BACK-PROPAGATION NEURAL NETWORK-BASED METHOD FOR PREDICTING THE INTERVAL NATURAL FREQUENCIES OF STRUCTURES WITH UNCERTAIN-BUT-BOUNDED PARAMETERS

Pengbo Wang[1*], Wenting Jiang[2] and Qinghe Shi[3]

[1]Institute of Solid Mechanics, Beihang University, Beijing, 100191, China
[2]Institute of Engineering Thermophysics, Chinese Academy of Sciences, Beijing, 100190, China
[3]School of Materials and Engineering, Jiangsu University of Technology, Changzhou, Jiangsu, 213001, China

ABSTRACT

Uncertain-but-bounded parameters have a significant impact on the natural frequencies of structures, and it is necessary to study their inherent relationship. However, their relationship is generally nonlinear and thus very complicated. Taking advantage of the strong non-linear mapping ability and high computational efficiency of BP neural networks, namely the error back-propagation neural networks, a BP neural network-based method is proposed to predict the interval natural frequencies of structures with uncertain-but-bounded parameters. To demonstrate the proposed method's feasibility, a numerical example is tested. The lower and upper frequency bounds obtained using the proposed approach are compared with those obtained using the interval-based perturbation method, which is a commonly used method for problems with uncertainties. A Monte Carlo simulation is also conducted because it is always referred to as a reference solution for problems related to uncertainties. It can be observed that as the varying ranges of uncertain parameters become larger, the accuracy of the perturbation method deteriorates remarkably, but the proposed method still maintains a high level of accuracy. This study not only puts forward a novel approach for predicting the interval natural frequencies but also exhibits the broad application prospect of BP neural networks for solving problems with uncertainties.

KEYWORDS

Back-propagation neural network, Natural frequency, Interval parameter, Perturbation method, Monte Carlo simulation.

1. INTRODUCTION

Natural frequencies have strong effects on the dynamic behaviours of structures and are of considerable importance for analysis, design, and optimization. However, uncertainty is ubiquitous in scientific research and engineering applications, and all structural analyses and designs involve varying degrees of uncertainty [1-3]. Owing to manufacturing deviations, modelling approximations, measurement inaccuracies, and external environmental changes, we may encounter many types of information uncertainty, such as external loads, material

David C. Wyld et al. (Eds): ISPR, ARIN, MLCL, NBIoT - 2020
pp. 11-31, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.100702

properties, geometric dimensions, and boundary conditions [4-6]. Thus, the natural frequencies of a structure are affected significantly by uncertainties. A commonly used approach to deal with an uncertainty problem is to model these uncertain parameters as random variables, and the resolution process is conducted using the probabilistic theory [7]. However, information about the probabilistic distributions of random variables may be unavailable or sometimes inaccurate, and often a small error in probabilistic information may lead to serious errors in probabilistic results [8,9]. Therefore, the probabilistic approach has some intrinsic limitations.

Recently, many scholars have shifted from probabilistic theory to interval theory to deal with uncertainty problems. In interval theory, all uncertain-but-bounded parameters can be described using interval numbers [10-13]. Interval theory has been applied to structural analysis and many papers have been published regarding this approach [14-17]. An interval-based perturbation method is commonly used to deal with uncertainties and has been successfully employed in the prediction of the interval natural frequencies of structures with uncertain-but-bounded parameters [18-20]. The perturbation method is highly accurate for linear problems, but its accuracy is poor for nonlinear problems. The mathematical relationship between the uncertain parameters and the natural frequencies usually remains nonlinear. The perturbation method only remains at a high level of accuracy when the varying ranges of interval parameters are small. As the varying ranges of interval parameters become larger, the errors generated by the perturbation method also increase rapidly. To obtain an accurate interval of natural frequencies with wide varying ranges of interval parameters, we focus on the application of artificial neural networks.

Artificial neural networks possess excellent information-handling capacity and have become powerful tools to study nonlinear systems [21]. Among all types of networks, the BP neural network, namely the error back-propagation neural network, is the most maturely studied neural network and has been successfully employed in structural analysis. Xu and Zhang used a BP neural network to implement a safety assessment for a bridge crane structure [22]. Yuan et al. employed a BP neural network to optimize the static and dynamic characteristics of machine tool structures [23]. Yang et al. studied the application of BP neural networks for hydraulic metal structure health diagnosing [24]. Li et al. used a BP neural network to predict the mechanical properties of shape memory alloy [25]. Although many scholars have aimed attention at the adoption of BP neural networks for structural analysis with deterministic parameters, the research of BP neural networks for structural analysis with uncertain parameters still remains at the initial stage, and very few papers have been published. In this study, we take advantage of the BP neural network properties of strong non-linear mapping ability and high computational efficiency and propose a BP neural network-based method to predict the interval natural frequencies of structures with uncertain-but-bounded parameters.

The remainder of this paper is organized as follows. In Section 2, the problem of predicting the lower and upper bounds of natural frequencies for structures with interval parameters is briefly introduced. In Section 3, the mechanism of a BP neural network is presented. In Section 4, the procedures of a BP neural network-based method are proposed. In Section 5, a numerical example is provided to illustrate the accuracy and efficiency of the proposed method. Section 6 presents the conclusions of this study.

2. PROBLEM FORMULATION

For a general engineering structure with n degrees of freedom, the governing equation for eigenvalue problems is given by:

$$\boldsymbol{K}\boldsymbol{u} = \lambda \boldsymbol{M}\boldsymbol{u} \tag{1}$$

where $\boldsymbol{K} = (k_{ij}) \in R^{n \times n}$ and $\boldsymbol{M} = (m_{ij}) \in R^{n \times n}$ are the stiffness and mass matrices, respectively; λ represents the eigenvalue and $\boldsymbol{u}$ denotes the eigenvector. The natural frequency corresponding to λ is: $f = \sqrt{\lambda}/2\pi$.

Taking uncertainty into consideration, the stiffness and mass matrices are subjected to the following constraints

$$\underline{\boldsymbol{K}} \le \boldsymbol{K} \le \bar{\boldsymbol{K}}, \quad \underline{\boldsymbol{M}} \le \boldsymbol{M} \le \bar{\boldsymbol{M}} \tag{2}$$

where $\underline{\boldsymbol{K}}$ and $\bar{\boldsymbol{K}}$ represent the lower and upper bounds of the stiffness matrix, respectively, and $\underline{\boldsymbol{M}}$ and $\bar{\boldsymbol{M}}$ are the lower and upper bounds of the mass matrix, respectively.
We define the nominal value matrices as

$$\boldsymbol{K}^c = (\underline{\boldsymbol{K}} + \bar{\boldsymbol{K}})/2, \quad \boldsymbol{M}^c = (\underline{\boldsymbol{M}} + \bar{\boldsymbol{M}})/2 \tag{3}$$

and the deviation amplitude matrices as

$$\Delta\boldsymbol{K} = (\bar{\boldsymbol{K}} - \underline{\boldsymbol{K}})/2, \quad \Delta\boldsymbol{M} = (\bar{\boldsymbol{M}} - \underline{\boldsymbol{M}})/2 \tag{4}$$

By means of interval matrix notation, the inequalities in Eq. (2) can be noted as

$$\boldsymbol{K} \in \boldsymbol{K}^I = [\boldsymbol{K}^c - \Delta\boldsymbol{K} \ , \boldsymbol{K}^c + \Delta\boldsymbol{K}], \quad \boldsymbol{M} \in \boldsymbol{M}^I = [\boldsymbol{M}^c - \Delta\boldsymbol{M} \ , \boldsymbol{M}^c + \Delta\boldsymbol{M}] \tag{5}$$

where $\boldsymbol{K}^I$ and $\boldsymbol{M}^I$ are the interval matrices.

When the stiffness matrix and mass matrix are assigned their nominal values, the nominal eigenvalues and the nominal eigenvectors can be obtained as

$$\boldsymbol{K}^c \boldsymbol{u}_i^c = \lambda_i^c \boldsymbol{M}^c \boldsymbol{u}_i^c, \ (i = 1, 2, 3, ..., n) \tag{6}$$

where λ_i^c is the ith order nominal eigenvalue and $\boldsymbol{u}_i^c$ is the ith order nominal eigenvector.

