Available online at www.eccomasproceedia.org
Eccomas Proceedia COMPDYN (2021) 3894-3911

ECCOMAS
Proceedia

COMPDYN 2021
8th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis (eds.)
Streamed from Athens, Greece, 28 - 30 June 2021

PREDICTION OF THE EIGENPERIODS OF MDOF SHEAR BUILDINGS USING NEURAL NETWORKS

Vagelis Plevris[1] **and German Solorzano**[2]

[1] Department of Civil and Architectural Engineering, Qatar University
P.O. Box: 2713, Doha, Qatar
e-mail: vplevris@qu.edu.qa

[2] Department of Civil Engineering and Energy Technology, OsloMet–Oslo Metropolitan University
Pilestredet 35, Oslo 0166, Norway
e-mail: germanso@oslomet.no

Abstract

The study of multi-degree of freedom (MDOF) systems is essential to evaluate and understand the seismic response of buildings. Through a MDOF idealization, the dynamic properties of the building such as its natural frequencies and modal shapes can be approximated. These properties are then used to determine the final design of the structural system of the building. A shear building MDOF system consists of an idealized model of the building in which the masses are concentrated at the floor levels and each floor is connected to other adjacent floors with elements that provide stiffness and only allow horizontal displacements. The dynamic properties of the idealized system are obtained by numerically solving a generalized eigenvalue problem which is a computationally expensive operation. In this paper, we propose a methodology to replace the required solution of the generalized eigenvalue problem with a Machine Learning NN-based approach. Two shear building models with 3 and 5 stories are considered, where the mass and the stiffness are held constant for every story. For every model, a database with the solution of several idealized models with varying mass and stiffness is created using a small number of samples (m, k pairs). Finally, an Artificial Neural Network is trained with the database to predict the eigenperiods of other similar models avoiding the computation of the eigenvalue problem. The results show a high level of accuracy in the predictions and a significant reduction of the computational time compared to the hard-computing mathematical approach. Furthermore, the approach demonstrated in this study can be easily expanded to be applied to more complex dynamic systems for future research.

Keywords: Structural Dynamics, MDOF system, Eigenperiod, Natural frequency, NN, prediction.

ISSN:2623-3347
Peer-review under responsibility of the organizing committee of COMPDYN 2021.
doi: 10.7712/120121.8755.20415

1 INTRODUCTION

The field of structural dynamics covers the behavior of structures subjected to dynamic loading. Such loads include wind, waves, traffic, earthquake, blasts and others. In practice, the dynamic response of buildings is obtained by applying a series of assumptions and simplifications that attempt to capture the most relevant dynamical properties of the system. In structural engineering, it is common to use the finite element method (FEM) to perform this kind of analysis [1]. In this case, the structure, which is essentially a continuous system, is idealized into a series of lumped masses interconnected by elements acting as springs and providing stiffness.

Using FEM, there are various methodologies that can be used to perform the dynamic analysis of a building against earthquake loading, such as explicit step-by-step integration of the coupled equations of motion [2], explicit step-by-step integration of the full (or a partial) set of uncoupled equations, response spectrum modal analysis (RSMA) [3] and the simplified equivalent lateral force (ELF) method [4]. Although RSMA and ELF are both approximate methods, they can give good results that can be used in engineering practice, especially if certain conditions are met. RSMA is widely used today as the standard procedure for designing earthquake resistant structures.

Both RSMA and ELF require the calculation of the natural periods (or eigenperiods) of the structure, i.e. the periods at which it will naturally resonate. These periods of vibration are very important in earthquake engineering and in structural dynamics in general. In structural design against earthquake loading, it is imperative that the natural frequency of the building does not match the frequency of expected earthquakes in the region the building is to be constructed. If the two frequencies match, resonance will occur, and the structure can experience severe damage. Determining the natural frequencies, or equivalently the eigenperiods of buildings, is of great importance in earthquake engineering.

The types of equations arising in FEM-based modal analysis are those seen in algebraic eigensystems. The physical interpretation of the eigenvalues and eigenvectors which come from solving the system is that they represent the frequencies and the corresponding mode shapes of the structure. In practical applications, the main desired modes are the lowest frequencies (or highest eigenperiods) because they represent the most prominent modes at which the structure will vibrate, dominating all higher frequency modes. In other words, the first (greatest) eigenperiods are the most important for determining the dynamic response of a structure. The ELF method is particularly based on the very first eigenperiod and the corresponding first eigenmode, only, while RSMA usually takes into account a limited number of eigenperiods and eigenmodes.

ANNs have been successfully employed in the structural engineering field to replace complex and time-consuming mathematical procedures and give predictions of analysis results, among others. Plevris and Asteris [5, 6] proposed a novel method with applying ANNs to approximate the failure surface for brittle anisotropic materials. Afaq et al. [7] created a novel knowledge-based structural analysis system based on ANN to predict the load-carrying capacity of RC members, reducing substantially the computational time compared to traditional Nonlinear finite element analysis. ANNs have also been used for the dynamic analysis of structures. Oh et al. [8] proposed a model to predict the seismic response of buildings based on the correlation of the ground motion and the structure using ANN. Worden and Green [9] developed a machine learning approach for the inversion of the modal transformation in nonlinear modal analysis. Gu and Oyadiji [10] presented an innovative approach using diagonal recurrent neural networks for the control of vibration in structural systems overperforming conventional control strategies such as the linear quadratic regulator for linear MDOF systems.

Bojórquez et. al [11] employed a Neural Network approach to obtain the transformation factors from single-degree of freedom (SDOF) to multi-degree of freedom (MDOF) to predict the seismic response of steel framed buildings, reducing considerably the computational effort. Payán-Serrano et al. [12] investigated the prediction of maximum story drift of MDOF structures subjected to dynamic wind load using ANNs through the combination of several structural and turbulent wind parameters. Chakraverty et al. [13] used ANNs to simulate and estimate the structural response of a two-story shear building by training the model for a particular earthquake. Lagaros and Papadrakakis [14] proposed a new adaptive scheme to predict the structural non-linear behavior when earthquake actions of increased severity are considered.

In this paper, a methodology is developed to replace the numerical procedure used for the calculation of the eigenperiods of MDOF shear buildings with an Artificial Neural Network (ANN). Two models with 3 and 5 stories are examined, where the mass and the stiffness are held constant for every story. For every model, a database with the solution of several idealized models with varying mass (m) and stiffness (k) is created using a number of (m, k) pairs. Finally, an ANN is trained with the database to predict the eigenperiods of other similar models avoiding the direct computation of the eigenvalue problem. The remaining of this paper is organized as follows: In Section 2, a brief description of dynamic modal analysis is given with the definitions of SDOF and MDOF systems and the relevant equations. In section 3, the concept of ANNs for regression problems is explained, together with the introduction of some error metrics that can be used for NN predictions. Section 4 presents the methodology and the numerical results followed by section 5 where the conclusions and future research directions are discussed.

