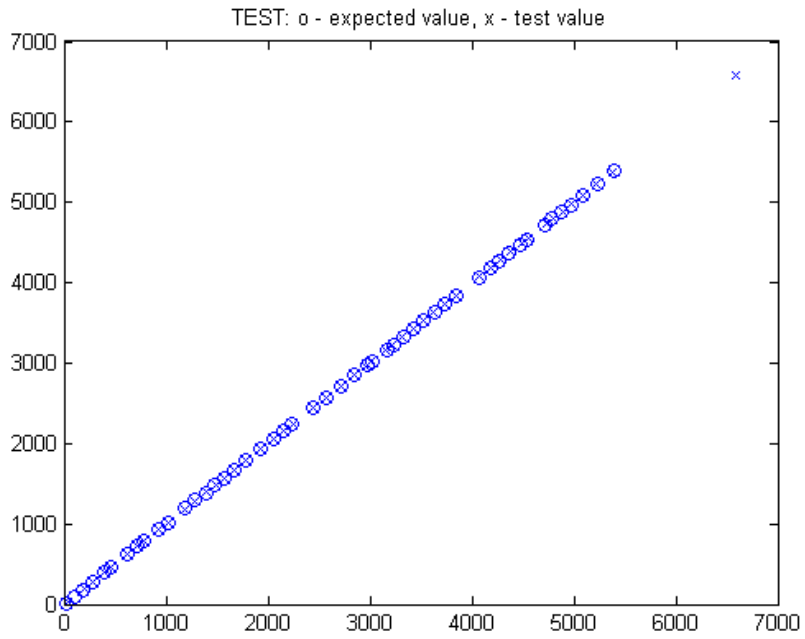

Figure 8. Test results of a beam simply supported at left end and free with translational spring at right end using three frequencies.

3.1.6 Simply supported – translational and rotational spring

The prediction efficiency parameters of a beam that is simply supported at left end and has a translational and rotational spring at right end are revealed in Table 7.

Table 7. Prediction of the support parameter at right end of a beam simply supported at left end and translationally and rotationally restrained at right end.

frequencies	R^2	VAF	$\frac{\text{MAE}}{\max(\text{expected})}(\%)$	Training time (s)
3	0,999654	0,999626	0,17734	10,108303
4	0,997630	0,997297	1,218785	13,645111
5	0,999064	0,998755	0,194191	15,642321
6	0,999971	0,999968	0,121235	14,310933
9	0,999981	0,999981	0,099666	25,621451

The shape of the trend of the values of *VAF* and R^2 over different numbers of frequency is similar, analogously to the beam described in Chapter 3.1.3 (clamped – translational and rotational spring). This can be seen on Figure 5.

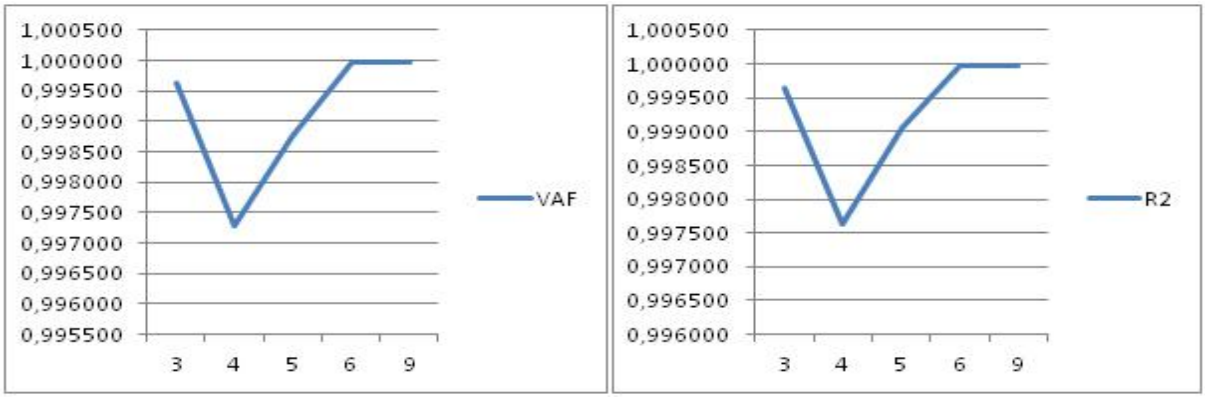

Figure 9. Comparison the trends of *VAF* and R^2 in case of a simply supported at left end and translationally and rotationally supported at right end.