It can be inferred from Eq. (1) that if the matrices ***K***, ***M*** involve uncertainties, the eigenvalues and the eigenvectors are also uncertain. The eigenvalues with uncertainties can also be expressed with interval notation as follows:

$$\lambda_i = \lambda_i^c + \Delta\lambda_i \ , \quad (i = 1, 2, 3, ..., n) \tag{7}$$

where $\Delta\lambda_i$ represents the uncertain part of the *i*th order eigenvalue λ_i .

An interval-based perturbation method is a commonly used method to deal with problems related to uncertainties, and it has already been introduced for the prediction of lower and upper bounds of eigenvalues for structures with uncertain parameters. The uncertain part of the eigenvalues can be calculated using the perturbation method as follows:

$$\Delta\lambda_i = (\boldsymbol{u}_i^c)^T \cdot \Delta\boldsymbol{K} \cdot \boldsymbol{u}_i^c - \lambda_i^c \cdot (\boldsymbol{u}_i^c)^T \cdot \Delta\boldsymbol{M} \cdot \boldsymbol{u}_i^c, \ (i = 1, 2, 3, ..., n) \tag{8}$$

and the interval eigenvalue λ_i^I can be expressed as:

$$\lambda_i^I = [\lambda_i^c - \Delta\lambda_i \, , \, \lambda_i^c + \Delta\lambda_i \,], \quad (i = 1, 2, 3, ..., n) \tag{9}$$

For an interval parameter $a^I = [\underline{a}, \, \overline{a}] = [a^c - \Delta a, \; a^c + \Delta a]$, the percent change β is defined as:

$$\beta = \Delta a / a^c \tag{10}$$

The percent change β is also represented as the uncertainty factor, and it designates the extent to which an uncertain parameter changes. With the nominal value remaining constant, the bigger β is, the wider range the interval parameter varies within.

The perturbation method is convenient for implementation and is widely used in uncertainty fields. However, as the percent change β increases, the difference between the results obtained using the perturbation method and the exact solutions becomes significant. The bigger β is, the bigger the errors are. When dealing with uncertainty problems with large percent changes, the results may be wildly inaccurate. This is a prominent problem for the application of the perturbation method. To obtain the interval natural frequencies with a high level of accuracy, we employ a BP neural network.

3. Mechanism of a BP Neural Network

An artificial neural network can simulate the human thought process and holds a strong non-linear mapping ability. The BP neural network, namely the error back-propagation neural network, is the most maturely studied neural network and has been successfully employed in many research fields. In this study, it is used to predict the bounds of natural frequencies for structures with interval parameters. The BP neural network is a multilayer forward, one-way transmission network, and it consists of three layers: input layer, hidden layer, and output layer. Firstly, the known parameters are inputted to the input layer. Subsequently, each neuron in the hidden layer produces an activation signal to the output layer. Finally, the neurons in the output layer produce a result based on the linear combination of the activations passed from the hidden layer. The topology structure of a BP neural network is presented in Figure 1.

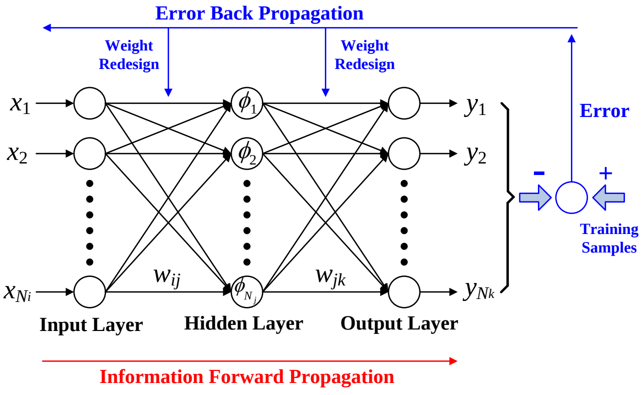


Figure 1. Topology structure of a BP neural network.

After training, the BP neural network can establish the correlation between the input data and output data, expressed as

$$f:\ \boldsymbol{x} \to \boldsymbol{y} \tag{11}$$

where f is the mathematical relationship function, $\boldsymbol{x} = [x_1, x_2, \ldots\ldots, x_{N_i}]$ is the N_i-dimensional input vector, and $\boldsymbol{y} = [y_1, y_2, \ldots\ldots, y_{N_k}]$ is the N_k-dimensional output vector. In between the input layer together with the output layer, there is a hidden layer. The hidden layer consists of multiple nodes, which are also called neurons.

The input layer firstly transfers the input data to the hidden layer. Each neuron in the hidden layer implements an excitation function. The excitation function in the neuron of the hidden layer usually uses a sigmoid function, namely:

$$\phi_j(u) = 1 \big/ (1 + e^{-u}),\ \ (j = 1, 2, \ldots, N_j) \tag{12}$$

where N_j denotes the number of neurons of the hidden layer.

The number of neurons in the hidden layer affects the performance of BP neural networks. However, so far, there is no general rule to determine this number. The number of neurons in the hidden layer can be determined using the following relationship:

$$N_j = \log_2 N_i \tag{13}$$

where N_i represents the number of input variables. In most situations, the number of neurons in the hidden layer obtained using Eq. (13) can satisfy the accuracy requirement. If the accuracy requirement is not satisfied, we may increase the number of neurons in the hidden layer.

The input of the j^{th} neuron in the hidden layer is a_j, expressed as:

$$a_j = \sum_{i=1}^{N_i} (w_{ij} x_i - \theta_j),\ (i = 1, 2, \ldots, N_i\ ,\ j = 1, 2, \ldots, N_j) \tag{14}$$

where w_{ij} is the weighting factor connecting the i^{th} input layer neuron and the j^{th} hidden layer neuron, and θ_j is the threshold value of the j^{th} hidden layer neuron.

The output of the j^{th} neuron in the hidden layer is b_j, expressed as:

$$b_j = \phi_j(a_j),\ (j = 1, 2, \ldots, N_j) \tag{15}$$

where ϕ_j is the excitation function, as given in Eq. (12).

Now calculate the output of the k^{th} neuron in the output layer, expressed as:

$$y_k = \sum_{j=1}^{N_j} (w_{jk} b_j - \theta_k), \ (j = 1, 2, ..., N_j, \ k = 1, 2, ..., N_k) \tag{16}$$

where w_{jk} is the weight corresponding to the connection between the *j*th hidden layer neuron and the *k*th output layer neuron, and θ_k is the threshold value of the *k*th output layer neuron. The weights and threshold values reflect the contribution of a certain hidden layer neuron to an individual output.

A BP neural network can establish a certain mapping function from an N_i-dimensional space to an N_k-dimensional space. A set of input data, for which the corresponding outputs are already known, should be provided. These data are called the training samples. The weights and threshold values of the network can be determined after the training process. The training process of the network is essentially a minimization process of the error function. When a sample, indicated by s, is inputted to the BP neural network, an output vector can be calculated. The error function corresponding to this sample is the sum of the squared error of each output in the output layer, expressed as:

$$E^{(s)} = \frac{1}{2} \sum_{k=1}^{N_k} (y_k^{(s)} - d_k^{(s)})^2 \tag{17}$$

where $\boldsymbol{y}^{(s)} = [y_1^{(s)}, y_2^{(s)}, ..., y_q^{(s)}]$ is the network output vector corresponding to the *s*th sample, and $\boldsymbol{d}^{(s)} = [d_1^{(s)}, d_2^{(s)}, ..., d_q^{(s)}]$ is the expected output vector corresponding to the *s*th sample.

Suppose that there are N_s training samples in total. After all the samples are inputted to the BP neural network, the network error function can be obtained as follows:

$$E_{network} = \sum_{s=1}^{N_s} E^{(s)} = \frac{1}{2} \sum_{s=1}^{N_s} \sum_{k=1}^{N_k} (y_k^{(s)} - d_k^{(s)})^2 \tag{18}$$

where $E_{network}$ is called the network error function.

