Interval and Fuzzy Computing in Neural Network for System Identification Problems

Dissertation submitted in partial fulfillment

of the requirements of the degree of

Doctor of Philosophy

in

Mathematics

by

Deepti Moyi Sahoo

(Roll Number: 512MA601)

based on research carried out

under the supervision of

Prof. Snehashish Chakraverty

August, 2017

Department of Mathematics

National Institute of Technology Rourkela

Department of Mathematics
National Institute of Technology Rourkela

August 2017

Certificate of Examination

Roll Number: *512MA601*
Name: *Deepti Moyi Sahoo*
Title of Dissertation*: Interval and Fuzzy Computing in Neural Network for System Identification Problems*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of *Doctor of Philosophy in Mathematics at National Institute of Technology Rourkela*. We are satisfied with the volume, quality, correctness, and originality of the work.

---------------------------
Co-Supervisor

--------------------------
Snehashish Chakraverty
Principal Supervisor

---------------------------
S. K. Jena
Member (DSC)

--------------------------
Anil Kumar
Member (DSC)

---------------------------
B.K. Ojha
Member (DSC)

--------------------------
Examiner

-------------------------
K.C. Pati
Chairman (DSC)

Mathematics
National Institute of Technology Rourkela

Prof. /Dr. Snehashish Chakraverty
Professor, Department of Mathematics

August, 2017

Supervisor's Certificate

This is to certify that the work presented in this dissertation entitled "*Interval and Fuzzy Computing in Neural Network for System Identification Problems"*, Roll Number 512MA601, is a record of original research carried out by her under my supervision and guidance in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Mathematics*. Neither this dissertation nor any part of it has been submitted for any degree or diploma to any institute or university in India or abroad.

Snehashish Chakraverty
Professor

Dedicated to

My Beloved

Parents and

My Sister

Declaration of Originality

I, Deepti Moyi Sahoo, Roll Number 512MA601 hereby declare that this dissertation entitled "*Interval and Fuzzy Computing in Neural Network for System Identification Problems*" represents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, it contains no material previously published or written by another person, nor any material presented for the award of any other degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the section "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

August, 2017

NIT Rourkela *Deepti Moyi Sahoo*

Acknowledgment

This thesis is a result of the research that has been carried out at National Institute of Technology Rourkela. The work presented in this thesis would not have been possible without the encouragement of numerous people including my well wishers, my friends and my family. I take this opportunity to acknowledge them and extend my sincere gratitude for helping me make this Ph.D. thesis a possibility.

First and foremost, I would like to start with the person who made the biggest difference in my life, my guide, Professor Snehashish Chakraverty. He has been a living role model to me, he takes up new challenges every day and tackles them with all his grit. I am greatly indebted to him for his determination, motivation and inspiration which helped me to learn new things. This work would not have been possible without his guidance, support and encouragement. Under his guidance I successfully overcame many difficulties and learned a lot. His deep insights helped me at various stages of my research. Above all, he offered me so much invaluable advice, patiently supervising and always guiding me in the right direction. I am also thankful to his family especially his wife Mrs. Shewli Chakraborty and daughters Shreyati and Susprihaa for their love and support.

I would like to thank Prof. Animesh Biswas, Director, National Institute of Technology Rourkela for providing facilities in the institute for carrying out this research. I would like to thank the members of my doctoral scrutiny committee for being helpful and generous during the entire course of this work and express my gratitude to all the faculty and staff members of the Department of Mathematics, National Institute of Technology Rourkela for their support.

I take this opportunity to sincerely acknowledge the **Ministry of Earth Sciences (MoES)**, Government of India, New Delhi, for providing financial support under the project entitled Fuzzified and Interval Neural Network Modeling for System Identification of Structure Through the use of Seismic Response Data. This financial assistance as Junior and Senior Research Fellowship buttressed me to perform my work comfortably. I would also like to acknowledge the Department of Earthquake Engineering, IIT Roorkee for the data.

I will forever be thankful to my lab mates Smita, Diptiranjan, Laxmi, Nisha, Karunakar, Dileswar, Sumit and Subrat for their help, encouragement and support during my stay in laboratory and making it a memorable experience in my life.

Last but not the least I would like to express my sincere gratitude to the people who mean world to me, my parents (Mr. Daya Nidhi Sahoo and Mrs. Tilottama Sahoo), my sister Trupti Moyi Sahoo, my brother Deepak Kumar Sahoo and my family members. I don't imagine a life without their love and blessings. I am greatly indebted to my sister for her constant unconditional support and invariable source of inspiration. I would also thank my father-in-law Mr. Duryodhan Sahoo and mother-in-law Mrs. Rukmini Sahoo for their help and motivation. I am thankful to my devoted husband Mr. Soumyaranjan Sahoo for showing faith in me and giving me liberty to choose what I desired. I would like to thank my beloved son Nishit. He is the origin of my happiness. My husband has been a constant source of strength and inspiration. I owe my every achievement to my family.

August, 2017 *Deepti Moyi Sahoo*

NIT Rourkela Roll Number: 512MA60

Abstract

Increase of population and growing of societal and commercial activities with limited land available in a modern city leads to construction up of tall/high-rise buildings. As such, it is important to investigate about the health of the structure after the occurrence of manmade or natural disasters such as earthquakes etc. A direct mathematical expression for parametric study or system identification of these structures is not always possible. Actually System Identification (SI) problems are inverse vibration problems consisting of coupled linear or non-linear differential equations that depend upon the physics of the system. It is also not always possible to get the solutions for these problems by classical methods. Few researchers have used different methods to solve the above mentioned problems. But difficulties are faced very often while finding solution to these problems because inverse problem generally gives non-unique parameter estimates. To overcome these difficulties alternate soft computing techniques such as Artificial Neural Networks (ANNs) are being used by various researchers to handle the above SI problems. It is worth mentioning that traditional neural network methods have inherent advantage because it can model the experimental data (input and output) where good mathematical model is not available. Moreover, inverse problems have been solved by other researchers for deterministic cases only. But while performing experiments it is always not possible to get the data exactly in crisp form. There may be some errors that are due to involvement of human or experiment. Accordingly, those data may actually be in uncertain form and corresponding methodologies need to be developed.