In this case, the worst results are received when using four frequencies, and the best results are received when using six or nine frequencies.

3.1.7 Translational and rotational spring – translational and rotational spring

For a beam with translational and rotational restraints at both ends, the parameters of left and right end are symmetric. The predictions are made about the translational parameter coefficient and the accuracy measurements are displayed in Table 8.

Table 8. Prediction of the translational parameter at right end of a beam translationally and rotationally restrained at both ends.

frequencies	R^2	VAF	$\frac{\mathbf{MAE}}{\mathbf{max(expected)}}$ (%)	Training time (s)
3	0,929910	–	1,227524	32,497530
4	0,977716	–	0,921568	13,662677
5	0,998894	–	0,251842	14,227158
6	0,999981	0,999040	0,054800	12,162234
9	0,999991	0,999752	0,073718	19,569463

The values of *VAF* are discarded from the comparison table in cases of three, four and five frequencies. The *VAF* values for six and nine frequencies on the other hand are significantly close to perfect (*VAF* > 0.999).

The ratio of *MAE* and the maximum expected result is around 1% in case of three and four frequencies. With six and nine frequencies it is quite close to 0 (0.055 and 0.074 respectively).

3.1.8 Conclusions

Chapters 3.1.1 to 3.1.7 gave an overview of the test results of predicting either the translational or rotational spring coefficients on one end of the examined cases of vibrating beams. The accuracy parameters were compared among different numbers of input frequencies for each case.

The overall results were rather pleasant in most of the cases, but the most accurate predictions based on the comparison of *VAF* and R^2 were made in the cases of beams clamped or simply supported at left end and translationally and rotationally supported at right end.

Figures 10 and 11 show the comparison of *VAF* and R^2 values for each case of a beam elastically restrained at the boundary conditions. The cases marked with *1-5*, *1-7*, *1-8*, *2-5*, *2-7*, *2-8* and *8-8* are respectively the cases of beams analyzed in chapters 3.1.1 to 3.2.7.

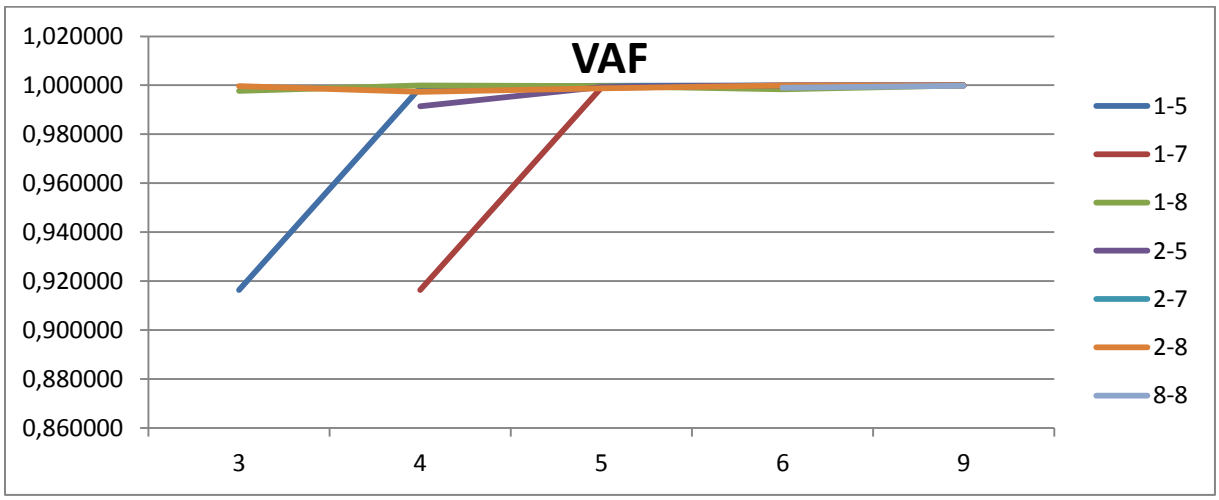

Figure 10. Comparison of *VAF* results among beams with different support conditions.