The values of the weights and thresholds are first randomly assigned, and then optimized using iteration procedures. This approach guarantees the robustness of the network for arbitrary inputs. The error corresponding to the *k*th output at the *t*th iteration step is defined as $e_k^{(t)}$, expressed as:

$$e_k^{(t)} = d_k - y_k^{(t)}, \ \ (k = 1, 2, ..., N_k) \tag{19}$$

where d_k and $y_k^{(t)}$ are the desired output and network output for the *k*th neuron in the output layer, respectively; the superscript (*t*) represents the *t*th iteration step.

The iteration equations for the weights which connect two adjacent layers at the (*t*+1)th iteration step are expressed as follows:

$$\begin{cases} w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta b_j^{(t)}(1-b_j^{(t)})x_i \cdot (\sum_{k=1}^{N_k} w_{jk}^{(t)} e_k^{(t)}) \\ w_{jk}^{(t+1)} = w_{jk}^{(t)} + \eta b_j^{(t)} e_k^{(t)} \end{cases} \tag{20}$$

where η is called the learning rate, and $\eta \in (0,1)$; $i = 1, 2, ..., N_i$; $j = 1, 2, ..., N_j$; $k = 1, 2, ..., N_k$. The iteration equations for the threshold values in the hidden layer and the output layer at the (t+1)th iteration step are expressed as follows:

$$\begin{cases} \theta_j^{(t+1)} = \theta_j^{(t)} + \eta b_j^{(t)}(1-b_j^{(t)}) \cdot (\sum_{k=1}^{N_k} w_{jk}^{(t)} e_k^{(t)}) \\ \theta_k^{(t+1)} = \theta_k^{(t)} + e_k^{(t)} \end{cases} \tag{21}$$

where η is also the learning rate; $j = 1, 2, ..., N_j$ and $k = 1, 2, ..., N_k$.

When the network error function $E_{network}$ in Eq. (18) is smaller than a specified small quantity e_0, (for example, $e_0 = 10^{-4}$), one can assume that the iterations for the network training have terminated. At this stage, the values of the weights and thresholds are assumed to have converged. The termination condition can be expressed as:

$$E_{network} < e_0 \tag{22}$$

When the termination condition is satisfied, the training process is completed. After the training process, the BP neural network is assumed to have successfully established the correct functional relationship between the inputs and the outputs. At this stage, the parameters of this network are determined, and the network can be used for many applications, such as functional approximation, image processing and pattern recognition. In this study, the BP neural network is adopted to evaluate the bounds of natural frequencies for structures with interval parameters.

4. BP Neural Network-based Method for Prediction on Interval Natural Frequencies

In this study, we proposed a BP neural network-based method to calculate the lower and upper bounds of natural frequencies for structures with interval parameters. The procedures of implementing this method are described in the subsequent sections.

4.1. Establishment of FEM Model

The finite element method (FEM) is a well-accepted computational tool for assessing the mechanical properties of structures. In this study, we use a FEM model to obtain the natural frequencies of a structure. The stiffness matrix ***K*** and mass matrix ***M*** can be obtained using the FEM model, and Eq. (1) can be adopted to calculate the natural frequencies. Usually, the lower order natural frequencies have greater effects on the dynamic behaviors than the higher order ones, and our study focuses on the lower order natural frequencies.

4.2. Designation of Inputs and Outputs

The purpose of this paper is to reveal the inherent law between uncertain parameters and natural frequencies. Owing to variances of the environment, measurement errors, or manufacturing inaccuracy, the structural parameters will definitely exhibit some uncertainties. The uncertain parameters are listed as a vector as follows:

$$\boldsymbol{a} = [a_1, a_2, ..., a_{N_a}]^T \tag{23}$$

where N_a represents the number of uncertain parameters. The uncertain parameters can be the Young's modulus, the mass density, or Poisson's ratio. Based on the interval analysis, the vector $\boldsymbol{a}$ is assumed to vary within an interval vector $\boldsymbol{a}^I$, i.e.,

$$\boldsymbol{a} \in \boldsymbol{a}^I = [a_1^I, a_2^I, ..., a_{N_a}^I]^T \tag{24}$$

or in the element form as follows:

$$a_l \in a_l^I = [\underline{a}_l, \bar{a}_l], \ (l = 1, 2, ..., N_a) \tag{25}$$

where a_l^I is an interval number, and $\underline{a}_l$, $\bar{a}_l$ denote the lower and upper bounds of a_l^{I}.

The uncertain parameters are designated as the inputs of the BP neural network, expressed as:

$$\boldsymbol{x} = \boldsymbol{a} \tag{26}$$

where $\boldsymbol{x}$ is the input vector expressed in Eq. (11). In addition, we have:

$$N_i = N_a \tag{27}$$

where N_i is the neuron amount in the input layer of a BP neural network.

The natural frequencies of concern are listed as a vector as follows:

$$\boldsymbol{f} = [f_1, f_2, ..., f_{N_f}]^T \tag{28}$$

where N_f is the order of frequencies we care about. In our following numerical example, $N_f = 4$, i.e., our study focuses on the first 4 natural frequencies.

The natural frequencies of concern are designated as the inputs of the BP neural network, expressed as:

$$\boldsymbol{y} = \boldsymbol{f} \tag{29}$$

where $\boldsymbol{y}$ is the output vector expressed in Eq. (11). In addition, we have:

$$N_k = N_f \tag{30}$$

where N_k is the number of neurons in the output layer of a BP neural network.

4.3. Generation of Training Samples

Neural networks are mathematical models which are strictly data-driven. The prediction accuracy of a BP neural network is highly dependent on the training samples, which are designated to train the network to reveal the correct mathematical relationship between the inputs and outputs. Thus, the selection of training samples plays a significant role in the BP neural network establishment. The training samples are required to cover the design variable space uniformly. Usually, samples are generated using the design of experiments (DOE), such as orthogonal arrays design, full factorial design, Latin hypercube design, and optimal Latin hypercube design. In this study, the training samples are generated by an optimal Latin hypercube design, which is an improvement over the classical Latin hypercube design. The optimal Latin hypercube design distributes the sample data over the design variable space in a uniform manner and can avoid the occurrence of data clustering.

There is no rule for determining the appropriate number of training samples for a BP neural network. It is preferable to analyse as many samples as possible to obtain an accurate network. However, too many samples mean a heavy computational load, and often, it is unnecessary. Therefore, the number of samples should be determined by balancing accuracy and efficiency. The number of training samples is firstly assigned with an initial value N_{s0}, then if this number is not enough, we may enhance the accuracy by applying an increment ΔN_s . The numbers of different sets of training samples are expressed as follows:

$$N_1 = N_{s_0} + \Delta N_s \text{, } N_2 = N_{s_0} + 2 \times \Delta N_s , \cdots\cdots , N_s = N_{s_0} + s \times \Delta N_s \tag{31}$$

where N_s represents the number of the s^{th} set of training samples. In the following sub-section 4.7, the termination condition for N_s is proposed. When the termination condition is satisfied, the number of training samples can be assumed to be large enough.

The s^{th} set of training samples can be expressed as follows

$$S_{input}^{(N_s)} = \{(\boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, ..., \boldsymbol{a}^{(i)}, ..., \boldsymbol{a}^{(N_s)}) \,|\, \boldsymbol{a}^{(i)} \in \boldsymbol{a}^{I}, 1 \le i \le N_s \} \tag{32}$$

where $S_{input}^{(N_s)}$ represents the s^{th} set of training samples; $\boldsymbol{a}^{(i)}$ is the i^{th} input vector and varies within the interval $\boldsymbol{a}^{I}$; the samples $\boldsymbol{a}^{(i)}$ $(1 \le i \le N_s)$ are distributed by the optimal Latin hypercube design; N_s is the number of training samples.