2 DYNAMIC MODAL ANALYSIS

Modal analysis is based on using the mass and stiffness of a structure to find the various periods at which it will naturally resonate. The RSMA method requires the calculation of the natural mode shapes and frequencies of a structure during free vibration as the method is based on the superposition of the responses of the structure and the use of a response spectrum. Instead of solving the time history problem for each mode, the method uses the response spectrum to compute the maximum response in each mode. Then the maximum modal responses are combined using some statistical technique, such as square root of the sum of the squares (SRSS) or complete quadratic combination (CQC).

2.1 Single-degree of Freedom System

An idealized shear building resulting in a SDOF system is depicted in Figure 1, where m is the mass of the system and k is its stiffness, provided by the column(s).

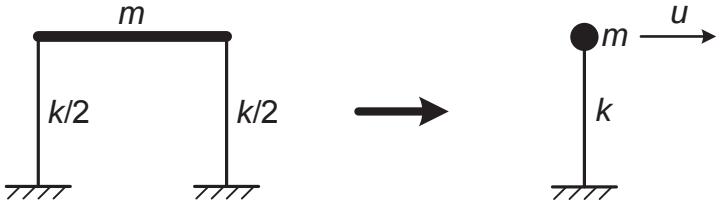


Figure 1: A SDOF dynamic system.

The dynamic equilibrium of the system at any instant of time t can be expressed as follows:

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = F(t) \tag{1}$$

where $\ddot{u}$, $\dot{u}$ and u are the acceleration, velocity, and displacement, respectively, c is the damping of the system and F is the externally imposed dynamic loading. Eq. (1) is referred as the equation of motion and its solution is a well-known problem that has been studied thoroughly by many authors [15]. For a SDOF system without damping (c=0), the circular frequency ω of the system is given by the formula

$$\omega = \sqrt{\frac{k}{m}} \tag{2}$$

The cyclic frequency f is given by

$$f = \frac{\omega}{2\pi} \tag{3}$$

and the natural period of vibration, or eigenperiod T, is given by

$$T = \frac{1}{f} = \frac{2\pi}{\omega} = 2\pi\sqrt{\frac{m}{k}} \tag{4}$$

2.2 Multi-degree of Freedom System

The number of degrees of freedom for a vibrating system depends on the number of inertial elements (masses or rigid bodies) and the number of constraints imposed on the motion. A shear building idealized into a MDOF system with 3 degrees of freedom is depicted in Figure 2.

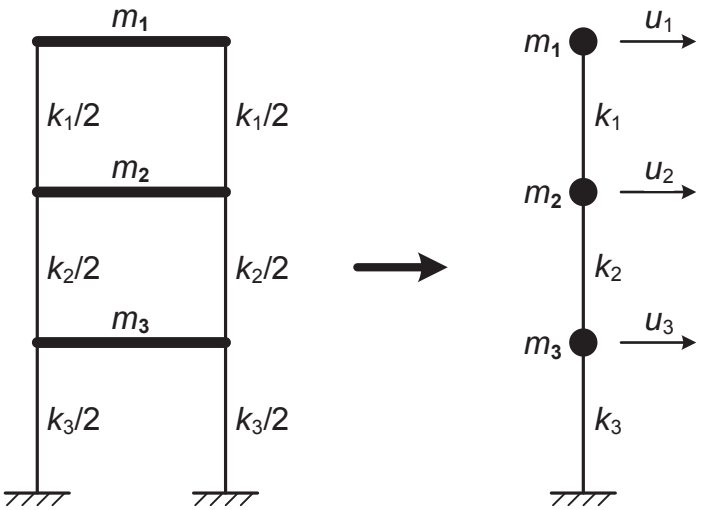


Figure 2: A MDOF dynamic system with 3 degrees of freedom (shear building).

The dynamic equilibrium of a MDOF structure at any instant of time t can be expressed with the following equation of motion:

$$\mathbf{M}\ddot{\boldsymbol{u}}(t) + \mathbf{C}\dot{\boldsymbol{u}}(t) + \mathbf{K}\boldsymbol{u}(t) = \boldsymbol{F}(t) \tag{5}$$

where $\mathbf{M}$ is the mass matrix; $\mathbf{K}$ is the stiffness matrix; $\mathbf{C}$ represents the damping matrix, $\ddot{\boldsymbol{u}}$, $\dot{\boldsymbol{u}}$ and $\boldsymbol{u}$ are the acceleration, velocity and displacement vectors, respectively, and $\boldsymbol{F}$ is the external force vector.

In the case of free vibration without damping, Eq. (5) becomes

$$\mathbf{M}\ddot{\boldsymbol{u}}(t) + \mathbf{K}\boldsymbol{u}(t) = \mathbf{0} \tag{6}$$

In free vibration without damping, the system will oscillate in a steady-state harmonic manner. The resulting motion will be a mixture of modes, each having its own circular frequency ω_i so that

$$\ddot{\boldsymbol{u}}(t) = -\omega_i^2 \boldsymbol{u}(t) \tag{7}$$

By substituting Eq. (7) on Eq. (6), the following expression is obtained

$$(\mathbf{K} - \omega_i^2 \mathbf{M}) \cdot \boldsymbol{u}(t) = \mathbf{0} \tag{8}$$

The n circular frequencies ω_i (i=1, 2, …, n) of the system can be found using Eq. (8), where the trivial solution is obtained when $\boldsymbol{u}(t)$=**0**. For a non-trivial solution, the term inside the first parenthesis in Eq. (8) must be equal to zero, such that

$$\left|\mathbf{K} - \omega_i^2 \mathbf{M}\right| = 0 \tag{9}$$

After having found the circular frequencies ω_i, we can now substitute those values back into Eq. (8) and solve for the vector $\boldsymbol{u}$ which corresponds to the i-th modal shape. Each modal shape or eigenvector is denoted as $\boldsymbol{\Phi}_i$ (n×1) so that Eq. (8) can be rewritten as

$$\left(\mathbf{K} - \omega_i^2 \mathbf{M}\right) \cdot \boldsymbol{\Phi}_i = \mathbf{0} \tag{10}$$

Eq. (10) represents a classic problem in Mathematics referred as generalized eigenvalue problem [16] and its solution involves a series of matrix decompositions which can be computationally expensive, especially for large systems.

For the special case of a MDOF shear building with n degrees of freedom (stories), like the one of Figure 2 (shown there for the case n=3), the mass matrix **M** is a n×n diagonal matrix given by

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & m_n \end{bmatrix} \tag{11}$$

where m_i is the mass of every story starting numbering from the top to the bottom, according to the notation used in this manuscript. The stiffness matrix **K** is a n×n square matrix calculated as

$$\mathbf{K} = \begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 \\ 0 & -k_2 & \ddots & -k_{n-1} \\ 0 & 0 & -k_{n-1} & k_{n-1} + k_n \end{bmatrix} \tag{12}$$

where k_i is the stiffness of each story, with numbering again from top to bottom.