Figure 10 indicates that the gained *VAF* values that fit in the desired interval of [0, 1] are relatively uniform in most of the cases – only the cases of beams clamped or simply supported at left end and translationally and rotationally supported at right end show some deviation from the rest of the results.

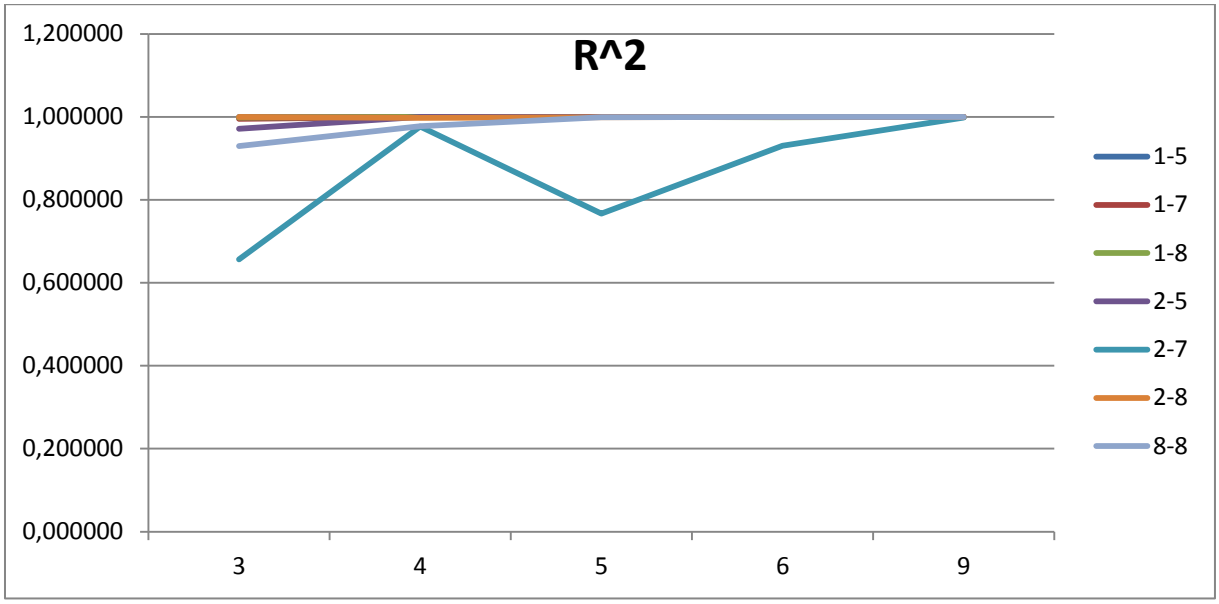

Figure 11. Comparison of R^2 results among beams with different support conditions.

On Figure 11, it can be seen that the results of R^2 are rather uniform and close to 1 in most cases. The only exception is in case of a beam simply supported at left end and with a translational and rotational spring at right end – for some frequencies, the value of R^2 drops down to between around 0.65 and 0.75.

Based on the analysis of the gained test results, the usage of neural networks for predicting the elastic support coefficients on one end of a beam with elastic supports at the boundaries is fairly justified.

3.2 Cases of beams with intermediate elastic support

Current chapter presents the test results of training neural networks to predict the rotational spring coefficients at the intermediate point of a beam with intermediate elastic support. Similarly to the previous section, the accuracy parameters under study are *VAF*, R^2, the ratio of *MAE* and the maximal expected result, and neural network

training time. Four different cases of input frequencies are investigated – three, four, five and six.

The input data is divided into two portions – size of the training set is 110 patterns and size of the test set is 15 patterns. The amount of data is smaller than in case of beams elastically restrained at the boundaries, since the calculation of the input data for a beam with intermediate support is significantly more time consuming.