The desired outputs corresponding to the training samples should be obtained using the FEM analysis, expressed as:

$$S_{output}^{(N_s)} = \{ \boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, ... \boldsymbol{f}^{(i)} ..., \boldsymbol{f}^{(N_s)} \,|\, \boldsymbol{f}^{(i)} = [f_1^{(i)}, f_2^{(i)}, ..., f_{N_f}^{(i)}], 1 \le i \le N_s \} \tag{33}$$

where $S_{output}^{(N_s)}$ represents the desired outputs corresponding to N_s training samples; $\boldsymbol{f}^{(i)}$ is the i^{th} output natural frequency vector; N_f is the order of frequencies we care about.

4.4. Establishment of BP Neural Network

The training samples, presented in Eq. (32), and their desired outputs, presented in Eq. (33), are employed to create the BP neural network. The correctness of the outputs is based on the topology of the network. In Section 3, the procedures of how to determine the values of weights and thresholds are presented, and the process of determining these parameters is called the training process. After the BP neural network is successfully trained, the mapping function between the input data and the output data can be obtained. At this stage, the BP neural network corresponding to N_s training samples is established, and we can retrieve the relationship between the uncertain parameters and the natural frequencies. To a certain extent, this implicit relationship becomes explicit by utilizing the BP neural network. Subsequently, the established BP neural network can be employed to predict the lower and upper bounds of natural frequencies for structures with interval parameters.

4.5. Generation of Testing Samples

The established BP neural network reveals the inherent law between the uncertain parameters and the natural frequencies. However, our direction is to calculate the lower and upper bounds of natural frequencies. To achieve this goal, we need to generate testing samples. The purpose of generating testing samples is to set them as inputs to the BP neural network, and we observe the maximum and minimum from the corresponding outputs. As for the sampling distribution, we also apply the technique of optimal Latin hypercube design, which distributes the sample data over the design variable space uniformly and avoids the data clustering problem. The number of testing samples is much larger than that of the training samples, because we want these samples to fill the design variable space as thoroughly as possible.

The set of testing samples can be expressed as follows:

$$T_{input} = \{(\boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, ..., \boldsymbol{a}^{(i)}, ..., \boldsymbol{a}^{(N_T)}) \mid \boldsymbol{a}^{(i)} \in \boldsymbol{a}^I, 1 \le i \le N_T\} \tag{34}$$

where T_{input} represents the set of testing samples; $\boldsymbol{a}^{(i)}$ is the ith input vector and varies within the interval $\boldsymbol{a}^I$; the samples $\boldsymbol{a}^{(i)}$ $(1 \le i \le N_T)$ are distributed by the optimal Latin hypercube design; N_T denotes the number of testing samples and in our study, $N_T = 10^6$.

4.6. Calculation of Lower and Upper Bounds

At this stage, the BP neural network corresponding to Ns training samples has been established, and the testing samples in Eq. (34) can be inputted into the network. The outputs corresponding to these testing samples are called the testing outputs, expressed as:

$$T_{output}^{(N_s)} = \{\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}, ...\boldsymbol{f}^{(i)}, ..., \boldsymbol{f}^{(N_T)} \mid \boldsymbol{f}^{(i)} = [f_1^{(i)}, f_2^{(i)}, ..., f_{N_f}^{(i)}], 1 \le i \le N_T\} \tag{35}$$

where $T_{output}^{(N_s)}$ represents the set of testing outputs corresponding to N_s training samples; $\boldsymbol{f}^{(i)}$ is the ith output natural frequency vector; N_f is the order of frequencies we care about.

The upper bound vector of the natural frequencies is the maximum of the testing output set $T_{output}^{(N_s)}$, expressed as:

$$\overline{\boldsymbol{f}}^{(N_s)} = \max\{\boldsymbol{f}^{(i)} \mid \boldsymbol{f}^{(i)} \in T_{output}^{(N_s)}, 1 \le i \le \mathrm{N}_T\} \tag{36}$$

or in the element form expressed as:

$$\overline{\boldsymbol{f}}^{(N_s)} = [\overline{f}_1^{(N_s)}, \overline{f}_2^{(N_s)}, ..., \overline{f}_{N_f}^{(N_s)}] \tag{37}$$

where $\overline{\boldsymbol{f}}^{(N_s)}$ is the upper bound vector corresponding to N_s training samples.

Similarly, the lower bound vector of the natural frequencies is the minimum of the testing output set $T_{output}^{(N_s)}$, expressed as:

$$\underline{\boldsymbol{f}}^{(N_s)} = \min\{\boldsymbol{f}^{(i)} \mid \boldsymbol{f}^{(i)} \in T_{output}^{(N_s)}, 1 \le i \le \mathrm{N}_T\} \tag{38}$$

or in the element form expressed as:

$$\underline{\boldsymbol{f}}^{(N_s)} = [\underline{f}_1^{(N_s)}, \underline{f}_2^{(N_s)}, ..., \underline{f}_{N_f}^{(N_s)}] \tag{39}$$

where $\underline{\boldsymbol{f}}^{(N_s)}$ is the upper bound vector corresponding to N_s training samples.

Now, the lower and upper bounds of natural frequencies are obtained using the established BP neural network, and presented in Eq. (36) and Eq. (38). It should be noted that the bound results obtained here correspond to Ns training samples, and if the bound results do not satisfy the termination condition in sub-section 4.7, the number of training samples should be increased. Besides, the lower and upper bounds should be calculated again with the increased training samples.

4.7. Termination Condition

The number of training samples is of tremendous significance for the BP neural network establishment, because insufficient training samples may lead to inaccurate outputs, whereas, excessive training samples consume too many computing resources. It is crucial to determine a proper number of training samples. In this study, the number of training samples is presented in Eq. (31), and we put forward the termination condition below to check whether the number is enough or not. This is a feasible approach to strike a good balance between the demands of accuracy and efficiency for determining the number of training samples.

The lower and upper bounds of natural frequencies corresponding to N_s training samples are $\underline{\boldsymbol{f}}^{(N_s)}$ and $\overline{\boldsymbol{f}}^{(N_s)}$, respectively, and the lower and upper bounds of natural frequencies corresponding to $N_{s\text{-}1}$ training samples are $\underline{\boldsymbol{f}}^{(N_{s-1})}$ and $\overline{\boldsymbol{f}}^{(N_{s-1})}$, respectively. We propose a bound error function based on the adjacent sets of bounds obtained using different training samples, expressed as:

$$E_{bound} = \| \underline{\boldsymbol{f}}^{(N_s)} - \underline{\boldsymbol{f}}^{(N_{s-1})} \|^2 + \| \overline{\boldsymbol{f}}^{(N_s)} - \overline{\boldsymbol{f}}^{(N_{s-1})} \|^2 \tag{40}$$

where E_{bound} is the bound error function; the symbol || || is the Euclidean norm of a vector; N_s and $N_{s\text{-}1}$ are the numbers of different training samples, presented in Eq. (31).

The bound error function E_{bound} can also be expressed in the element form as follows:

$$\begin{aligned} E_{bound} &= [\underline{f}_1^{(N_s)} - \underline{f}_1^{(N_{s-1})}]^2 + [\underline{f}_2^{(N_s)} - \underline{f}_2^{(N_{s-1})}]^2 + \ldots\ldots + [\underline{f}_{N_f}^{(N_s)} - \underline{f}_{N_f}^{(N_{s-1})}]^2 \\ &+ [\overline{f}_1^{(N_s)} - \overline{f}_1^{(N_{s-1})}]^2 + [\overline{f}_2^{(N_s)} - \overline{f}_2^{(N_{s-1})}]^2 + \ldots\ldots + [\overline{f}_{N_f}^{(N_s)} - \overline{f}_{N_f}^{(N_{s-1})}]^2 \end{aligned} \qquad (41)$$

where N_f is the order of natural frequencies of concern.

When the bound error function E_{bound} in Eq. (40) is smaller than a specified small quantity ε_0, (for example, $\varepsilon_0 = 0.1$), we can assume that the value N_s is adequate for network establishment, and we do not have to increase the number of training samples anymore. This termination condition can be expressed as follows:

$$E_{bound} < \varepsilon \qquad (42)$$

and at this stage, the calculation of lower and upper bounds of natural frequencies is assumed to be completed.