It is an important issue about dealing with variables, parameters or data with uncertain value. There are three classes of uncertain models, which are probabilistic, fuzzy and interval. Recently, fuzzy theory and interval analysis are becoming powerful tools for many applications in recent decades. It is known that interval and fuzzy computations are themselves very complex to handle. Having these in mind one has to develop efficient computational models and algorithms very carefully to handle these uncertain problems.

As said above, in general we may not obtain the corresponding input and output values (experimental) exactly or in crisp form but we may have only uncertain information of the data. Hence, investigations are needed to handle the SI problems where data is available in uncertain form. Identification methods with crisp (exact) data are known and traditional neural network methods have already been used by various researchers. But when the data are in uncertain form then traditional ANN may not be applied. Accordingly, new ANN models need to be developed which may solve the targeted uncertain SI problems. Hence present investigation targets to develop powerful methods of neural network based on interval and fuzzy theory for the analysis and

simulation with respect to the uncertain system identification problems. In this thesis, these uncertain data are assumed as interval and fuzzy numbers. Accordingly, identification methodologies are developed for multistorey shear buildings by proposing new models of Interval Neural Network (INN) and Fuzzy Neural Network (FNN) models which can handle interval and fuzzified data respectively. It may however be noted that the developed methodology not only be important for the mentioned problems but those may very well be used in other application problems too. Few SI problems have been solved in the present thesis using INN and FNN model which are briefly described below.

From initial design parameters (namely stiffness and mass in terms of interval and fuzzy) corresponding design frequencies may be obtained for a given structural problem viz. for a multistorey shear structure. The uncertain (interval/fuzzy) frequencies may then be used to estimate the present structural parameter values by the proposed INN and FNN. Next, the identification has been done using vibration response of the structure subject to ambient vibration with interval/fuzzy initial conditions. Forced vibration with horizontal displacement in interval/fuzzified form has also been used to investigate the identification problem.

Moreover this study involves SI problems of structures (viz. shear buildings) with respect to earthquake data in order to know the health of a structure. It is well known that earthquake data are both positive and negative. The Interval Neural Network and Fuzzy Neural Network model may not handle the data with negative sign due to the complexity in interval and fuzzy computation. As regards, a novel transformation method have been developed to compute response of a structural system by training the model for Indian earthquakes at Chamoli and Uttarkashi using uncertain (interval/fuzzified) ground motion data. The simulation may give an idea about the safety of the structural system in case of future earthquakes. Further a single layer interval and fuzzy neural network based strategy has been proposed for simultaneous identification of the mass, stiffness and damping of uncertain multi-storey shear buildings using series/cluster of neural networks.

It is known that training in MNN and also in INN and FNN are time consuming because these models depend upon the number of nodes in the hidden layer and convergence of the weights during training. As such, single layer Functional Link Neural Network (FLNN) with multi-input and multi-output model has also been proposed to solve the system identification problems for the first time. It is worth mentioning that, single input single output FLNN had been proposed by previous authors. In FLNN, the hidden layer is replaced by a functional expansion block for enhancement of the input patterns using orthogonal polynomials such as Chebyshev, Legendre and Hermite, etc. The computations become more efficient than the traditional or classical multi-layer neural network due to the absence of hidden layer. FLNN has also been used for structural response prediction of multistorey shear buildings subject to earthquake ground motion. It is seen that FLNN can very well predict the structural response of different floors of

multi-storey shear building subject to earthquake data. Comparison of results among Multi layer Neural Network (MNN), Chebyshev Neural Network (ChNN), Legendre Neural Network (LeNN), Hermite Neural Network (HNN) and desired are considered and it is found that Functional Link Neural Network models are more effective and takes less computation time than MNN.

In order to show the reliability, efficacy and powerfulness of INN, FNN and FLNN models variety of problems have been solved here. Finally FLNN is also extended to interval based FLNN which is again proposed for the first time to the best of our knowledge. This model is implemented to estimate the uncertain stiffness parameters of a multi-storey shear building. The parameters are identified here using uncertain response of the structure subject to ambient and forced vibration with interval initial condition and horizontal displacement also in interval form.

Keywords: System Identification; Artificial neural network; Interval; Fuzzy; Interval neural network; Fuzzy neural network, Orthogonal polynomials, Chebyshev, Legendre, Hermite, Functional Link neural network.

Contents

Certificate of Examination **ii**

Supervisor's Certificate **iii**

Dedication **iv**

Declaration of Originality **v**

Acknowledgment **vi**

Abstract **viii**

List of Abbreviations **xv**

List of Symbols **xv**

Chapter 1 Introduction **1**

1.1 Literature Review 3

1.2 Gaps 16

1.3 Aims and Objectives 17

1.4 Organization of the Thesis 18

Chapter 2 Preliminaries **22**

2.1 Artificial Neural Network (ANN) 22

2.2 Basic Definitions 29

2.3 Algorithm for Interval Neural Network (INN) 32

2.4 Algorithm for Fuzzy Neural Network (FNN) 36

Chapter 3 System Identification from Frequency Data Using Interval and Fuzzy Neural Network **41**

3.1 Analysis and Modelling for Interval Case 41

3.2 Interval Neural Network Model 43

3.3 Analysis and Modelling for Fuzzy Case 45

3.4 Fuzzy Neural Network Model 47

3.5 Case Studies 49

3.5.1 Crisp Case 49

3.5.2 Interval Case 52

3.5.3 Testing for Interval Case 58

3.5.4 Fuzzy Case 59

3.5.6 Testing for Fuzzy Case 66

3.6 Conclusion 68

Chapter 4 System Identification from Response Data Using Interval and Fuzzy Neural Network 70

4.1 Analysis and Modelling for Interval Case 70

4.2 Interval Neural Network Model 72

4.3 Analysis and Modelling for Fuzzy Case 72

4.4 Fuzzy Neural Network Model 74

4.5 Results and Discussion 74

4.5.1 Interval Case 74

4.5.2 Fuzzy Case 88

4.6 Conclusion 105

Chapter 5 System Identification through Seismic Data Using Interval and Fuzzy Neural Network 107