The beams under investigation have the following support conditions:

1. clamped at left end, translational and rotational spring at intermediate support, clamped at right end;
2. clamped at left end, translational and rotational spring at intermediate support, free at right end;
3. simply supported at left end, translational and rotational spring at intermediate support, simply supported at right end;
4. sliding at left end, a translational and a rotational spring at intermediate support and at the right end.

The following four subsections give an overview of each of these cases at length.

The placement of the intermediate support can vary along the length of the beam. In all the studied cases the intermediate support is such that the left part of the beam forms 0.2 of the length of the beam and the right part is 0.8 of it. In the case of the beam clamped at both ends (characterized in Chapter 3.2.1) another example is performed where the placement of the intermediate support is such that the left part of the beam is 0.6 of the length of the beam.

For each case, the value of the rotational stiffness is varied by 10 at each step (starting from the value 10); whereas the translational parameter stiffness of the intermediate support and the stiffness coefficients of both end all have a fixed value of 10.

3.2.1 Clamped – translational and rotational spring – clamped

The vibrations of a beam clamped at both ends and translationally and rotationally restrained at the intermediate support are discussed in two parts. In the first example

the intermediate support is such that the left part of the beam forms 0.2 of the length of the beam, and in the second example, the according length is 0.6. The results of the accuracy criterions are compared on the terms of the lengths of the left and right part of the beam, and the number of input frequencies.

The calculated data includes some patterns where the frequencies are substantially different from the rest. These patterns are retained in the training set, but discarded from the test set. The results of the prediction efficiency measurements are displayed in Table 9.

Table 9. Prediction accuracy measurements for a beam clamped at both ends and with translational and rotational spring support along the span. Length of the left part of the beam is 0.2 of the beam length.

frequencies	R^2	VAF	$\frac{\mathbf{MAE}}{\mathbf{max(expected)}}(\%)$	Training time (s)
3	0,899566	–	7,424009	5,011460
4	0,977833	–	3,497116	28,664304
5	0,921434	–	6,813843	32,815058
6	0,831708	–	9,765779	10,080591

The results of *VAF* are in all cases out of the boundaries of the region of the expected value; hence they are marked with the "–" sign. The values of the ratio of *MAE* and the maximum expected result are not very outstanding – varying from 3.5% to 9.8%. On the other hand, the values of the multiple coefficient of determination are considerably high – varying from 0.83 to 0.98. Better results are received when using four of five input frequencies; however, in these cases the training of the neural network takes significantly more time (on average, around 30 seconds), as opposed to 5 to 10 seconds in the other cases.

In Table 10, the results of the prediction of the intermediate rotational spring coefficients for a beam clamped at both ends and the length of the left part of the beam being 0.6 of the length of the beam are depicted.

Table 10. Prediction accuracy measurements for a beam clamped at both ends and with translational and rotational spring support along the span. Length of the left part of the beam is 0.6 of the beam length.

frequencies	R^2	VAF	$\frac{\mathbf{MAE}}{\mathbf{max(expected)}}$ (%)	Training time (s)
3	0,999630	0,980841	0,457090	5,290129
4	0,999908	0,998835	0,235176%	16,517292
5	0,997454	0,083723	1,316335	7,293741
6	0,999891	0,998221	0,252580	10,970367

From Table 10 it can be seen that the predictions in this case are much better than in case of the beam where the length of the left part is 0.2. The values of variance account for all fit in the desired range of [0, 1], whereas when using three, four or six frequencies the *VAF* is actually quite high – over 0.98. The only exception is when using five frequencies (*VAF* = 0.084). Also the values of R^2 are incredibly high – in most cases above 0.999 and in one case 0.997.

The results displayed in tables 9 and 10 indicate that the position of the intermediate support plays a quite substantial role in the parameter coefficient identification – when the placement of the intermediate support is close to the boundary, the accuracy of the prediction is lower than with the intermediate support placed closer to the middle point of the beam.