3 ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network is a collection of interconnected units referred as artificial neurons that resembles the basic structure of biological neurons that constitute animal and human brains. It is known that when a biological brain receives an input signal such as the desire to move a limb, the signal is processed through the interconnected neurons inside the brain to produce a corresponding output signal as a response [17]. In the case of moving a

limb, the response is a signal that contracts the required muscles to move it. The connectivity of the neurons inside the brain is the fundamental key to make that happen. Through the experiences of the individual, the connections of the neurons are constantly self-adjusting to improve the quality of the response. This process is what we, humans, know as learning.

An ANN tries to mimic the functionality of biological brains using an analogous but much more simplistic model. The ANN receives a set of numeric input values that are processed by the artificial neurons to produce a numeric response. To obtain the desired or "correct" response, the ANN is trained with known data containing numerous examples of inputs with their matching output values. Then, by using a mathematical optimization algorithm, the network parameters are adjusted so that the ANN learns to map the inputs to their corresponding outputs. Therefore, a properly trained ANN can be used to accurately predict the output values of new inputs that were not used in the training data [18].

3.1 Back Propagation Neural Networks

For this study, the selected ANN type is a backpropagation neural network (BPNN). The BPNN is composed of several layers and each layer has a specific number of neurons. The neurons of the first layer are connected to the neurons of the second layer, which in turn are connected to the neurons of the third layer, and so on. The very first layer and the last layer are known as the input and output layers respectively, while the other layers are called "hidden layers". The notation used to denote the layer composition is generally as follows:

$$N - H_1 - H_2 - \ldots - H_{L\text{-}1} - M,$$

where N denotes the number of neurons in the input layer (number of inputs), H_i stands for the number of neurons in the i-th hidden layer, M is the number of neurons in the output layer, and L is the total number of layers, including the output layer but excluding the input layer. For example, a 2-3-3-2 BPNN consists of an input layer with 2 neuron, 2 hidden layers with 3 neurons each, and an output layer with 2 neurons, as shown in Figure 3.

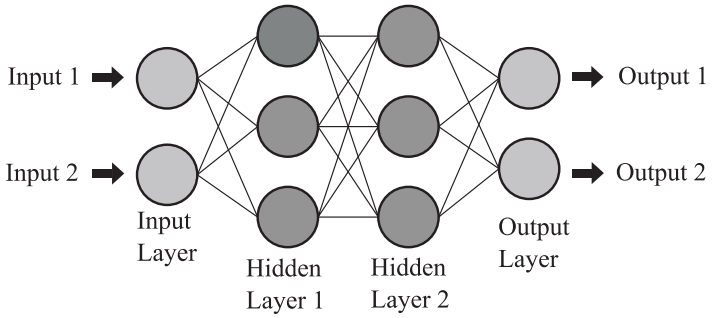


Figure 3: A 2-3-3-2 BPNN.

To obtain the output values from the ANN, the input values are fed to the network by a feedforward process. It starts by setting the input values at the first layer N, and then, moving forward to the next layer, the values of all the neurons at that layer are computed. The process is repeated until the output layer M is reached and the output values are obtained. At each neuron, a weighted summation of the values coming from the other connected neurons and an additional bias neuron is performed, see Figure 4.

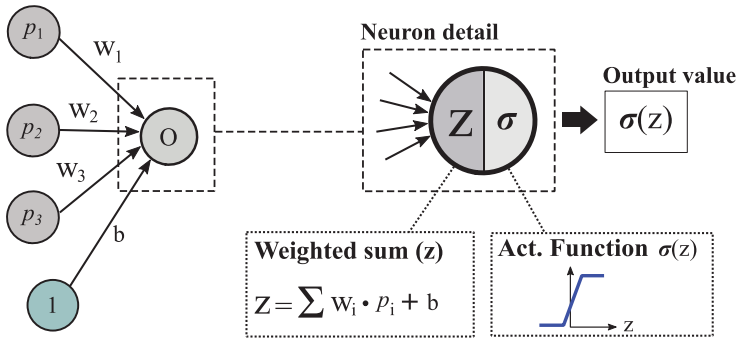


Figure 4: Detail of a single neuron connected to 3 other neurons from the previous layer. The picture shows the process on how the numerical output value is obtained in a neuron.

For the case of Figure 4, the operation can be written in matrix form as follows

$$z = \boldsymbol{W} \cdot \boldsymbol{p} + b \tag{13}$$

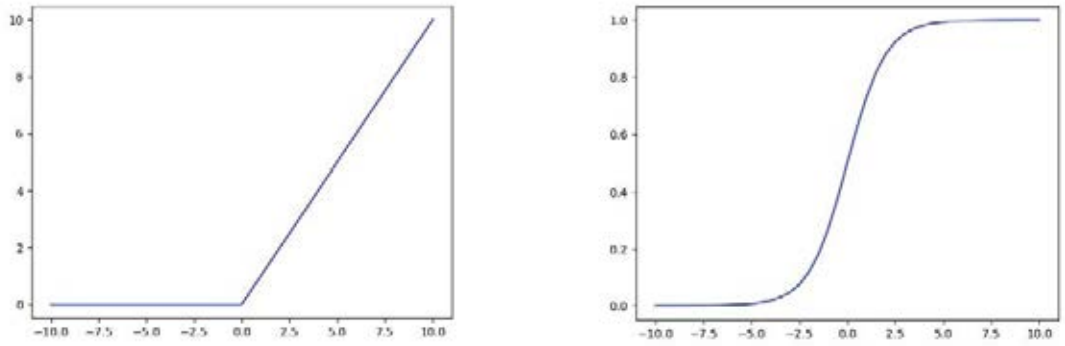
$$\boldsymbol{W} = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \tag{14}$$

$$\boldsymbol{p} = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^T \tag{15}$$

$$O = \sigma(z) = \sigma(\boldsymbol{W} \cdot \boldsymbol{p} + b) \tag{16}$$

where O is the output of the output neuron, $\boldsymbol{p}$ is a column vector with the values of the neurons at the previous layer; $\boldsymbol{W}$ is the weight row vector that connects this output neuron with the previous layer; and b is a bias neuron that is added to the weighted summation to form the input for the activation function σ.

Activation functions are used to add non-linearity into the neural network and to facilitate the training process in which the gradient of the model parameters is computed. Additionally, they help to restrict the neuron values to a certain limit providing stability to the network. Therefore, the activation functions are usually simple, continuous, and differentiable. Among the most commonly used are the Linear, ReLu and Sigmoid activation functions. Figure 5 presents the Relu and the Sigmoid activation functions.



(a) (b)

Figure 5: ReLu (a) and Sigmoid (b) activation functions, plotted from -10 to 10.

3.2 Training Process

The training process consists of adjusting the initial random weights of the network to fit the given training data. The weight updating process starts by feeding a set of input values

through the network and obtaining the corresponding outputs. The obtained values are then compared with their known values from the training dataset and an error is estimated using a predefined error function. The idea is to determine the gradient or the rate of change of the error function with respect to the model parameters (i.e. all the weights). Thanks to the network architecture, the computation of the gradient is possible using a chain rule of partial derivatives with the known quantities. Then, by using the back propagation scheme, the gradients computed at the output neurons are propagated back through the hidden layers so that a gradient value is defined at each one of the weights in the model. Finally, the weights of the model are updated by a small quantity obtained by multiplying the gradients with a factor called *learning rate*. This weight-update scheme is an optimization method known as gradient decent (GD) which attempts to minimize the error function.