5.1 Response Analysis for Single Degree of Freedom System Subject to Ground Motion for Interval Case 107

5.2 Interval Neural Network Model with Single Input and Output Node 109

5.3 Response Analysis for Single Degree of Freedom System Subject to Ground Motion for Fuzzy Case 110

5.4 Transformation of Data from Bipolar to Unipolar for Fuzzy Case 112

5.5 Fuzzy Neural Network Model with Single Input and Output Node 112

5.6 Response Analysis for Multi Degree of Freedom System Subject to Ground Motion for Interval Case 113

5.7 Transformation of Data from [-1, 1] to [0, 1] 116

5.8 Learning Algorithm of Interval Neural Network Using Unipolar Activation Function 116

5.9 Transformation of Data from [0, 1] to [-1, 1] by Inverse Transformation 117

5.10 Numerical Results and Discussion 118

5.10.1 Single Degree of Freedom System for Interval Case 118

5.10.2 Single Degree of Freedom System for Fuzzy Case 129

5.10.3 Multi Degree of Freedom System for Interval Case 138
5.11 Conclusion 153
Chapter 6 System Identification by Cluster of Interval and Fuzzy Neural Network 155
6.1 System Identification of Structural Parameters in Interval Form 155
6.2 Learning Algorithm for Single Layer Interval Neural Network 159
6.3 System Identification of Structural Parameters in Fuzzified Form 161
6.4 Learning Algorithm for Single Layer Fuzzy Neural Network 164
6.5 Numerical Examples 167
6.5.1 Interval Case 167
6.5.2 Fuzzy Case 172
6.6 Conclusion 178
Chapter 7 Functional Link Neural Network Based System Identification from Frequency Data 179
7.1 Modelling for System Identification of Multi-Storey Shear Buildings 179
7.2 Functional Link Neural Network 180
7.3 Learning Algorithm of Functional Link Neural Network (FLNN) 181
7.3.1 Structure of Chebyshev Neural Network 182
7.3.2 Structure of Legendre Neural Network 183
7.3.3 Structure of Hermite Neural Network 184
7.4 Results and Discussion 184
7.4.1. Chebyshev Neural Network Based Results 185
7.4.2. Legendre and Hermite Neural Network Based Results 190
7.5 Conclusion 199
Chapter 8 Functional Link Neural Network Based System Identification through Seismic Data 200
8.1 Modelling for Response Analysis for Multi-Degree of Freedom System 200
8.2 Functional Link Neural Network 202
8.2.1 Learning Algorithm of Functional Link Neural Network (FLNN) 202
8.2.2 Structure of Chebyshev Neural Network 203
8.2.3 Structure of Legendre Neural Network 204

8.3 Results and Discussions 204

8.3.1 Training Case 204

8.3.2 Testing Case 212

8.4 Conclusion 215

Chapter 9 Interval Based Functional Link Neural Network for System Identification via Response Data 217

9.1 Analysis and Modelling with Interval Case 217

9.2 Interval Orthogonal Polynomials 218

9.2.1 Interval Chebyshev Polynomial 218

9.2.2 Interval Legendre Polynomial 219

9.3 Learning Algorithm of Interval Functional Link Neural Network (IFLNN) 219

9.4 Results and Discussion 221

9.4.1 Ambient Vibration 222

9.4.2 Force Vibration 223

9.5 Conclusion 229

Chapter 10 Conclusions and Future Directions 230

10.1 Overall Conclusion 230

10.2 Scope for Further Research 234

References 236

Dissemination 255

List of Abbreviations

1	SI	System Identification
2	TFN	Triangular Fuzzy Number
3	TrFN	Trapezoidal Fuzzy Number
4	ANN	Artificial Neural Network
5	EBPTA	Error Back Propagation Training Algorithm
6	INN	Interval Neural Network
7	FNN	Fuzzy Neural Network
8	MNN	Multi Layer Neural Network
9	SDOF	Single Degree of Freedom
10	MDOF	Multi Degree of Freedom
11	FLNN	Functional Link Neural Network
12	ChNN	Chebyshev Neural Network
13	LeNN	Legendre Neural Network
14	HNN	Hermite Neural Network
15	IFLNN	Interval Functional Link Neural Network

List of Symbols

1	$\sim$	Interval form
2	$\underline{\quad}$	Lower bound
3	$\overline{\quad}$	Upper bound
4	$\wedge$	Fuzzified form
5	$[\underline{\quad}]_h$	h- level lower bound
6	$[\overline{\quad}]_h$	h-level upper bound
7	γ	Positive parameter
8	f	Activation function
9	η	Learning constant
10	Σ	Net summation
11	θ	Bias
12	λ	Frequency
13	ω	Eigen value
14	ω^2	Natural frequency parameter
15	ξ	Damping ratio
16	ψ	Influence co-efficient vector
17	Φ	Mode shape vector
18	Γ	Modal participation factor
19	τ	Small time interval

Chapter 1

Introduction

Different structures such as buildings, bridges, monuments etc get exposed to various natural phenomena like winds or earthquakes. After long period of excitation caused due to these natural distresses, the structures deteriorate losing its original designed behaviour which results change in structural as well as materialistic properties. These structural properties or parameters are natural frequencies, stiffness, mode shapes etc. Changes in structural parameters should be detected and estimated for safety of the structures. Hence, structural health monitoring like estimation of parameters is essential to know the present health/condition of the structure. Structural dynamic problems are generally of two types, direct and inverse problems. In direct problems, the equations governing the system and the parameters of the system are known. These parameters are used to find the response of the system for a specific input. In inverse problems, the output response for a given input is known, but either the governing equation or some of the parameters of the system are unknown.

Dynamic behaviour of complicated systems often needs to be investigated by System Identification (SI), since it usually has to meet certain requirements. In general, the system identification problems are the inverse (vibration) problems whose solutions are sometimes not unique and difficult to handle by direct computational and mathematical models. Rapid progress in the field of computer science and computational mathematics during recent decades has led to an increasing use of computers and efficient models to analyze, supervise and control technical processes. The use of computers and efficient mathematical tools allow identification of the process dynamics by evaluating the input and output signals of the system. Result of such process identification is usually a mathematical model by which the dynamic behaviour can be estimated or predicted. Modal-parameter SI and physical parameter SI are two major branches in SI. SI techniques are also applied to determine vibration characteristics, modal shapes, damping ratios and structural response of complex structural system so as to frame knowledge for modelling and assessing current design procedures. In System Identification, mathematical models need to be developed for a physical system from given experimental data. With the help of a model, the engineers would able to locate and detect the damage in the structures. It is becoming an interesting field of research and has attracted many researchers in recent decades.