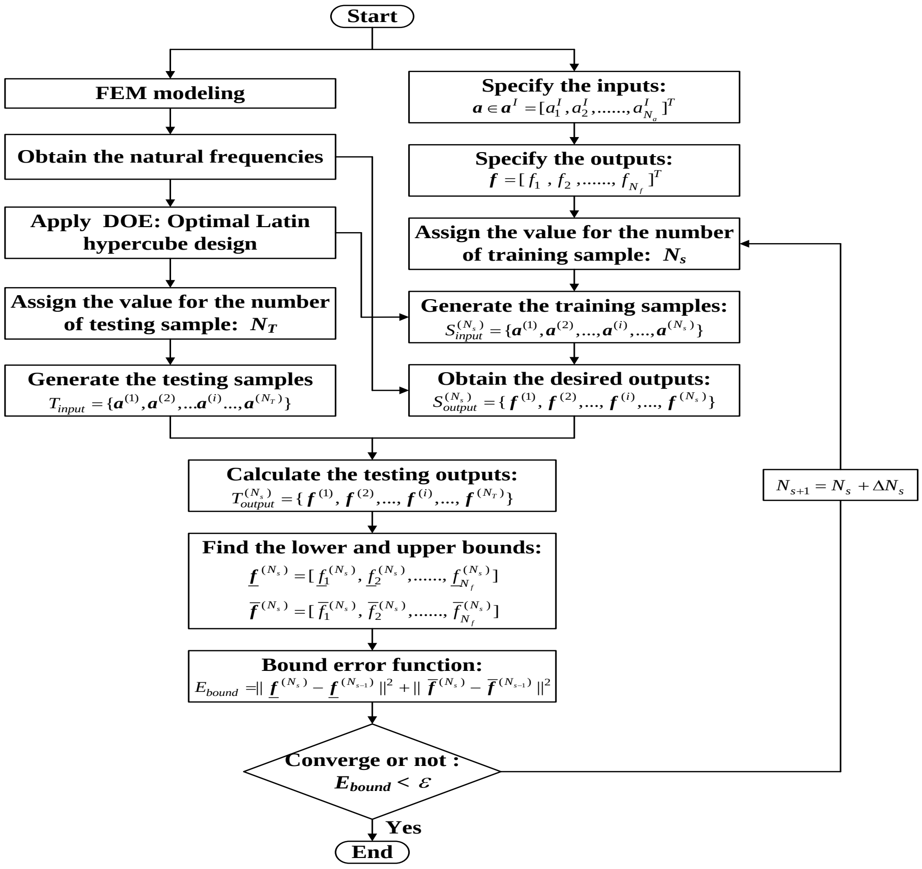

Start
FEM modeling
Obtain the natural frequencies
Apply DOE: Optimal Latin hypercube design
Assign the value for the number of testing sample: N_T
Generate the testing samples $T_{input} = \{a^{(1)}, a^{(2)}, \ldots a^{(i)} \ldots, a^{(N_T)}\}$
Specify the inputs: $a \in a^I = [a_1^I, a_2^I, \ldots\ldots, a_{N_a}^I]^T$
Specify the outputs: $f = [f_1, f_2, \ldots\ldots, f_{N_f}]^T$
Assign the value for the number of training sample: N_s
Generate the training samples: $S_{input}^{(N_s)} = \{a^{(1)}, a^{(2)}, \ldots, a^{(i)}, \ldots, a^{(N_s)}\}$
Obtain the desired outputs: $S_{output}^{(N_s)} = \{f^{(1)}, f^{(2)}, \ldots, f^{(i)}, \ldots, f^{(N_s)}\}$
Calculate the testing outputs: $T_{output}^{(N_s)} = \{f^{(1)}, f^{(2)}, \ldots, f^{(i)}, \ldots, f^{(N_T)}\}$
Find the lower and upper bounds: $\underline{f}^{(N_s)} = [\underline{f}_1^{(N_s)}, \underline{f}_2^{(N_s)}, \ldots\ldots, \underline{f}_{N_f}^{(N_s)}]$ $\overline{f}^{(N_s)} = [\overline{f}_1^{(N_s)}, \overline{f}_2^{(N_s)}, \ldots\ldots, \overline{f}_{N_f}^{(N_s)}]$
Bound error function: $E_{bound} = \| \underline{f}^{(N_s)} - \underline{f}^{(N_{s-1})} \|^2 + \| \overline{f}^{(N_s)} - \overline{f}^{(N_{s-1})} \|^2$
Converge or not : $E_{bound} < \varepsilon$
$N_{s+1} = N_s + \Delta N_s$
Yes
End

Figure 2. Flowchart of the proposed BP neural network-based method.

If the termination condition is not satisfied, the number of training samples should be increased, expressed as:

$$N_{s+1} = N_s + \Delta N_s \tag{43}$$

where N_{s+1} correspond to the $(s+1)^{\text{th}}$ set of training samples, and the BP neural network should be re-established. In addition, the lower and upper bounds of natural frequencies should be re-calculated.

Now, the BP neural network based method to predict the lower and upper bounds of natural frequencies for structures with uncertain-but-bounded parameters has been thoroughly put forward, and in the next section, a numerical example is tested to demonstrate the feasibility of the proposed method. To illustrate the procedures of the proposed method, a flowchart is presented in Figure 2.

5. NUMERICAL EXAMPLE

In order to demonstrate the feasibility of the proposed method to predict the interval of natural frequencies of structures with interval parameters, the following numerical example is presented. The accuracy of the proposed method is compared with that of the perturbation method. The lower and upper bounds calculated using the Monte Carlo simulation are regarded as reference solutions. The Monte Carlo simulation obtains the distribution of responses through an extremely large number of samples and is very time-consuming. Thus, the Monte Carlo simulation is always used to verify the accuracy of other methods; however, it is not convenient or suitable for engineering applications.

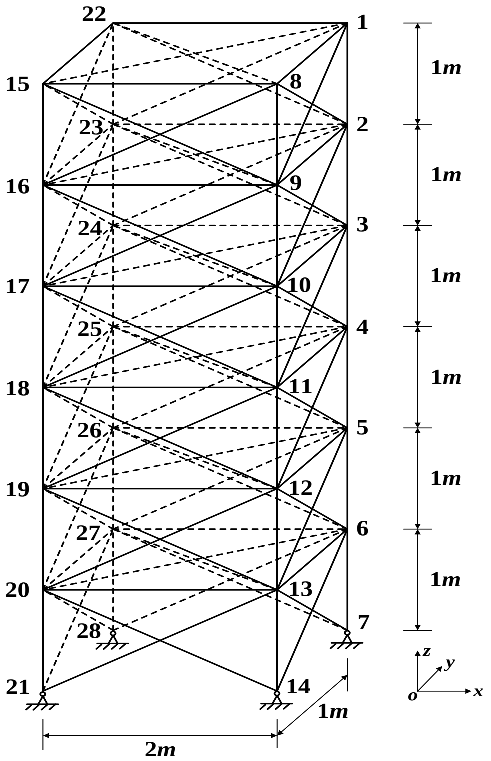


Figure 3. A 108-bar space truss structure

Consider a 108-bar space truss structure. Figure 3 indicates the dimensions and boundary conditions of the truss. The cross-sectional area of each rod is $A = 1.0 \times 10^{-4}\ \text{m}^2$. Young's modulus of the material is $E = 210\ \text{GPa}$, the mass density is $\rho = 7900\ \text{kg/m}^3$, and Poisson's ratio is $\nu = 0.3$. In FEM analysis, each bar is regarded as a rod element. So, there are 108 elements in total. When the structural parameters are assigned their nominal values, we can obtain the nominal natural frequencies by using the existing FEM theory. The first 4 nominal natural frequencies are listed in Table 1.