3.2.2 Clamped – translational and rotational spring – free

The results of the measured efficiency parameters of a cantilever beam with an intermediate support are displayed in Table 11.

Table 11. Comparison of efficiency parameters of a cantilever beam with translational and rotational intermediate support.

frequencies	R^2	VAF	$\frac{\mathbf{MAE}}{\mathbf{max(expected)}}$ (%)	Training time (s)
3	–	–	1452,780136	10,795572
4	–	–	1458,280199	14,416095
5	–	–	501,611082	21,008090
6	0,931531	–	7,221825	8,267138

When using three, four or five frequencies, the values of *VAF* and R^2 were extensively out of the boundaries of the interval [0, 1] where they should be, and in these cases the results are marked with the "–" sign. The ratio of *MAE* and the maximum expected outcome was enormous in these cases (1453%, 1458% and 502% respectively).

The only case that produced meaningful results was when using nine frequencies as the input. The R^2 in this case is 0.93 and the ratio of *MAE* and the maximum expected result is 7.2%.

The reason behind the anomaly of the accuracy parameters of the cantilever beam with intermediate elastic support is that the training and test sets contain some patterns which are significantly different from other patterns. This causes overtraining – in some cases if the training set contains errors or very distinct values, or on the contrary, has very similar training patterns, the network may adapt to this noise and the capability of generalization may decrease, therefore producing random output for unseen inputs [23].

In the case of a beam clamped at both ends (described in Chapter 3.2.1) the test set was compiled so that it would not contain "alien" patterns, but in case of the cantilever beam, the test patterns were chosen randomly by not paying attention to the contrast of the patterns. The efficiency of the prediction could be improved by removing the strongly distinct patterns from the test set, and by also removing them from the training set the results improve even more significantly.

3.2.3 Simply supported – translational and rotational spring – simply supported

The results of the efficiency of training the neural networks to predict the rotational intermediate support coefficient of a beam simply supported at both ends and with translational and rotational springs as the intermediate support are presented in Table 12.

Table 12. Efficiency parameters of a beam simply supported at both ends, with translational and rotational spring at intermediate support.

frequencies	R^2	VAF	$\frac{\mathbf{MAE}}{\mathbf{max(expected)}}$ (%)	Training time (s)
3	0,988805	–	2,137762	7,689307
4	0,991339	–	2,233350	7,821255
5	0,981170	–	3,278713	5,351338
6	0,993917	–	1,997735	5,595005

Similarly to the previous case, since the training and test sets contain some patterns with frequencies that are distinct from the others, the results of *VAF* are not acceptable in the analysis, which is why the *VAF* results are discarded from the table.

The values of the ratio of *MAE* and the maximum expected result are not remarkably wonderful, varying from 2% to 3.3%. On the other hand, the values of R^2 are sufficiently high (0.98 to 0.99), and the training times of the neural networks are low quite stable (around 5 seconds to 8 seconds).

Analogously to the previous case, the results of the predictions of the beam simply supported at both ends and with intermediate elastic support could be improved by removing the patterns out of line with the other patterns from either the training set of the test set.

3.2.4 Sliding – translational and rotational spring – translational and rotational spring

Table 13 outlines the efficiency results of the prediction of the intermediate rotational parameter coefficient in case of a beam which is sliding at left end and has a translational and rotational spring at right end and as an intermediate support.

Table 13. Accuracy of prediction of rotational intermediate coefficient for a beam sliding at left end, translationally and rotationally supported along the span and at the right end.

frequencies	R^2	VAF	$\frac{\mathbf{MAE}}{\mathbf{max(expected)}}$ (%)	Training time (s)
3	–	–	9,379086	4,956197
4	0,906975	–	3,173953	5,754903
5	0,989193	–	0,962713	24,194533
6	0,877611	–	4,572388	7,033587

Once again, the values of *VAF* did not fit in the expected range of accuracy and they are left out from the table. When using three frequencies as the input for training the neural network, the efficiency of the prediction is very low – the ratio of MAE and the maximum expected result is 9.4% which means that the identified results differ from the expected results quite heavily.