It may be noted that direct mathematical expression for parametric study of structures is not always possible. The governing equations of inverse vibration problems are generally coupled linear or nonlinear differential equations that depend upon the physical model of the structure. Again, it is not always possible to get the solutions for above mentioned problems using classical methods. As such alternate soft computing methods like Artificial Neural Networks (ANNs) are being used in recent years to handle the above problems. It is well known that traditional neural network methods have inherent advantage because it can model the experimental data where good mathematical model is not available. Although, in order to train neural network for SI problems we require experimental data. Further, it may not be always possible to get the data in exact or crisp form due to the errors and uncertainty involved while experimenting. Hence investigations are needed to handle the SI problems where data in uncertain form may be available. In this investigation, uncertainty has been considered as interval and fuzzy. As such, the study involves to solve the related differential equations by developing interval and fuzzy neural network model which can handle uncertain data. On the other hand, traditional multilayer neural network usually takes more time for computing. Accordingly, functional link neural network has been developed to solve the above said problems in order to have less computation time. The main objective of the present investigation is to solve system identification problems of structural dynamics viz. that of multistory shear buildings using interval, fuzzy and functional link neural network models. Some studies and research have already been done in SI using various methods. Based on these methods the literature survey has been categorized as below:

- ❖ SI Based on Probabilistic, Model Updating, Eigenvalue and Other Numerical Approach;
- ❖ Damage Detection Using SI Techniques;
- ❖ Artificial Neural Network Approach to SI Problems;
- ❖ Damage Detection Based on ANN Models;
- ❖ Interval Neural Network (INN) Models;
- ❖ Fuzzy Neural Network (FNN) Models;
- ❖ Functional Link Neural Network (FLNN) Models.

1.1 Literature Review

1.1.1 SI Based on Probabilistic, Model Updating, Eigenvalue and Other Numerical Approach

Different techniques for improving structural dynamic models were surveyed in review papers like Bekey [1], Hart and Yao [2], Ibanez [3], Sinha and Kuszta [4], Datta et al. [5]. Kerschen et al. [6] discussed about the current status and future direction of nonlinear system identification problems in structural dynamics. Review paper related to vibration based damage identification methods has been written by Fan and Qiao [7]. Recently Sirca and Adeli [8] presented a very interesting and important review report on state of the art of system identification in structural engineering. Great amount of research in SI using different methods has been conducted. Some research and related works on SI may be stated as Masri et al. [9], Natke [10]. Zhao et al. [11] proposed a localized identification approach through substructuring in the frequency domain. The proposed approach can be used to identify the structural parameters in any part of interest in a structure. System identification technique for detecting changes in both linear and non-linear structural parameters has been given by Loh and Tou [12]. Yuan et al. [13] identified the structural mass and stiffness matrices of shear buildings from test data. Udwadia and Proskurowski [14] used an associative memory approach to identify the properties of structural and mechanical systems. Parameter identification of different structures and buildings using varies procedures has been studied by few authors. Sanayei et al. [15] introduced a new error function to use natural frequencies and associated mode shapes which is measured at a selected subset of degrees of freedom for stiffness and mass parameter estimation at element level. The eigenspace structural identification technique for tall buildings subjected to ambient excitations was introduced by Quek et al. [16]. Huang [17] identified structural parameters from ambient vibration using multivariate AR model. Koh et al. [18] proposed several GA based substructural identification methods, which work by solving parts of the structure at a time to improve the convergence of mass and stiffness estimates particularly for large systems. Reliable and stable method for simultaneous identification of stiffness-damping of shear type buildings have been developed by Takewaki and Nakamura [19] using stationary random records under limited observation. An innovative algorithm based on probabilistic approach is developed by Lei et al. [20] for damage identification considering measurement noise uncertainties. The probability of identified structural damage is further derived based on the reliability theory.

Dynamic behaviour of structures can be studied analytically, numerically and/or experimentally. Though different methods are used to study the dynamic behaviour, each method has its own advantages and limitations. In order to reconciliate these limitations or short comings and to determine the dynamic properties of a structure, model correlation and model updating procedure/method should be performed. Model updating refers to the methodology that determines the most plausible structural model for an instrumented structural system. In this regard, few papers may be mentioned on model updating of structural systems as Friswell and Mottershead [21], Fassois and Sakellariou [22]. State space-based structural identification theory, its implementation and applications has been presented by Alvin et al. [23]. Yu et al. [24] formulated and improved a finite element model-updating method for parameter identification of framed structures. Modal updating is studied on various structures but most widely studied structural systems are the shear buildings. Previous works state that model updating of shear buildings depend mostly on the use of modal parameter identification and physical or structural parameter identification to drive the corresponding update procedures. Since eigen mode data are obtainable from the established modal testing techniques and eigen modes contain a large amount of information about the structure in compact form, modal parameter data such as mode shapes, damping ratios and frequencies have been frequently used in modal updating of the structural systems, Bhat [25], Lu and Tu [26], Perry et al. [27]. Brownjohn [28] used modal analysis procedure, Natural Excitation Technique with Eigensystem Realization Algorithm and frequency domain decomposition (or IPP) to study ambient vibration in tall buildings. Method was developed by Mahmoudabadi et al. [29] for parametric system identification for a classically damped linear system using frequency domain and extended their work for non-classically damped linear systems subjected to six components of earthquake ground motions. Tang et al. [30] utilized a differential evolution (DE) strategy for parameter estimation of the structural systems with limited output data, noise polluted signals, and no prior knowledge of mass, damping, or stiffness matrices. Nandakumar and Shankar [31] presented a novel inverse scheme based on consistent mass transfer matrix to identify the stiffness parameters of structural members. They used a non-classical heuristic particle swarm optimization algorithm (PSO). Billmaier and Bucher [32] discussed selective sensitivity analysis and used this method to solve system identification problems.

Different numerical methodologies have been used by researchers to study system identification problems, Muthukumaran et al. [33], Lagaros et al. [34]. Identification of physical parameters such as mass, damping and stiffness matrices of linear structures has been studied by Yang et al. [35] based on Hilbert–Huang spectral analysis. Procedure which systematically modifies and identifies the structural parameters using the prior known estimates of the parameters with the corresponding vibration characteristics and the known dynamic data is given by Chakraverty [36]. Nicoud et al. [37] presented a system identification methodology that explicitly treats factors which affect the success of identification. Chakraverty [38] used Holzer criteria along with some other numerical methods to estimate the global mass and stiffness matrices of the structure from modal test data. Physical parameter system identification methods to determine the stiffness and damping matrices of shear-storey buildings have been proposed by Yoshitomi and Takewaki [39]. A two-stage Kalman estimator approach is proposed by Lie and Jiang [40] for identification of nonlinear structural parameters under limited acceleration output measurements. Structural parameter identification algorithm using additional known masses has been presented by Dinh et al. [41]. Beskhyroun et al. [42] investigated the dynamic behaviour of a full scale 13-storey-reinforced concrete building under forced vibration, ambient vibration and distal earthquake excitation. Modal parameter identification approaches and damage diagnosis methods based on Hilbert Huang transform (HHT) are proposed by Jianping et al. [43]. Wang et al. [44] used extended Kalman filter method for identification of structural stiffness parameters. Cho et al. [45] presented the decentralized system identification using stochastic subspace identification (SSI) for wireless sensor networks (WSNs).