Table 1. First 4 nominal natural frequencies (Unit: Hz)

Order	f_1	f_2	f_3	f_4
Value	13.6241	25.6548	49.4426	69.4558

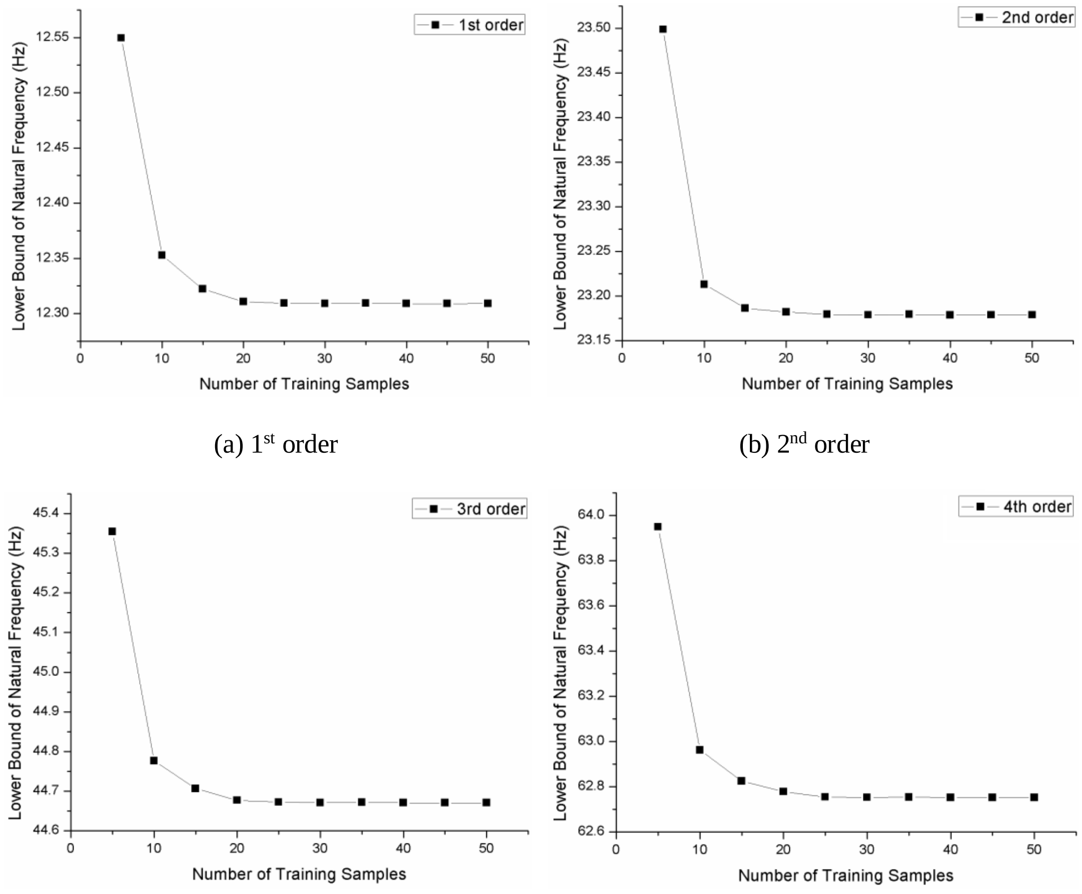
Owing to variances of the environment, manufacturing inaccuracy, or measurement errors, the parameters exhibit some uncertainties. In this example, Young's modulus and the mass density are considered to be the uncertain-but-bounded parameters. Young's modulus is assumed to vary within the interval $E^I = [1-\beta,\ 1+\beta] \cdot E^c$, where $E^c = 210\ \text{GPa}$; the mass density is assumed to vary within the interval $\rho^I = [1-\beta,\ 1+\beta] \cdot \rho^c$, where $\rho^c = 7900\ \text{kg/m}^3$; β is the percent change or also called the uncertainty factor, and in this example it is assigned the following values: β = 2%, 4%, 6%, 8%, 10%.

We employ the proposed method, the perturbation method and the Monte Carlo simulation to calculate the lower and upper bounds of the first 4 natural frequencies. The training samples play a significant role in neural network establishment. In this example, the sample data for training are generated by the optimal Latin hypercube design, which distributes the sample data over the design variable space in a uniform manner and can avoid the occurrence of data clustering. To study the effect of the sample number on BP neural network, we use different numbers of samples for training.

When $\beta = 10\%$, the lower bounds of natural frequencies obtained using the BP neural network corresponding to different numbers of training samples are listed in Table 2 and plotted in Figure 4; the upper bounds of natural frequencies obtained using the BP neural network corresponding to different numbers of training samples are listed in Table 3 and plotted in Figure 5. It can be observed that when the number of samples is greater than 40, the difference between the bound results of the two adjacent sample sets is smaller than 0.001. At this stage, the results can be assumed to have converged.

Table 2. Lower bounds of natural frequencies (Unit: Hz) obtained using the BP neural network with respect to different numbers of training samples (β=10%)

Number of Training Samples	$\underline{f}_1$	$\underline{f}_2$	$\underline{f}_3$	$\underline{f}_4$
5	12.5496	23.4988	45.3548	63.9491
10	12.3528	23.2130	44.7763	62.9616
15	12.3223	23.1863	44.7066	62.8249
20	12.3109	23.1822	44.6772	62.7784
25	12.3095	23.1795	44.6721	62.7543
30	12.3092	23.1789	44.6709	62.7527
35	12.3094	23.1794	44.6718	62.7540
40	12.3091	23.1787	44.6706	62.7522
45	12.3091	23.1788	44.6707	62.7524
50	12.3091	23.1788	44.6707	62.7524



(a) 1st order

(b) 2nd order

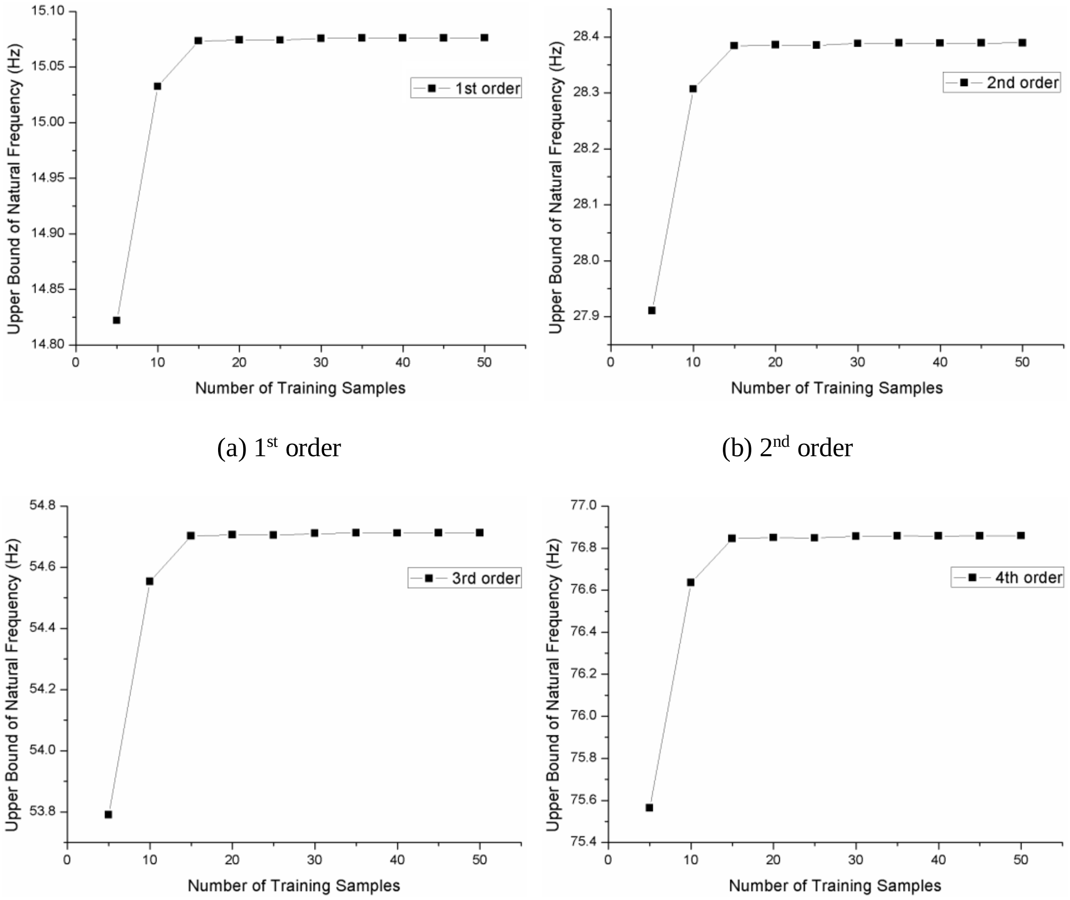
(c) 3rd order

(d) 4th order

Figure 4. Lower bounds of natural frequencies (Unit: Hz) obtained using the BP neural network with respect to different numbers of training samples (β=10%)