To properly train the BPNN, the full training dataset must be processed by the network multiple times. In other words, the weights must be updated with the GD scheme for every data point in the training set. Every time that the full training set has been processed with the GD scheme is called an Epoch. However, at every Epoch, the data must be fed in different order as it turns out that if the full training data is presented multiple times in a random order, the overall fluctuations of the gradient update path will average out and converge to a good solution. Such random reordering of the training dataset at every Epoch is known as stochastic gradient decent (SGD) and it is one of the most popular weight-optimization schemes for ANNs.

The training process can be quite demanding in terms of computational effort for large NN models with large training data sets. Nevertheless, in this paper that is not the case as the investigated problem can be solved with a relatively simple BPNN architecture and a small amount of training data. The stopping criteria of the training process is usually monitored by having a separated validation set of data which is not used for training. At every Epoch, the error of both the training and the validation sets are measured. Naturally, the error of the training set reduces with every Epoch due to the optimization scheme. However, if the error on the validation set suddenly starts to increase, while the training set error keeps reducing, it is a clear indication of an unwanted characteristic called overfitting in which the NN loses its generality. Therefore, the stopping criteria is set to the last Epoch before the overfitting was detected. Alternatively, the stopping criterion can be set for a fixed number of Epochs.

3.3 Prediction error metrics

The NN predictions are never completely accurate as each prediction always contains a small error. There are various metrics that can be used to measure the prediction error. Let $\boldsymbol{p}$ ($N\times1$ vector) be the predicted values and $\boldsymbol{x}$ ($N\times1$ vector) be the real values of a quantity calculated (or measured) a number N of times. For example, for predicting the fundamental period of a MDOF system, we ask the NN to predict it N times using different pairs of m and k parameters. The following metrics can be defined for calculating the prediction error.

The prediction error values $\boldsymbol{Err}$ (or residuals) can be expressed as the difference between the predicted values and the real values, as follows:

$$\boldsymbol{Err} = \boldsymbol{p} - \boldsymbol{x} \tag{17}$$

The absolute value of the residuals (or absolute error) $\boldsymbol{AbsErr}$ is a measure of how far from the regression line data points are. Each element $AbsErr_i$ of the vector $\boldsymbol{AbsErr}$ is given by

$$AbsErr_i = \left|Err_i\right| = \left|p_i - x_i\right| \tag{18}$$

The mean absolute error (MAE) is the average value of the absolute error, given as

$$MAE = \frac{\sum_{i=1}^{N} AbsErr_i}{N} = \frac{\sum_{i=1}^{N} |p_i - x_i|}{N} \tag{19}$$

The squared error ***SqErr*** is the squared value of the error, with its elements given by

$$SqErr_i = (p_i - x_i)^2 \tag{20}$$

The mean squared error (*MSE*) is the average value of the squared errors, given by

$$MSE = \frac{1}{N}\sum_{i=1}^{N} SqErr_i = \frac{1}{N}\sum_{i=1}^{N} (p_i - x_i)^2 \tag{21}$$

The root mean squared error (*RMSE*) is the square root of *MSE*, given by

$$RMSE = \sqrt{MSE} \tag{22}$$

The relative error values ***RelErr*** is the absolute error divided by the real value, provided that the real value is not equal to zero. It is given by

$$RelErr_i = \frac{|Err_i|}{x_i} = \frac{|p_i - x_i|}{x_i} \tag{23}$$

The mean relative error (*MRE*) is the average value of the relative errors, given by

$$MRE = \frac{1}{N}\sum_{i=1}^{N} RelErr_i = \frac{1}{N}\sum_{i=1}^{N} \frac{|p_i - x_i|}{x_i} \tag{24}$$

The Pearson correlation coefficient (*R*) is a measure of linear correlation between two sets of data, in our case the predicted values and the real values. It is the covariance of two variables, divided by the product of their standard deviations. Thus, it is essentially a normalized measurement of the covariance, such that the result always has a value between −1 and 1. If there is no error in the prediction, then the predicted values coincide with the real values and R=1. R is given by the formula

$$R = \frac{\sum_{i=1}^{N} (p_i - \bar{p})(x_i - \bar{x})}{\sqrt{\sum_{i=1}^{N} (p_i - \bar{p})^2}\sqrt{\sum_{i=1}^{N} (x_i - \bar{x})^2}} \tag{25}$$

In the above formula, $\bar{p}$ and $\bar{x}$ are the mean values of $\boldsymbol{p}$ and $\boldsymbol{x}$, respectively:

$$\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{26}$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{27}$$

4 NUMERICAL RESULTS

Two models have been examined, a 3-DOF shear building and a 5-DOF shear building. For both models, the parameters m (mass) and k (stiffness) are the same for all stories, as shown in Figure 6. This limits the input parameters of the Neural Network to 2 in every case.

The output parameters are the eigenperiods of the system, which are 3 for the 3-DOF system and 5 for the 5-DOF system. The mass m varies from m_{min}=2 ton to m_{max}=20 ton while the stiffness k varies from k_{min}=20,000 kN/m to k_{max}=200,000 kN/m.

We use standard intervals of m and k for the generation of the data used for training. The mass and stiffness intervals are set to 6, which means that we have 7 values of mass and 7 values of stiffness which makes 7×7=49 pairs of (m, k) for each structure. 70% of the database is used for training (35 pairs), while the rest (30%) of the database is used for validation (15%, 7 pairs) and testing (15%, 7 pairs).

For both cases, we use a simple NN architecture of one hidden layer with 10 neurons. In the 3-DOF system, the network architecture is 2-10-3 while for the 5-DOF system, the network architecture is 2-10-5.

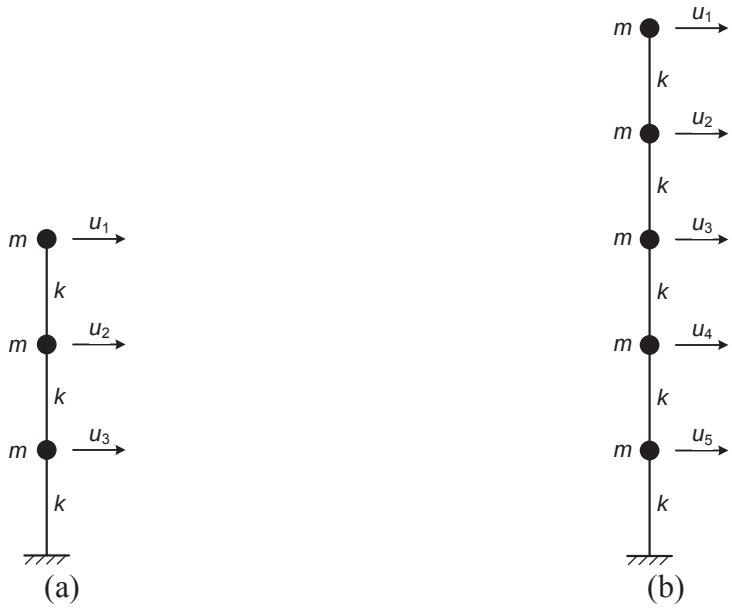


Figure 6: (a) 3-DOF shear building, (b) 5-DOF shear building.