1.1.2 Damage Detection Using SI Techniques

One of the most frightening and destructive phenomena of nature is a severe earthquake and its terrible after effects. An earthquake is the sudden, rapid shaking of the earth caused by the breaking and shifting subterranean rock as it releases stress that has accumulated over a long time. Earthquakes are one of the most costly natural hazards faced by the world posing a significant risk to the public safety. The risks that earthquakes pose to society, including death, injury and economic loss, can be greatly reduced by better planning, construction, mitigation practices before earthquakes happen, providing critical and timely information to improve response after they occur. There is no way to stop these natural phenomena, but seismologists have several methods so that they can estimate approximately or predict future earthquake events. By studying the amount of earthquakes and the time that they happen in a certain area, seismologist can then guess the probability of another earthquake occurring in the area within a given time. This will certainly give an idea to people about the time period of the occurrence of the next earthquake, so that they can prepare themselves for another possible quake. The prediction of the real earthquake ground motion at a particular building site is very complex and difficult. Earthquakes usually occur without warning. So, protection of cultural heritage against the effect of earthquake is an interdisciplinary research where the knowledge, skills and experience of earthquake along with structuralengineers assisted by architects, art historians and material scientists are required. Health monitoring, SI, theoretical and experimental assessment of structural performance, design, testing and implementation of retrofit are some of the main steps of any modern earthquake protection methodology for conservation of cultural heritage. As such after a long span of time, the historical or other structures deteriorate due to application of various man-made and natural hazards. So, it is a challenging task to know the present health of the above structures to avoid failure.

SI techniques play an important role in investigating and reducing gaps between the structural systems and their structural design models. This is also true in structural health monitoring for damage detection. SI techniques are also used in damage detection. Works related to structural damage detection using different methods were given by Angeles and Alvarez-Icaza [46], Niu [47]. Non-parametric structural damage detection methodology based on non-linear system identification approaches has been given by Masri et al. [48] for health monitoring of structure-unknown systems. Kao and Hung [49] gave two steps for structural damage detection. The first step involves system identification using neural system identification networks (NSINs) to identify the undamaged and damaged states of a structural system and the second step involves structural damage detection using the aforementioned trained NSINs to generate free vibration responses with the same initial condition or impulsive force. Several studies have already been done to identify structural parameters with the help of seismic response data. Procedure for nonlinear system identification using prediction error identification method with state-space description is presented in Furukawa et al. [50]. Modified random decrement method together with the Ibrahim time domain technique has been used by Lin et al. [51] to evaluate the modal frequencies, damping ratios and mode shapes of an asymmetric building. Pillai and

Krishnapillai [52] presented a multistage identification scheme for structural damage detection with the use of modal data using a hybrid neural network strategy. Wang and Cui [53] proposed a two-step method for simultaneous identification of structural parameters with unknown ground motion. Hegde and Sinha [54] proposed an efficient procedure to determine the natural frequencies, modal damping ratios and mode shapes for torsionally coupled shear buildings using earthquake response records The procedure applies eigenrealization algorithm to generate the state-space model of the structure using the cross-correlations among the measured responses. Methodology for identification of state-space models of a building structure using time histories of the earthquake-induced ground motion and of the corresponding structural responses is presented by Hong et al. [55]. Zhang et al. [56] proposed a probabilistic method to identify damages of the structures with uncertainties under unknown input. The proposed probabilistic method is developed from a deterministic simultaneous identification method of structural physical parameters and input based on dynamic response sensitivity.

Structural parameter identification and damage detection approach using displacement measurement time series has been given by Xu et al. [57]. Xu et al. [58] proposed a new computational method based on linear and nonlinear regression analysis technique, for identification of the linear and nonlinear physical parameters of base-isolated multi-storey buildings using earthquake records. Ebrahimian and Todorovska [59] introduced a non uniform Timoshenko beam model of a high rise building with piecewise constant properties along with an algorithm for system identification from earthquake records. Zhou et al. [60] investigated a simple method for physical parameters identification of a nonlinear hysteretic structure with pinching behaviour from seismic response data. New near real time hybrid frame work for system identification of structures dealing with data streaming from a structural health monitoring (SHM) system is proposed by Guo et al. [61]. Derras and Bekkouche [62] used the feed-forward artificial neural network method (ANN) with a conjugate gradient back-propagation rule to estimate the maximum Peak Ground Acceleration (PGA) of the three components (vertical, east-west and north-south). A novel adaptive scheme is presented by Lagaros and Papadrakakis [63] in order to predict the dynamic behaviour of structural systems under earthquake loading condition. Zamani et al. [64] used artificial neural network to train the responses of structural systems for a particular earthquake. Robles and Hernandez-Becerril [65] created a seismic alert system which is based on artificial neural networks and genetic algorithms. Application of the artificial neural network (ANN) to predict pseudo spectral acceleration or peak ground acceleration is explored in the studies of Hong et al. [66]. Ramhormozian et al. [67] used artificial neural network method to predict principal ground motion parameters for quick post-earthquake damage assessment of bridges.

1.1.3 Artificial Neural Network (ANN) Approach to SI Problems

Generally when the systems are modeled as linear identification problem, it often turns out to be a non-linear optimization problem. This requires an intelligent iterative scheme to have the required solution. There exist various online and offline methods, namely the Gauss-Newton, Kalman filtering and probabilistic methods such as maximum likelihood estimation etc. However, the identification problems for large number of parameters, following two basic difficulties are faced often:

a) Objective function surface may have multiple maxima and minima and the convergence to the correct parameters is possible only if the initial guess is considered as close to the parameters to be identified.
b) Inverse problem, in general gives non-unique parameter estimates.