Table 3. Upper bounds of natural frequencies (Unit: Hz) obtained using the BP neural network with respect to different numbers of training samples (β=10%)

Number of Training Samples	$\overline{f}_1$	$\overline{f}_2$	$\overline{f}_3$	$\overline{f}_4$
5	14.8221	27.9108	53.7903	75.5634
10	15.0326	28.3070	54.5539	76.6361
15	15.0736	28.3843	54.7028	76.8453
20	15.0744	28.3859	54.7059	76.8496
25	15.0742	28.3854	54.7050	76.8483
30	15.0758	28.3884	54.7107	76.8563
35	15.0762	28.3892	54.7122	76.8585
40	15.0761	28.3890	54.7119	76.8580
45	15.0762	28.3892	54.7122	76.8585
50	15.0763	28.3894	54.7127	76.8592



(a) 1st order

(b) 2nd order

(c) 3rd order

(d) 4th order

Figure 5. Upper bounds of natural frequencies (Unit: Hz) obtained using the BP neural network with respect to different numbers of training samples (β=10%)

The lower and upper bounds of the first 4 natural frequencies obtained using the proposed method, the perturbation method and the Monte Carlo simulation with respect to different values of percent change β are listed in Tables 4 to 7 and plotted in Figure 6.

Table 4. Lower and upper bounds of the 1st natural frequency with respect to different values of percent change β

β	$\underline{f}_1$			$\overline{f}_1$		
	BP Neural Network	Perturbation Method	Monte Carlo Simulation	BP Neural Network	Perturbation Method	Monte Carlo Simulation
0	13.6241	13.6241	13.6241	13.6241	13.6241	13.6241
0.02	13.3517	13.3497	13.3543	13.9028	13.9048	13.8993
0.04	13.0837	13.0658	13.0896	14.1852	14.2022	14.1804
0.06	12.8211	12.7788	12.8298	14.4762	14.5168	14.4676
0.08	12.5601	12.4750	12.5745	14.7756	14.8491	14.7613
0.10	12.3091	12.1782	12.3234	15.0763	15.1995	15.0620

Table 5. Lower and upper bounds of the 2nd natural frequency with respect to different values of percent change β

β	$\underline{f}_2$			$\overline{f}_2$		
	BP Neural Network	Perturbation Method	Monte Carlo Simulation	BP Neural Network	Perturbation Method	Monte Carlo Simulation
0	25.6548	25.6548	25.6548	25.6548	25.6548	25.6548
0.02	25.1419	25.1365	25.1468	26.1797	26.1834	26.1732
0.04	24.6372	24.6073	24.6484	26.7099	26.7435	26.7024
0.06	24.1429	24.0664	24.1591	27.2664	27.3359	27.2432
0.08	23.6513	23.5131	23.6784	27.8234	27.9615	27.7963
0.10	23.1788	22.9467	23.2057	28.3894	28.6215	28.3625

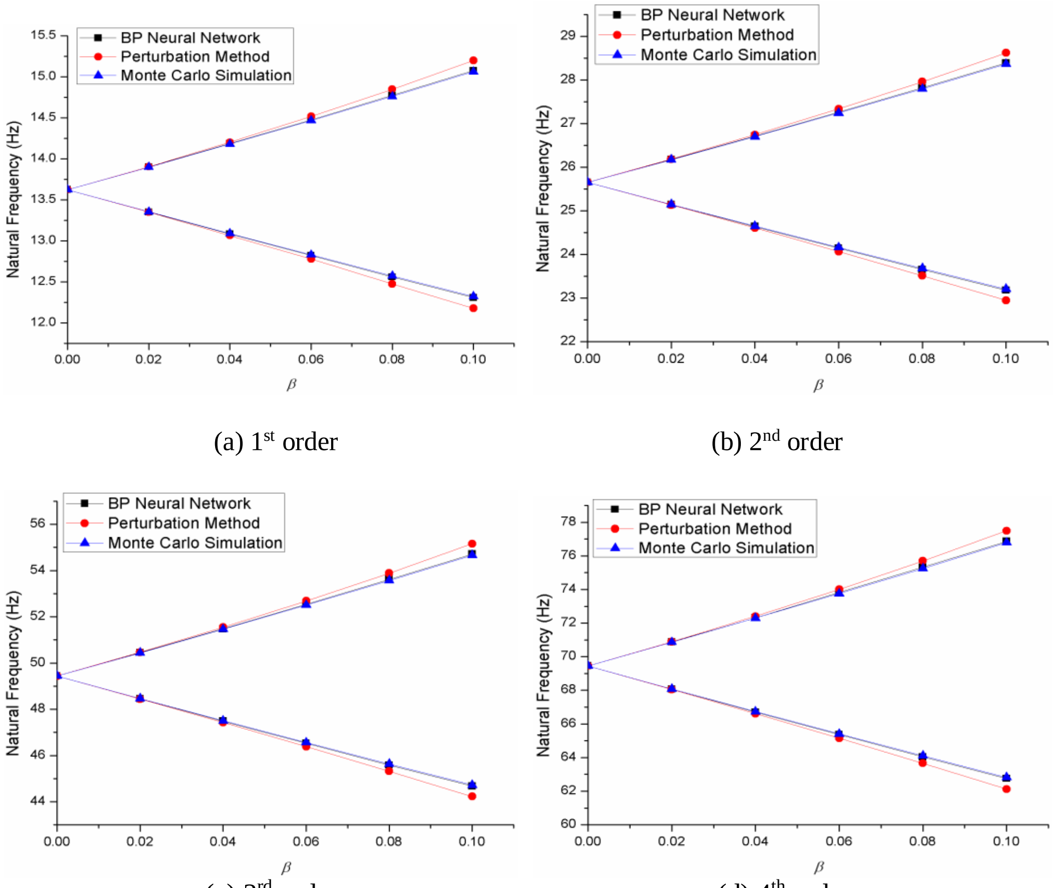
Table 6. Lower and upper bounds of the 3rd natural frequency with respect to different values of percent change β

β	$\underline{f}_3$			$\overline{f}_3$		
	BP Neural Network	Perturbation Method	Monte Carlo Simulation	BP Neural Network	Perturbation Method	Monte Carlo Simulation
0	49.4426	49.4426	49.4426	49.4426	49.4426	49.4426
0.02	48.4540	48.4436	48.4634	50.4540	50.4613	50.4415
0.04	47.4814	47.4236	47.5029	51.4759	51.5407	51.4614
0.06	46.5286	46.3813	46.5599	52.5449	52.6823	52.5037
0.08	45.5813	45.3150	45.6334	53.6218	53.8881	53.5697
0.10	44.6707	44.2233	44.7225	54.7127	55.1601	54.6608

Table 7. Lower and upper bounds of the 4th natural frequency with respect to different values of percent change β

β	$\underline{f}_4$			$\overline{f}_4$		
	BP Neural Network	Perturbation Method	Monte Carlo Simulation	BP Neural Network	Perturbation Method	Monte Carlo Simulation
0	69.4558	69.4558	69.4558	69.4558	69.4558	69.4558
0.02	68.0671	68.0525	68.0803	70.8766	70.8869	70.8591
0.04	66.7007	66.6197	66.7310	72.3122	72.4032	72.2919
0.06	65.3624	65.1554	65.4063	73.8199	74.0069	73.7560
0.08	64.0316	63.6575	64.1048	75.3267	75.7007	75.2534
0.10	62.7524	62.1239	62.8251	76.8592	77.4875	76.7863

The results obtained using the Monte Carlo simulation are referred to as exact solutions. The sampling size of the Monte Carlo simulation is 10^6. It can be observed that when the percent change β is relatively small, there is no fundamental difference between the bounds obtained using the proposed method and the perturbation method. However, as the percent change β increases, the difference between the bounds obtained using the proposed method and the perturbation method is significant. The comparison indicates that the bounds obtained using the proposed method are more accurate than those obtained using the perturbation method.