4.1 Three DOF system

The first numerical example is a 3-DOF system where the NN is asked to predict the three eigenperiods of the system, i.e. T_1, T_2, T_3 where $T_1>T_2>T_3$. The NN architecture used is 2-10-3. The maximum number of validation increases is set to 6, i.e. if the error in the validation set increases for 6 times then the training is stopped. Given this criterion, training is stopped in Epoch 110, as shown in Figure 7.

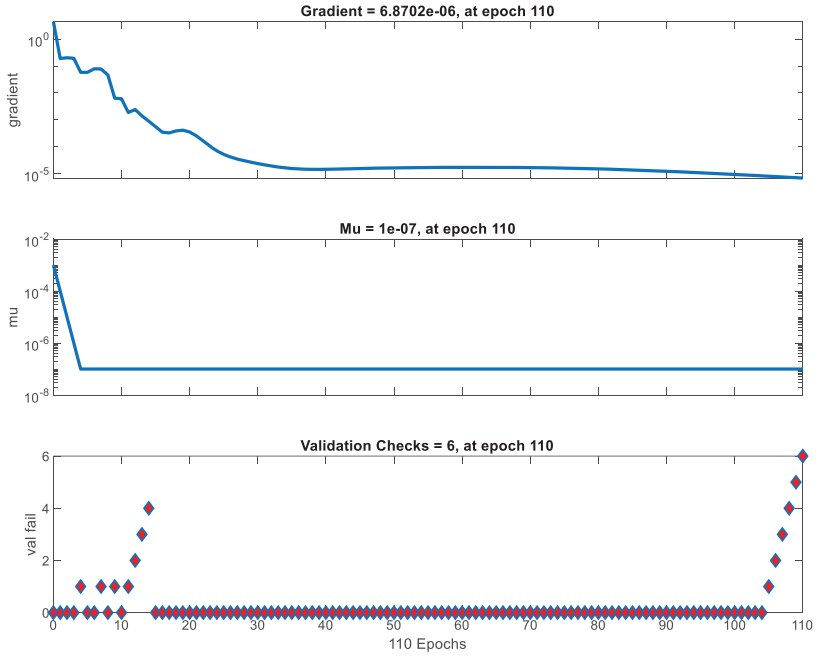


Figure 7: Example 1, Final NN Train State.

The performance of the NN is depicted in Figure 8, where the three lines show the error obtained for the training set (blue line), the validation set (green line) and the test set (red line).

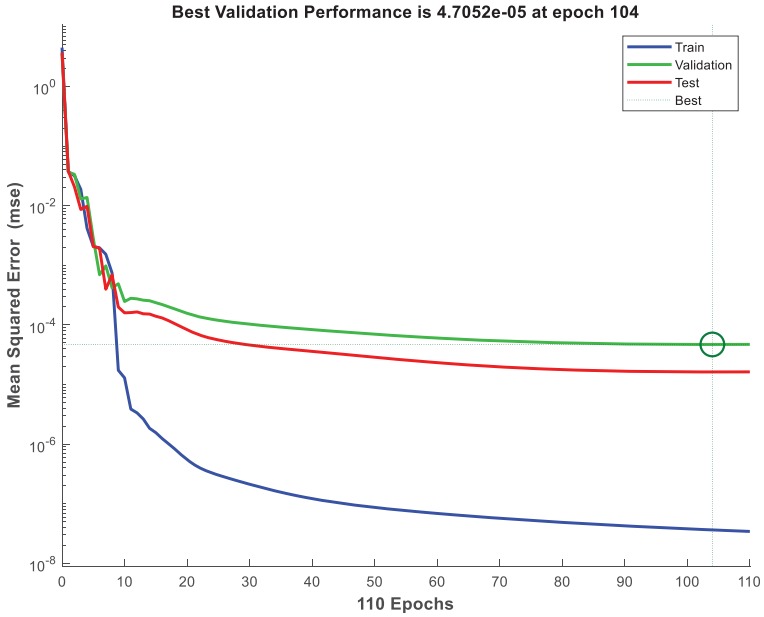


Figure 8: Example 1, NN Training Performance.

The training regression of the NN data is depicted in Figure 8, where the three colored lines correspond again to the three different sets and the fourth black line contains all the data sets together. The *R* values reported correspond to the normalized data.

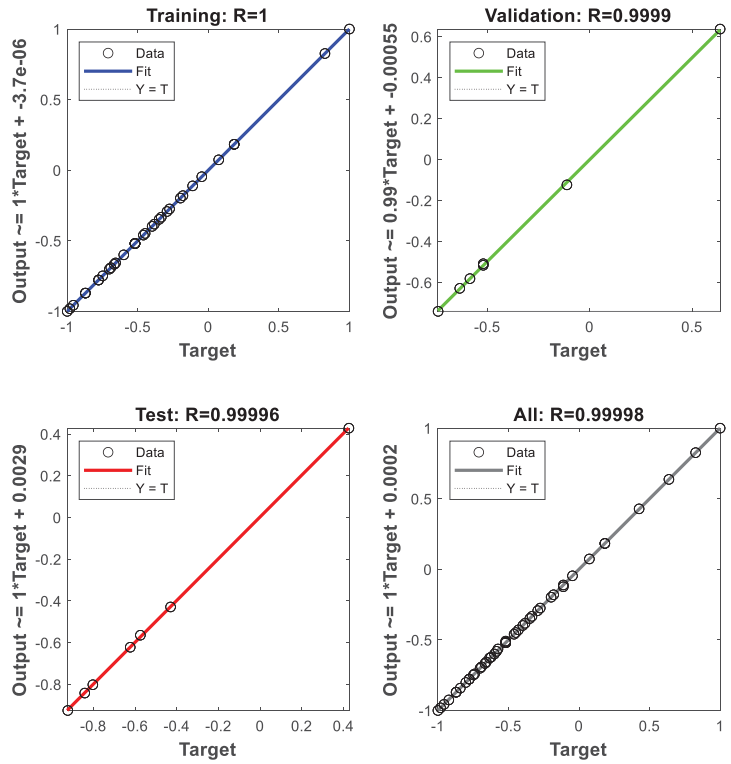


Figure 9: Example 1, NN Training Regression (for the Normalized Data).