To overcome these difficulties, various researchers have proposed identification methodologies for the said problems using powerful technique of Artificial Neural Network (ANN) models. Artificial Neural Network (ANN) is a class of mathematical algorithm inspired by biological nervous system. This is one of the popular areas in the mathematics of artificial intelligence research and based on organizational structure of human brain. ANN is a powerful computational approach which depends upon various parameters and learning methods. In recent years, artificial neural networks have been used widely in various fields of engineering and science. ANN is mathematical processes by which one may study pattern learn tasks and solve complex problems like identification, function approximation, clustering and predication etc.ANN is a field which is growing from the last few decades, so an enormous amount of literature has been written on the topic of Artificial Neural Networks which helps to solve system identification problems. As regards few research works are reviewed and cited here for better understanding of the problems.

ANNs provide a fundamentally different approach to SI problems. They have been successfully applied for identification and control of dynamic systems in various fields of engineering because of its excellent learning capacity and high tolerance to partially inaccurate data. A number of studies namely Kosmatopoulos et al. [68], Lagaros and Papadrakakis [69] and the references mentioned below have used ANN for solving structural identification problems. Yun and Bahng [70] proposed a neural network-based substructural identification for the estimation of the stiffness parameters of a complex structural system, particularly for the case with noisy and incomplete measurement of the modal data. Decentralized stiffness identification method with neural networks for a multi-degree of freedom structure has been developed by Wu et al. [71]. Neural network-based strategy was developed for direct identification of structural parameters from the time-domain dynamic responses of a structure without any eigenvalue analysis by Xu et al. [72]. Neural network-based method to determine the modal parameters of structures from field measurement data was given by Chen [73]. Procedure for identification of structural parameters of two-storey shear buildings by an iterative training of neural networks was proposed by Chakraverty [74]. System identification of an actively controlled structure using frequency response functions with the help of ANNs has also been studied by Pizano [75] for single-input, single-output and multiple-input single-output system. Soft computing methods for model updating of multi storey shear buildings for simultaneous identification of mass, stiffness and damping matrices have been investigated by Khanmirza et al. [76]. Facchini et al. [77] presented an artificial neural network based technique for the output-only modal identification of structural systems. Khanmirza et al. [78] gave a novel method based on ANN for simultaneous identification of physical parameters as well as separation of linear physical parameters from the nonlinear ones, for nonlinear multi-DOF systems.

1.1.4 Damage Detection Based on ANN Models

Artificial Neural Network (ANN) has gradually been established as a powerful tool in various fields. ANN has recently been applied to assess damage in structures. In this regards lots of works in structural health monitoring and damage detection using ANN have been done by various researchers. Back-propagation neural network (BPN) to elucidate damage states in a three-storey frame by numerical simulation has been studied by Wu et al. [79]. Conte et al. [80] gave a neural network based approach to model the seismic response of multi-storey frame buildings. Pandey and Barai [81] detected damage in a bridge truss by applying ANN to numerically simulated data. Counter-propagation neural network (NN) to locate damage in beams and frames has been studied by Zhao et al. [82]. Localized damage detection and parametric identification method with direct use of earthquake responses for large scale infrastructures have also been proposed by Xu et al. [83]. Other advanced studies which include application of neural network techniques for damage detection have been cited here. Novel procedure for identifying dynamic characteristics of a building from seismic response data using NN model has been given by Huang et al. [84]. Mathur et al. [85] used feed forward, multilayer, supervised neural network with error back propagation algorithm to predict responses of typical rural house subject to earthquake motions. Chakraverty et al. [86] used artificial neural network model to compute response of structural system by training the model for a particular earthquake. An approach to detect structural damage using ANN method with progressive substructure zooming has been presented by Bakhary et al. [87]. This method also uses the substructure technique together with a multi-stage ANN models to detect the location and extent of the damage. Zhang et al. [88] studied the application of neural networks to damage detection in structures. In order to simulate and estimate structural response of two-storey shear building by training the model for a particular earthquake using the powerful technique of artificial neural network models has been presented by Chakraverty et al. [89]. Oliva and Pichardo [90] introduced new methods for seismic hazard evaluation in a geographic area. Kerh et al. [91] gave neural network approach for analyzing seismic data to identify potentially hazardous bridges. Application of neural network model for earthquake prediction in East China has been presented by Xie et al. [92].

The application of ANNs and wavelet analysis to develop an intelligent and adaptive structural damage detection system has been investigated by Shi and Yu [93]. Aghamohammadi et al. [94] used neural network to model and to estimate the severity and distribution of human loss as a function of building damage in the earthquake disaster in Iran. Application of neural networks in bridge health prediction based on acceleration and displacement data domainhave been given by Suryanita and Adnan [95]. Reyes et al. [96] applied artificial neural networks, to predict earthquakes in Chile. Niksarlioglu and Kulahci [97] determined the relationships between radon emissions based on the environmental parameters and earthquakes occurring along the East Anatolian Fault Zone (EAFZ), Turkiye and predicted magnitudes of some earthquakes with the artificial neural network (ANN) model. An optimized set of seismicity parameters for earthquake prediction has been obtained by Alvarez et al. [98]. Hakim et al. [99] developed an Adaptive Neuro-Fuzzy Inference System (ANFIS) and ANNs technique to identify damage in a model steel girder bridge using dynamic parameters. To highlight the recent trends in earthquake abnormality detection, including various ideas and applications, in the field of Neural Networks, valid papers related to ANNs are reviewed and presented by Sriram et al. [100]. It may be seen from the above that artificial neural networks (ANNs) provide a fundamentally different approach to system identification and dynamic problems.

1.1.5 Interval Neural Network (INN) Models

It is revealed from the above literature review that various authors developed different identification methodologies using ANN. They supposed that the data obtained are in exact or crisp form. But in actual practice the experimental data obtained from equipments are with errors that may be due to human or equipment error thereby giving uncertain form of the data along with uncertain structural parameters. In view of the above, some studies have been done by using Interval Neural Networks (INNs) and Fuzzy Neural Networks (FNNs) in different fields. Ishibuchi and Tanaka [101] extended the back-propagation algorithm to the case of interval input vectors. A new architecture of neural networks with interval weights and interval bias and its learning algorithm has been discussed by Ishibuchi et al. [102]. Kwon et al. [103] gave three approaches to the learning of neural networks that realize nonlinear mappings of interval vectors. Beheshti et al. [104] defined interval neural network and categorized general three-layer neural network training problems into two types that is type1 and type2 according to their mathematical model. Using these general algorithms one can develop specific software which can efficiently solve interval weighted neural network problems. Garczarczyk [105] studied a four layer feed forward network considering interval weights and interval biases. Escarcina et al. [106] developed interval computing in neural network which is based on one layer interval neural network. Application of interval valued neural networks to a regression problem has been presented by Chetwynd et al. [107]. Their work was concerned with exploiting uncertainty in order to develop a robust regression algorithm for a pre-sliding friction process based on a nonlinear Auto-Regressive with eXogeneous inputs neural network. Wang et al. [108] used interval analysis technique for structural damage identification. Influences of uncertainties in the measurements and modeling errors on the identification were also investigated in this paper.