(a) 1st order

(b) 2nd order

(c) 3rd order

(d) 4th order

Figure 6. Bounds of the first 4 natural frequencies with respect to different values of percent change β

Table 8. Details of the computation environment

Computation Environment	
CPU :	3.40 GHz
Memory :	16 GB
System Type :	64 bit
Number of Processors :	8
Operating System :	Windows 7
Programing Environment :	ANSYS 16.0 APDL; MATLAB R2013a

The computation time of the proposed method, the perturbation method and Monte Carlo simulation are presented. The details of the computation environment for this numerical example are listed in Table 8. When the percent change β=10%, the computation time corresponding to different methods are listed in Table 9. The Monte Carlo simulation requires a large number of FEM computation samplings, which leads to a heavy computation load. Especially when the structure is complicated, even a one-time FEM computation would be time-consuming. The proposed method also requires FEM computation samplings, but the sampling size is much smaller than that of the Monte Carlo simulation. Thus, the proposed method is more applicable for engineering structures than the Monte Carlo simulation.

Table 9. Comparison of the computation time corresponding to different methods in the numerical example when β =10%

Method	BP Neural Network	Perturbation Method	Monte Carlo Simulation
Computation Time	145.39 s	32.12 s	7.79×10^4 s

The computation time of our proposed method is more than that of the perturbation method within an acceptable range, in this example, it is about 113 s longer, but the comparison results in Figure 6 indicate that the accuracy of the proposed method is much higher than that of the perturbation method. The results obtained using the proposed method are in good accordance with those obtained using the Monte Carlo simulation. The proposed method can strike a good balance between the demands of accuracy and efficiency. This is the principal advantage of the proposed method to predict the interval natural frequencies of structures with interval parameters.

6. CONCLUSION

In this study, a BP neural network-based method was proposed to predict the interval natural frequencies of structures with uncertain-but-bounded parameters. The inherent law between the uncertain parameters and the natural frequencies is revealed using a BP neural network. A numerical example is employed to manifest the feasibility of the proposed method. The results indicate that the interval natural frequencies obtained using the proposed method are of higher accuracy than those obtained using the perturbation method, especially when the varying ranges of interval parameters are large. Our purpose is not only to put forward a novel method to predict the interval natural frequencies with a high level of accuracy but also to manifest the enormous potential of BP neural networks for solving uncertain problems in mechanics, such as uncertain static problems, uncertain dynamic problems, and uncertain buckling problems.

REFERENCES

[1] Y. Ben-Haim & I. Elishakoff, (1990) *Convex Models of Uncertainty in Applied Mechanics*, Elsevier, New York.

[2] X.J. Wang & L. Wang, (2011) "Uncertainty quantification and propagation analysis of structures based on the measurement data", *Mathematical and Computer Modelling*, Vol. 54, No. 11, pp2725–2735.

[3] Z. Lv & Z.P. Qiu, (2016) "A direct probabilistic approach to solve state equations for nonlinear systems under random excitation", *Acta Mechanica Sinica*, Vol. 32, No. 5, pp941–958.

[4] S.S. Rao & L. Berke, (1997) "Analysis of uncertain structural systems using interval analysis", *AIAA Journal*, 35, No. 4, pp727–735.

[5] Z.P. Qiu & L. Wang, (2016) "The need for introduction of non-probabilistic interval conceptions into structural analysis and design", *Science China: Physics, Mechanics & Astronomy*, Vol. 59, No. 11, pp114632.

[6] L. Wang, X.J. Wang & Y. Xia, (2014) "Hybrid reliability analysis of structures with multi-source uncertainties", *Acta Mechanica*. Vol. 225, No. 2, pp413–430.

[7] G. Augusti, A. Baratta & F. Casciati, (1984) *Probabilistic Methods in Structural Engineering*, CRC Press, Boca Raton.

[8] I. Elishakoff & J.T.P. Yao, (1983) *Probabilistic Methods in the Theory of Structures*, Wiley Press, New York.

[9] L. Wang, D.L. Liu & Z.P. Qiu, (2017) "A novel method of non-probabilistic reliability-based topology optimization corresponding to continuum structures with unknown but bounded uncertainties", *Computer Methods in Applied Mechanics and Engineering*, Vol. 326, pp573–595.

[10] L. Jaulin, M. Kieffer & O. Didrit, (2001) *Applied Interval Analysis*, Springer, Berlin.

[11] R.E. Moore, (1979) *Methods and Applications of Interval Analysis*, Prentice-Hall, London.

[12] Z.P. Qiu & Z. Lv, (2017) "The vertex solution theorem and its coupled framework for static analysis of structures with interval parameters", *International Journal for Numerical Methods in Engineering*, Vol. 112, No. 7, pp711–736.

[13] L. Wang & X.J. Wang, (2017) "A novel method of Newton iteration-based interval analysis for multidisciplinary systems", *Science China: Physics, Mechanics & Astronomy*, Vol. 60, No. 9, pp094611.

[14] W. Verhaeghe, W. Desmet, D. Vandepitte & D. Moens, (2013) "Interval fields to represent uncertainty on the output side of a static FE analysis", *Computer Methods in Applied Mechanics and Engineering*, Vol. 260, pp50–62.

[15] N. Impollonia & G. Muscolino, (2011) "Interval analysis of structures with uncertain-but-bounded axial stiffness", *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, pp1945–1962.

[16] S.H. Chen & X.W. Yang, (2000) "Interval finite element method for beam structures", *Finite Elements in Analysis and Design*, Vol. 34, pp75–88.

[17] Z. Lv, Z.P. Qiu & Q. Li, (2017) "An interval reduced basis approach and its integrated framework for acoustic response analysis of coupled structural-acoustic system", *Journal of Computational Acoustics*, Vol. 25, No. 3, pp1750009.

[18] S.H. Chen, (2007) *Matrix Perturbation Theory in Structural Dynamic Design*, Science Press, Beijing.

[19] J.C. Chen & B.K. Wade, (1977) "Matrix perturbation for structural dynamics", *AIAA Journal*, Vol. 15, No. 8, pp1095–1100.

[20] C.S. Rudisill, (1974) "Derivatives of eigenvalues and eigenvectors for a general matrix", *AIAA Journal*, Vol. 12, No. 1, pp721–722.

[21] M. Gevrey, I. Dimopoulos & S. Lek, (2003) "Review and comparison of methods to study the contribution of variables in artificial neural network models", *Ecological Modelling*, Vol. 160, pp249–264.

[22] G.N. Xu & Q. Zhang, (2012) "Safety assessment of the general overhead traveling crane metal structure based on BP neural network", *Applied Mechanics and Materials*, Vol. 101, pp15–20.

[23] S.M. Yuan, Z.F. Zhan & Y. Li, (2013) "Optimum design of machine tool structures based on BP neural network and genetic algorithm", *Advanced Materials Research*, vol. 655, pp1291–1295.

[24] G.M. Yang, C.S. Gu, Y. Huang & K. Yang, (2014) "BP neural network integration model research for hydraulic metal structure health diagnosing", *International Journal of Computational Intelligence Systems*, Vol. 7, No. 6, pp1148–1158.

[25] Q. Li, J.Y. Yu, B.C. Mu & X.D. Sun, (2006) "BP neural network prediction of the mechanical properties of porous NiTi shape memory alloy prepared by thermal explosion reaction", *Materials Science and Engineering*, Vol. 419, pp214–217.