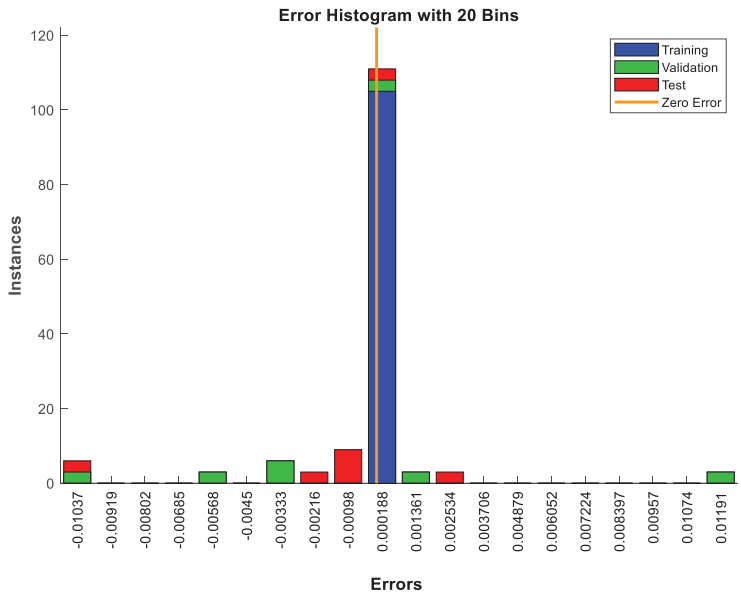


Figure 10: Example 1, NN Error Histogram (for the Normalized Data).

In order to test the NN results, we generated 100 random (m, k) pairs within the training region and we calculated the errors in the NN predictions. The results are presented in Table 1.

Metric	T_1	T_2	T_3	All periods
MSE	4.33E-06	5.52E-07	2.64E-07	1.72E-06
RMSE	2.08E-03	7.43E-04	5.14E-04	1.31E-03
MAE	8.46E-04	3.02E-04	2.09E-04	4.52E-04
MRE	4.10E-03	4.10E-03	4.10E-03	4.10E-03
R	0.999745	0.999745	0.999745	0.99986

Table 1: Example 1, Error metrics for the independent test with 100 random (m, k) pairs..

It is shown that the error of the NN predictions is very small. The mean relative error is only 0.41% for all the NN predictions together (T_1, T_2 and T_3). As an example, for m=5.52 ton and k=137,000 kN/m, the real values of the eigenperiods are T_1=0.0896 s, T_2=0.0320 s and T_3=0.0221 s, as shown in Figure 11 together with the three corresponding eigenmodes. The corresponding NN predictions are T_1=0.0898 s, T_2=0.0320 s and T_3=0.0222 s. The NN manages to give very accurate results, although the data set used contains only 49 (m, k) pairs, of which only 35 (70%) have been used for the NN training itself.

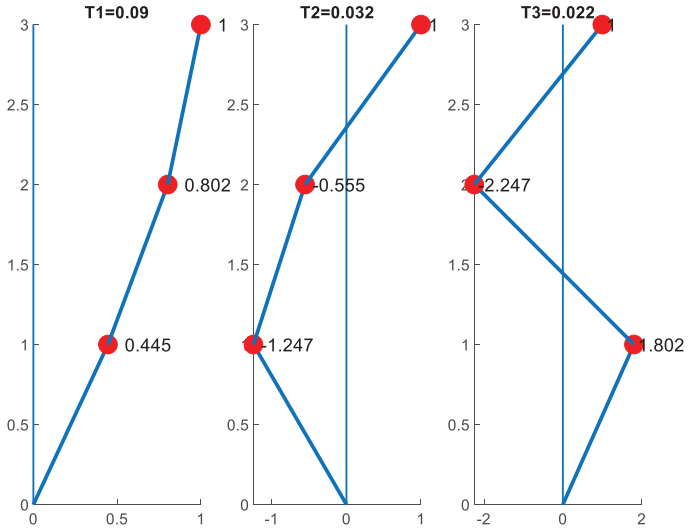


Figure 11: Eigenmodes of a 3-DOF shear building with m=5.52 ton and k=137,000 kN/m (uniformly, at every story).

4.2 Five DOF system

The second numerical example is a 5-DOF system where the NN is asked to predict the five eigenperiods of the system, i.e. T_1 to T_5, where T_1>T_2>T_3>T_4>T_5. The NN architecture used is 2-10-5, with 10 neurons in the hidden layer. The maximum number of validation increases is again set to 6. This time training stops earlier, in Epoch 30, as shown in Figure 12.

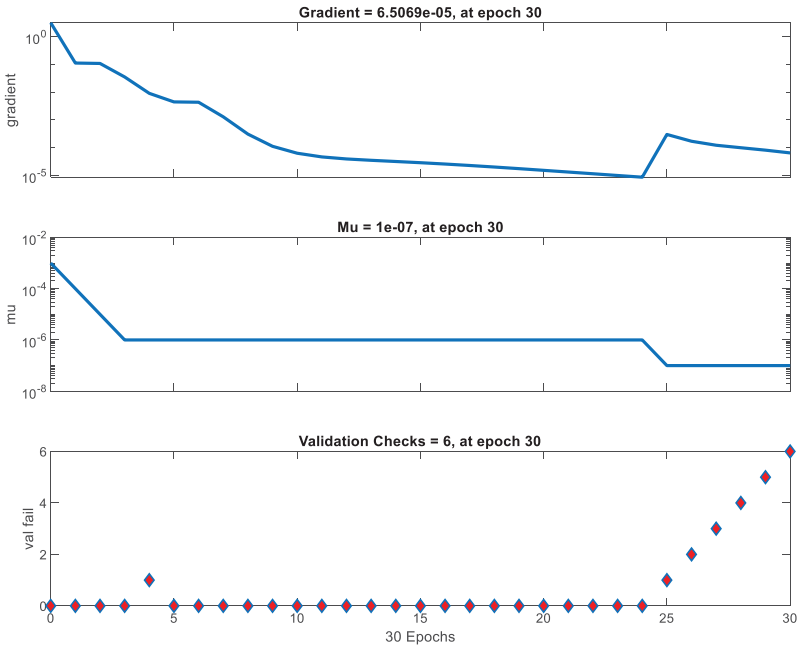


Figure 12: Example 2, Final NN Train State.

The performance of the NN is depicted in Figure 13, where the three lines show the error obtained for the training set (blue line), the validation set (green line) and the test set (red line).