The best results in the context of the present case are received when using five frequencies as an input – R^2 is almost 0.99 and the ratio of mean absolute error and maximum expected result is below 1%.

The results could be boosted the same way as for the previous two cases – by reviewing the test and training sets and removing patterns that contain "noise" in the sense of training the neural networks.

3.2.5 Conclusions

Chapters 3.2.1 to 3.2.4 provided an analysis of the test results of the prediction of the coefficient of the rotational spring parameter in cases of the vibrating Euler-Bernoulli beams with intermediate elastic support. The efficiency of the predictions was measured for four cases of beams with intermediate elastic support. The results were compared among different number of input frequencies for the artificial neural networks.

The analysis of the prediction efficiency shows that the results depend heavily on the input data of training the neural network. If the training set contains noise (some patterns that are substantially distinct from the others), it can cause overtraining of the

artificial neural network and become unable to make reasonable predictions on the test set. To avoid this behavior, the training and test sets of the cases should be reviewed and modified where needed. These results show that in some cases the natural frequencies do not contain sufficient information for the identification of the parameters of non-classical boundary conditions. This fact has also been noticed by other authors in the case of structural health detection [24]. Therefore, some additional data, e.g. mode shapes [25] could be applied to extract the important features for the parameter identification in vibrating systems, but this is out of the scope of the present thesis.

Conclusion

In the present thesis, an overview of the Euler-Bernoulli beam theory and the basics of artificial neural networks were presented. The main emphasis was on the practical implementation of training the artificial neural networks for predicting the stiffness parameters of the support conditions of the vibrating beams.

The main purpose of the current paper was to study the frequencies of vibrating Euler-Bernoulli beams with different non-classical support conditions, and to analyze the efficiency of predicting the support condition coefficients (either translational or rotational). The calculated natural frequencies of the vibrating beams were used as the input for training the neural networks. The results were computed for various cases, using different numbers of input frequencies (three, four, five, six, or nine) besides the different support conditions.

The results of the predictions were analyzed in two different parts: the efficiency of prediction in case of beams with elastic support at the boundaries, and the efficiency of prediction in case of beams with intermediate elastic support.

The analysis of the efficiency of prediction in case of beams with elastic support at the boundaries showed that the overall efficiency of the predictions was substantially high and the identified results were quite similar to the expected outcomes. The best average results among all conditions were received with the beam clamped or simply supported at left end and translationally and rotationally restrained at right end. But even in the worst cases, most of the results were considerably nice.

The analysis of the efficiency of predicting the rotational coefficient at the intermediate support in case of beams with intermediate elastic support showed that the results greatly depend on the generation of the training and test sets. If the training data contains noise, then the efficiency of the prediction is rather low, but it could be improved by modifying the training and test data sets. Also, alternative methods should be elaborated to extract features for parameter identification of vibrating systems.

Mitteklassikaliste kinnitustingimuste tuvastamine tehisnärvivõrkude abil

Magistritöö (30 EAP)

Mairit Vikat

Resümee

Käesolev magistritöö uurib mitteklassikaliste kinnitustingimustega elastsete Euler-Bernoulli talade vabavõnkumise resonantssagedusi. Eesmärgiks on vaatluse all olevate tala mudelite korral hinnata ning võrrelda tehisnärvivõrkude abil identifitseeritud jäikuse parameetreid elastsete kinnitustingimuste korral.

Vaatluse all on kahte tüüpi talad: tala elastse otsakinnitusega ning tala vahepealse elastse toega. Mõlema variandi kohta töötatakse läbi rida näiteid erinevate kinnitustingimustega.

Kuna kinnituste jäikusparameetrite arvutamine võnkumise diferentsiaalvõrrandist ei ole analüütiliselt võimalik, siis on mõistlik otsida sellele alternatiivi. Ühe variandina pakutakse käesolevas töös välja tehisnärvivõrkude rakendamine.