Zhang et al. [109] gave a numerically efficient approach to treat modeling errors with the help of intervals which results in bounding of the identified parameters. Interval GA (Genetic Algorithm) for evolving neural networks with interval weights and biases was developed by Okada et al. [110] where they have proposed an extension of genetic algorithm for neuro evolution of interval-valued neural networks. Interval based weight initialization method for a sigmoidal feedforward artificial neural network has been given by Sodhi and Chandra [111]. Lu et al. [112] presented an interval pattern matcher that can identify patterns with interval elements using neural networks. Interval neural network technique based on response data for system identification of multi storey shear building has been done by Chakraverty and Sahoo [113]. Chakraverty and Behera [114] investigated parameter identification of multistorey frame structure using uncertain dynamic data. Sahoo et al. [115] proposed identification methodologies for multi-storey shear buildings using Interval Artificial Neural Network (IANN) which can estimate the structural parameters. Very recently system identification problems using INN have been studied by Chakraverty and Sahoo [116].

1.1.6 Fuzzy Neural Network (FNN) Models

Various research works are being done by using FNN in different application problem. Algorithms on fuzzy neural network were given by Buckley and Hayashi [117], Hayashi et al. [118]. Survey paper on fuzzy neural network has been written by Buckley and Hayashi [119]. Buckley and Hayashi [120] showed how to represent fuzzy expert systems and fuzzy controllers as neural nets and as fuzzy neural nets. Ishibuchi et al. [121] developed architecture for neural networks where the input vectors are in terms of fuzzy numbers. Methodology for FNNs where the weights and biases are taken as fuzzy numbers and the input vectors as real numbers has been proposed Ishibuchi et al. [122]. FNN with trapezoidal fuzzy weights was also presented by Ishibuchi et al. [123]. They have developed the methodology in such a way that it can handle fuzzy inputs as well as real inputs. In this respect, Ishibuchi et al. [124] derived a general algorithm for training a fuzzified feed-forward neural network that has fuzzy inputs, fuzzy targets and fuzzy connection weights. The derived algorithms are also applicable to the learning of fuzzy connection weights with various shapes such astriangular and trapezoidal. Another new algorithm for learning fuzzified neural networks has also been developed by Ishibuchi et al. [125]. Different applications problems in fuzzy logic and fuzzy neural network in the field of science and engineering may be reffered as Som and Mukherjee [126], Pal and Mitra [127], Packirisamy et al. [128], Mitra and Hayashi [129], Mitra and Pal [130], Chakraborty et al. [131], Qiu et al. [132], Rani and Gulati [133-134]. Lu [135] proposed a FNN-based technique to construct an adaptive car-following indicator. The fuzzified neural network based on fuzzy number operations has been presented by Li et al. [136] as a powerful modelling tool. New learning law for Mamdani and Takagi-Sugeno-Kang type FNNs based on input-to-state stability approach was suggested by Yu and Li [137]. The new learning schemes employ a time-varying learning rate that is determined from input–output data and model structure. Stable learning algorithms for the premise and the consequence parts of fuzzy rules are also proposed. Self-constructing FNN employing extended Kalman filter was designed and developed by Er et al. [138].

Pankaj and Wilscy [139] proposed a method for face recognition using a fuzzy neural network classifier based on the Integrated Adaptive Fuzzy Clustering (IAFC) method. Wang [140] presented a generalized ellipsoidal basis function-based online self-constructing FNN which implements a Takagi-Sugeno-Kang (TSK) fuzzy inference system. Umoh et al. [141] developed a fuzzy-neural network model and applied the model for effective control of profitability in paper recycling to improve production accuracy, reliability, robustness and to maximize profit generated by an industry. Vijaykumar et al. [142] used T-S fuzzy neural network in speech recognition systems. Various deterrent factors influencing the supply chain to forecast the production plan have been presented by Sharma and Sinha [143]. Use of neural network combined with fuzzy logic for long term load forecasting has been given by Swaroop [144]. The states of the art for the application of FNN in diagnosis, recognition, image processing and intelligence robot control of medicine are reviewed by Zhang and Dai [145]. They also proposed the application of FNN in medicine. Three new algorithms for Takagi-Sugeno-Kang fuzzy system based on training error and genetic algorithm are proposed by Malek et al. [146]. Recent work on robust FNN sliding mode control scheme for IPMSM drives were also developed by Leu et al. [147]. Zahedi et al. [148] presented the prediction of ozone pollution as a function of meteorological parameters around the Shuaiba industrial area in Kuwait by a FNN modelling approach. An adaptive FNN controller for missile guidance has been given by Wang and Hung [149]. System identification problems based on FNN modelling for identification of structural parameters of multi-storey shear buildings have been done by Sahoo and Chakraverty [150], Chakraverty and Sahoo [151].