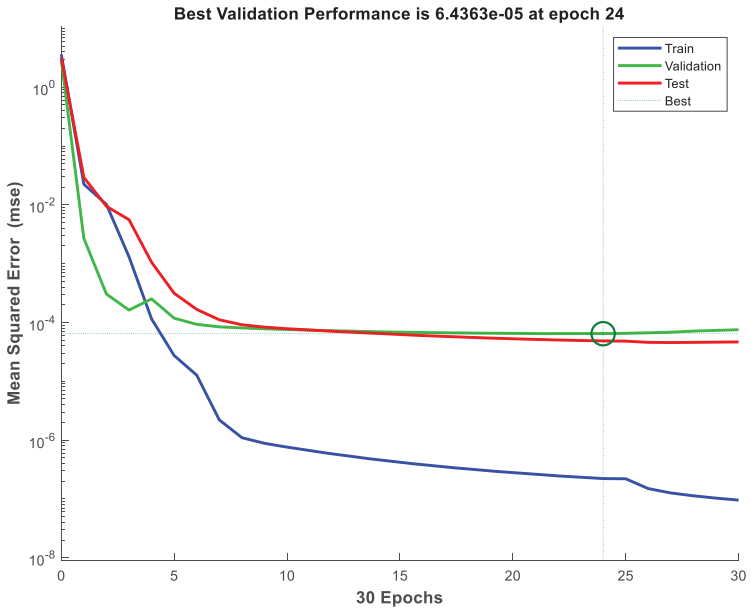


Figure 13: Example 2, NN Training Performance.

The training regression of the NN data is depicted in Figure 14, where the three colored lines correspond again to the three different sets and the fourth black line contains all the data sets together. The *R* values reported correspond to the normalized data.

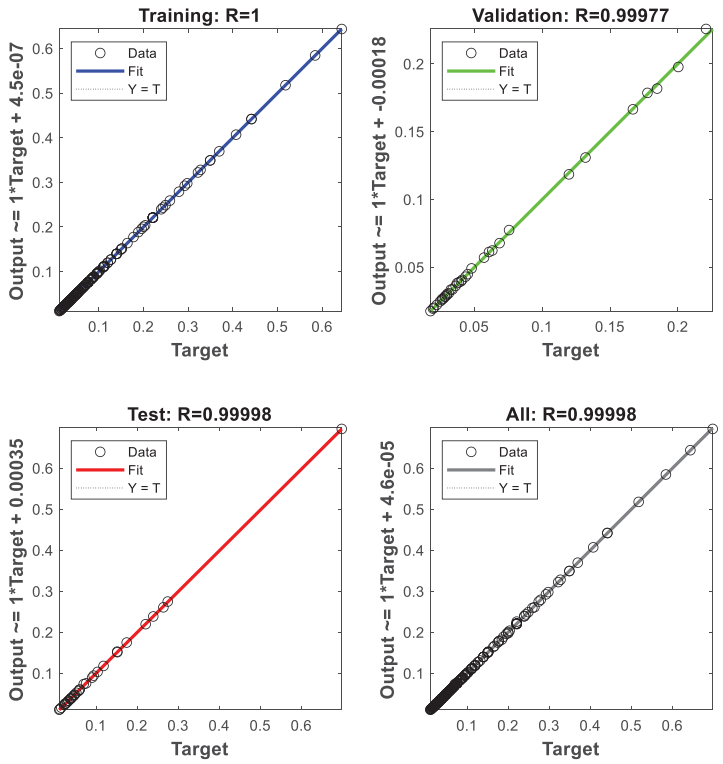


Figure 14: Example 2, NN Training Regression (for the Normalized Data).

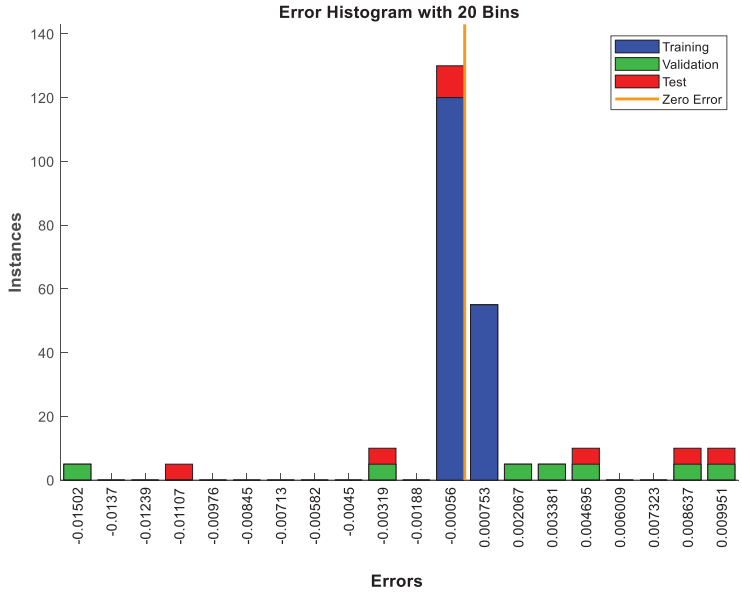


Figure 15: Example 2, NN Error Histogram (for the Normalized Data).

In order to test the NN results, we generated 100 random (m, k) pairs within the training region and we calculated the errors in the NN predictions. The results are presented in Table 1.

Metric	T_1	T_2	T_3	T_4	T_5	All periods
MSE	1.52E-05	1.78E-06	7.18E-07	4.35E-07	3.34E-07	3.69E-06
RMSE	3.90E-03	1.34E-03	8.47E-04	6.60E-04	5.78E-04	1.92E-03
MAE	2.02E-03	6.94E-04	4.40E-04	3.43E-04	3.00E-04	7.60E-04
MRE	7.95E-03	7.95E-03	7.95E-03	7.95E-03	7.95E-03	7.95E-03
R	0.999678	0.999678	0.999678	0.999678	0.999678	0.999859

Table 2: Example 2, Error metrics for the independent test with 100 random (m, k) pairs.

It is shown that the error of the NN predictions is again very small. The mean relative error is only 0.79% for all the NN predictions together (T_1 to T_5). As an example, for m=5.52 ton and k=137,000 kN/m, the real values of the eigenperiods are T_1=0.1401 s, T_2=0.0480 s, T_3=0.0305 s, T_4=0.0237 s, T_5=0.0208 s, as shown in Figure 16. The corresponding NN predictions are T_1=0.1384 s, T_2=0.0474 s, T_3=0.0301 s, T_4=0.0234 s, T_5=0.0205 s. The NN manages to give very accurate results for all five eigenperiods.