Tehisnärvivõrgud põhinevad bioloogilistel närvivõrkudel, nagu näiteks inimese aju. Tehisnärvivõrgu peamiseks eeliseks teiste meetodite ees on tema võime olemasolevate näidete põhjal õppida, mis tähendab, et närvivõrke on võimalik treenida sisendi abil soovitud tulemusi produtseerima. Seega, vajaliku ülesande lahendamiseks pole enam tarvis ise kõiki parameetrite koefitsiente arvutada, vaid piisab, kui meil on olemas teatud hulk näiteid oodatavate koefitsientide kohta, ning nende näidete abil treenitud tehisnärvivõrk on suuteline ülejäänud tulemusi ise identifitseerima.

Käesolevas töös antakse ülevaade võnkuvatest Euler-Bernoulli taladest ja nende võimalikest kinnitustingimustest, ning tutvustatakse tehisnärvivõrkude peamisi omadusi. Töö peamine rõhk on asetatud praktilisele osale, kus uuritakse kahte tüüpi elastseid talasid (elastsete otsakinnitustega ja elastse vahekinnitusega) ning

analüüsitakse tehisnärvivõrkude abil saavutatud ennustuste tulemusi erinevatel juhtudel.

Lisaks erinevatele kinnitustingimustele võrreldakse tulemusi erineva sisendsageduste arvu (kolm, neli, viis, kuus või üheksa sagedust) korral. Saadud tulemusi analüüsitakse ja võrreldakse teatud täpsusparameetrite põhjal.

Läbiviidud arvutuste ning analüüsi põhjal selgub, et enamikel juhtudel on ennustuse teel saavutatud tulemused üpris ligilähedased oodatavatele tulemustele, seega on võnkuvate Euler-Bernoulli talade kinnitustingimuste jäikusparameetrite ennustamisel närvivõrkude rakendamine mõistlik.

References

[1] M. Vable, "Symmetric Bending of Beams," in *Mechanics of Materials*, 2nd ed., 2012, p. 254.

[2] "Beam structure," [Online]. Available: http://www.absoluteastronomy.com/topics/Beam_%28structure%29. [Accessed 6 April 2012].

[3] M. A. De Rosa, P. M. Belles, M. J. Maurizi, "Free Vibrations of Stepped Beams with Intermediate Elastic Supports," *Journal of Sound and Vibration,* vol. 181(5), pp. 905-910, 1995.

[4] "Euler-Bernoulli beam equation," [Online]. Available: http://www.absoluteastronomy.com/topics/Euler-Bernoulli_beam_equation. [Accessed 6 April 2012].

[5] S. M. Han, H. Benaroya, T. Wei, "Dynamics of Transversely Vibrating Beams Using Four Engineering Theories," *Journal of Sound and Vibration,* vol. 225(5), pp. 935-988, 1999.

[6] K. Hyeong Koo, K. Moon Saeng, "An Analytical Method for Calculating Vibration Characteristics of PWR Fuel Assembly with Reactor End Boundary Conditions Using Fourier Series," 2001.

[7] C. M. Albarracin, L. Zannier, R. O. Grossi, "Some observations in the dynamics of beams with intermediate supports," *Journal of Sound and Vibration,* vol. 271, pp. 475-480, 2004.

[8] J. H. Lau, "Vibration Frequencies and Mode Shapes for a Constrained Cantilever," *Journal of Applied Mechanics,* vol. 51, pp. 182-187, 1984.

[9] G. K. Jha, "Artificial Neural Networks," [Online]. Available: http://www.iasri.res.in/ebook/EB_SMAR/e-book_pdf%20files/Manual%20IV/3-

ANN.pdf. [Accessed 10 May 2012].

[10] A. Krogh, "What are artificial neural networks?," *Nature Biotechnology,* vol. 26, no. 2, pp. 195-197, 2008.

[11] A. Abraham, "Artificial Neural Networks," in *Handbook of Measuring System Design*, John Wiley & Sons, Ltd, 2005, pp. 902-908.

[12] A. K. Jain, J. Mao, K. M. Mohiuddin, "Artificial Neural Networks: A Tutorial," *Computer,* vol. 29, no. 3, pp. 31-44, 1996.

[13] K. Worden, W. J. Staszewski, J. J. Hensman, "Natural computing for mechanical systems research: A tutorial overview," *Mechanical Systems and Signal Processing,* vol. 25, no. 1, pp. 4-111, 2011.