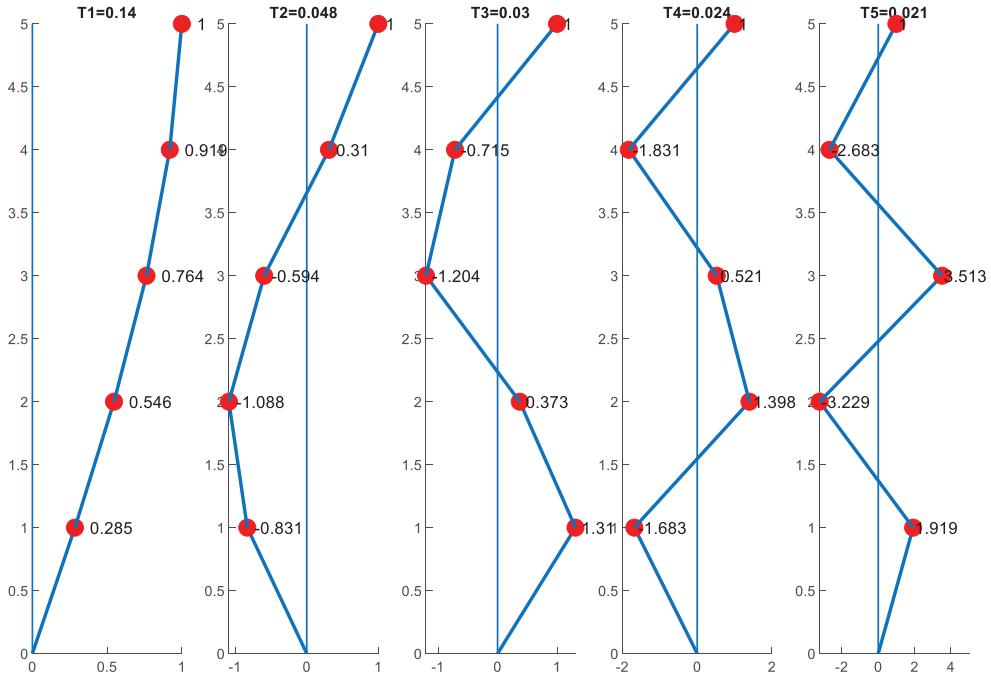


Figure 16: Eigenmodes of a 5-DOF shear building with m=5.52 ton and k=137,000 kN/m (uniformly, at every story).

5 CONCLUSIONS

In the present study, a methodology to predict the dynamic properties of a MDOF idealized model using a machine learning oriented strategy is implemented. An Artificial Neural Network is trained to accurately predict all the eigenperiods of MDOF shear buildings with a uniform distribution of mass and stiffness at every story. The goal of the study has been to reduce the computational effort of the complex mathematical process that is required to solve the generalized eigenvalue problem that arises from the modal analysis. It is noted that to train the ANN, a training database of previously solved eigenvalue problems is required, which is itself also a computationally expensive operation. However, the creating of the training set and the

training procedure are one-time processes, only. After the ANN is trained, it can be used to predict the behavior of any system (within the domain of the training data) with a very low computational effort.

The methodology has been tested for two different MDOF models, a 3-DOF system and a 5-DOF system. The results show highly accurate predictions reaching mean relative error values of 0.41% and 0.79%, respectively. All the error metrics used show a highly accurate prediction. This research has shown that for this type of problems, a relatively small database can be used to train highly accurate BPNNs. The developed framework is going to serve as a base for further research into more complex dynamic systems, for example systems with more degrees of freedom and also multi-story shear buildings with independent values of m and k for every single story. Having independent values of mass and stiffness at each story dramatically increases the complexity of the problem as the input parameters increase from 2 to 3×2=6 and 5×2=10 for the 3-DOF and the 5-DOF system, respectively.

REFERENCES

[1] Plevris, V. and G. Markeset, *Educational Challenges in Computer-based Finite Element Analysis and Design of Structures.* Journal of Computer Science, 2018. **14**(10): p. 1351-1362.

[2] Papazafeiropoulos, G. and V. Plevris, *OpenSeismoMatlab: A new open-source software for strong ground motion data processing.* Heliyon, 2018. **4**(9): p. 1-39.

[3] Arias, H. and J.D. Jaramillo, *Base shear determination using response-spectrum modal analysis of multi-degree-of-freedom systems with soil–structure interaction.* Bulletin of Earthquake Engineering, 2019. **17**(7): p. 3801-3814 DOI: 10.1007/s10518-019-00612-5.

[4] Yenidogan, C. and M. Erdik, *A comparative evaluation of design provisions for seismically isolated buildings.* Soil Dynamics and Earthquake Engineering, 2016. **90**: p. 265-286 DOI: https://doi.org/10.1016/j.soildyn.2016.08.016.

[5] Plevris, V. and P.G. Asteris, *Modeling of Masonry Failure Surface under Biaxial Compressive Stress Using Neural Networks.* Construction and Building Materials, 2014. **55**: p. 447-461 DOI: 10.1016/j.conbuildmat.2014.01.041.

[6] Asteris, P.G. and V. Plevris, *Anisotropic masonry failure criterion using artificial neural networks.* Neural Computing and Applications, 2017. **28**(8): p. 2207-2229 DOI: 10.1007/s00521-016-2181-3.

[7] Ahmad, A., D.M. Cotsovos, and N.D. Lagaros, *Framework for the development of artificial neural networks for predicting the load carrying capacity of RC members.* SN Applied Sciences, 2020. **2**(4): p. 1-21.

[8] Oh, B.K., et al., *Neural network-based seismic response prediction model for building structures using artificial earthquakes.* Journal of Sound and Vibration, 2020. **468**: p. 115109 DOI: https://doi.org/10.1016/j.jsv.2019.115109.

[9] Worden, K. and P.L. Green, *A machine learning approach to nonlinear modal analysis.* Mechanical Systems and Signal Processing, 2017. **84**: p. 34-53 DOI: https://doi.org/10.1016/j.ymssp.2016.04.029.

[10] Gu, Z.Q. and S.O. Oyadiji, *Diagonal Recurrent Neural Networks for MDOF Structural Vibration Control.* Journal of Vibration and Acoustics, 2008. **130**(6) DOI: 10.1115/1.2948369.

[11] Bojórquez, J., et al., *Probabilistic seismic response transformation factors between SDOF and MDOF systems using artificial neural networks.* Journal of Vibroengineering, 2016. **18**(4): p. 2248-2262 DOI: 10.21595/jve.2016.16506.

[12] Payán-Serrano, O., et al., *Prediction of Maximum Story Drift of MDOF Structures under Simulated Wind Loads Using Artificial Neural Networks.* Applied Sciences, 2017. **7**(6): p. 563.

[13] Chakraverty, S., P. Gupta, and S. Sharma, *Neural network-based simulation for response identification of two-storey shear building subject to earthquake motion.* Neural Computing and Applications, 2010. **19**(3): p. 367-375 DOI: 10.1007/s00521-009-0279-6.

[14] Lagaros, N.D. and M. Papadrakakis, *Neural network based prediction schemes of the non-linear seismic response of 3D buildings.* Advances in Engineering Software, 2012. **44**(1): p. 92-115 DOI: https://doi.org/10.1016/j.advengsoft.2011.05.033.

[15] Chopra, A.K., *Dynamics of structures. theory and applications to Earthquake Engineering*. 2017.

[16] Ghojogh, B., F. Karray, and M. Crowley, *Eigenvalue and Generalized Eigenvalue Problems: Tutorial.* ArXiv e-prints, 2019(arXiv:1903.11240v1): p. 8 DOI: https://arxiv.org/abs/1903.11240v1.

[17] Rickert, J., et al., *Dynamic Encoding of Movement Direction in Motor Cortical Neurons.* The Journal of Neuroscience, 2009. **29**(44): p. 13870-13882 DOI: 10.1523/jneurosci.5441-08.2009.

[18] Plevris, V., *Innovative computational techniques for the optimum structural design considering uncertainties*. 2009, National Technical University of Athens: Athens, Greece. p. 312.