[14] E. Petlenkov, "Tehisnärvivõrgud ja nende rakendused," 2004. [Online]. Available: http://www.dcc.ttu.ee/LAS/ISS0010/Tehisnarvivorgud-Eduard2004.pdf. [Accessed 10 May 2012].

[15] G. Zhang, B. E. Patuwo, M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting,* no. 14, pp. 35-62, 1998.

[16] C. Gupta, "Implementation of Back Propagation Algorithm (of neural networks) in VHDL," 2006. [Online]. Available: http://dspace.thapar.edu:8080/dspace/bitstream/123456789/270/1/r8044. [Accessed 10 May 2012].

[17] C. Gershenson, "Artificial Neural Networks for Beginners," [Online]. Available: http://arxiv.org/ftp/cs/papers/0308/0308031.pdf. [Accessed 10 May 2012].

[18] R. Hecht-Nielsen, "Theory of the Backpropagation Neural Network," [Online]. Available: http://s112088960.onlinehome.us/annProjects/Research%20Paper%20Library/backPropTheory.pdf. [Accessed 10 May 2012].

[19] "MathWorks Nordic - MATLAB - The Language of Technical Computing," MathWorks, [Online]. Available: http://www.mathworks.se/products/matlab/. [Accessed 29 April 2012].

[20] "MathWorks Nordic - Neural Network Toolbox - MATLAB," [Online]. Available: http://www.mathworks.se/products/datasheets/pdf/neural-network-toolbox.pdf. [Accessed 29 April 2012].

[21] "Mean absolute error - Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/Mean_absolute_error. [Accessed 28 April 2012].

[22] "Coefficient of determination - Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/Coefficient_of_determination. [Accessed 28 April 2012].

[23] R. P. W. Duin, "Learned from Neural Networks," 2000. [Online]. Available: http://homepage.tudelft.nl/a9p19/papers/asci_00_NNReview.pdf. [Accessed 11 May 2012].

[24] A. Deraemaeker, E. Reynders, G. De Roeck, J. Kullaa, "Vibration-based structural health monitoring using output-only measurements under changing environment," *Mechanical Systems and Signal Processing,* vol. 22, no. 1, p. 34–56, 2008.

[25] L. Feklistova, "Back Propagation Neural Network as a Prediction Tool for Vibrating Systems," Tartu, 2008.

Appendices

Appendix 1 – CD: Inputs and outputs of neural networks

The thesis includes a CD as an appendix. The CD contains two folders, called "supports-at-boundaries" and "intermediate-supports".

The folder "supports-at-boundaries" has seven subfolders:

1. "1-5" (clamped at left end, sliding with translational spring at right end);
2. "1-7" (clamped at left end, free with translational spring at right end);
3. "1-8" (clamped at left end, translational and rotational spring at right end);
4. "2-5" (simply supported at left end, sliding with translational spring at right end);
5. "2-7" (simply supported at left end, free with translational spring at right end);
6. "2-8" (simply supported at left end, translational and rotational spring at right end);
7. "8-8" (translational and rotational spring at both ends).

The folder "intermediate-supports" consists of five subfolders:

1. "1-8-1(0.2)" – beam clamped at both ends and with a translational and rotational intermediate support (the length of the left part of the beam is 0.2 of the whole length);
2. "1-8-1(0.6)" – beam clamped at both ends and with a translational and rotational intermediate support (the length of the left part of the beam is 0.6 of the whole length);
3. "1-8-3" – beam clamped at left end, free at right end, and translationally and rotationally restrained at intermediate support;
4. "2-8-2" – beam simply supported at both ends and with a translational and rotational intermediate support;
5. "4-8-8" – beam sliding at left end and translationally and rotationally restrained at intermediate support and right end.

Each of the above folders contains subfolders for each number of frequencies and a Microsoft Excel file that holds the information of the accuracy parameters.

Inside the folder of each number of frequencies there are two input files: training and test sets; two output files: expected and identified results; and two images: an image of the test results and an image of the training results.