1.1.7 Functional Link Neural Network (FLNN) Models

Computation in Multilayer neural network is sometimes time consuming due to presence of hidden layers. Hence an efficient learning method is required which will have fast computation. Therefore FLNN are developed because FLNN are highly effective and computationally more efficient than multilayer neural network. In FLNN the hidden layer is excluded by enlarging the input patterns with the help of orthogonal polynomials like Chebyshev, Legendre and Hermite polynomials. FLNN has been used in few problems which can be stated as Pao [152], Patra et al. [153]. Patra and Kot [154] solved dynamic nonlinear system identification problem using Chebyshev functional neural network. Functional link artificial neural network based active noise control structure is developed by Panda and Das [155] for active mitigation of nonlinear noise processes. Mackenzie and Tieu [156] developed a method for obtaining the correlation of a two Hermite neural network. Neural network for calculating the correlation of a signal with a Gaussian function is described in Mackenzie and Tieu [157]. Purwar et al. [158] used ChNN model for system identification of unknown dynamic nonlinear discrete time systems. Ma and Khorasani [159] proposed a new type of a constructive one-hidden-layer feedforward neural network (OHL-FNN) that adaptively assigns appropriate Orthonormal Hermite polynomials to its generated neurons. Misra and Dehuri [160] studied Functional Link Artificial Neural Networks (FLANN) for the task of classification. Patra et al. [161] studied the application of artificial neural networks (ANNs) for adaptive channel equalization in a digital communication system using 4-quadrature amplitude modulation (QAM) signal constellation. Patra et al. [162] presented a computationally efficient Legendre neural network (LeNN) for equalization of nonlinear communication channels for wireless communication systems. Dehuri and Cho [163] gave a survey report on FLNN models and also developed a new learning scheme for Chebyshev functional link neural network. Xiuchun et al. [164] constructed Chebyshev neural network to obtain the weight-direct-determination method. Mishra et al. [165] proposed a single layer FLANN structure for denoising of image corrupted with Gaussian noise. Mishra et al. [166] used a Chebyshev functional link artificial neural network for image denoising which is corrupted by Salt and Pepper noise. Patra and Bornand [167] used Legendre neural network for identification of nonlinear dynamic systems. Shaik et al. [168] used Chebyshev neural network to solve the problem of observer design for the twin rotor multi-input-multi-output (MIMO) system.

Dehuri [169] presented hybrid learning scheme to train ChNN in classification problems. Using Chebyshev neural network model, Patra [170] modeled the tunnel junction characteristics and developed models to predict the external quantum efficiency. Nanda and Tripathy [171] used Legendre neural network based air quality parameter prediction for environmental engineering application. Nanda and Tripathy [172] introduced the idea of designing noise prediction model for opencast mining machineries using functional link artificial neural network systems. Jiang et al. [173] proposed a Chebyshev functional link neural network based model for photovoltaic modules. Hassim and Ghazali [174] evaluated the functional link neural network using an artificial bee colony model for the task of pattern classification of 2-class classification problems. New methodology using LeNN has been investigated by Ali and Haweel [175] to enhance nonlinear multi-input multi-output signal processing. Li and Deng [176] constructed a MIMO Hermite neural network for dynamic gesture recognition. An enhanced Orthonormal Hermite polynomial basis neural network (EOHPBNN) predistorter is proposed by Yuan et al. [177] and it is also experimentally validated. Parija et al. [178] investigated and compared the future location prediction between Multilayer perceptron back Propagation (MLP-BP) and FLANN. Mishra and Dash [179] used Chebyshev functional link artificial neural network for credit card fraud detection problem. Chebyshev neural network based backstepping controller has been used by Sharma and Purwar [180] for light-weighted all-electric vehicle. Mall and Chakraverty [181] solved second order non-linear ordinary differential equations of Lane-Emden type using Chebyshev neural network. New algorithm has been proposed by Mall and Chakraverty [182] to solve singular initial value problems of Emden–Fowler type equations. Manu et al. [183] investigated the problem of designing a neural network observer based on Chebyshev neural network. Goyal et al. [184] presented a robust sliding mode controller for a class of unknown nonlinear discrete-time systems in the presence of fixed time delay. Chiang and Chu [185] gave a reference adaptive Hermite fuzzy neural network controller for a synchronous reluctance motor.

1.2 Gaps

It is already mentioned that system identification problems are inverse (vibration) problems consisting of coupled linear or non-linear differential equations that depend upon the physics of the system. Above literature review reveals that the inverse problems have been handled for deterministic cases in general. While experimenting, data in crisp or exact form (due to human or experimental errors) may not be possible to obtain. As such, those data may actually be in uncertain form and corresponding methodologies need to be developed.

It is an important issue about dealing with variables, parameters or data with uncertain value. There are three classes of uncertain models, which are probabilistic, fuzzy and interval. Recently, fuzzy theory and interval analysis are becoming powerful tools for various science and engineering problems. Although it is known that interval and fuzzy computations are themselves very complex to handle but these are excellent theories to mimic the uncertainty of the problems. Having these in mind, one has to develop efficient computational models and algorithms very carefully to handle these problems.

Hence, investigations are needed to solve the SI problems where data in uncertain form is available. Identification methods with crisp (exact) data are known and traditional neural network methods have already been used by various researchers. It is worth mentioning that neural network methods have inherent advantage because it can model the experimental data and may find the functional relation between input and output where good mathematical model is not available/possible. But when the data are in uncertain form then traditional ANN may not be applied. Accordingly, new ANN models need to be developed which may solve the targeted uncertain SI problems. In this respect, one may note from the above literature survey that Interval Neural Network (INN) and Fuzzy Neural Network (FNN) models have been developed for few other problems but none for SI problems. So, INN and FNN models should be developed which may handle the uncertain data (in term of interval and fuzzy) in SI problems.

Moreover this study also involves SI problems of structures (viz. shear buildings) with respect to earthquake data (that is due to the natural calamities) in order to know the health of a structure. It is well known that earthquake data are both positive and negative. The Interval Neural Network and Fuzzy Neural Network model may not handle the data with negative sign due to the complexity in interval and fuzzy computation. As such, methods should also be developed to handle this type of problem. Finally, it is also seen that training in Multilayer Neural Network (MNN), Interval Neural Network (INN) and Fuzzy Neural Network (FNN) are time consuming because it depends upon the number of nodes in the hidden layer and convergence of the weights during training. Keeping this in mind we may also target to develop efficient learning method which can handle such problem with less computation. It may however be noted that the developed methodology not only be important for the mentioned problems but those may very well be used in other application problems too.

1.3 Aims and Objectives

In reference to the above gaps, the main objective of this thesis is to solve uncertain coupled ordinary differential equations with respect to SI problems. As said above, in general we may not obtain the corresponding input and output values (experimental) exactly (or in crisp form) but we may have only uncertain information of the data. For our present work, these uncertain data have been considered in terms of interval and fuzzy numbers. Accordingly, identification methodologies should be developed for SI problems of multistory shear buildings by proposing new powerful technique of Interval Neural Network (INN) and Fuzzy Neural Network (FNN) models. Powerful and efficient model(s) (with respect to computation time) of neural network for system identification should also be investigated. In view of the above gaps and motivation, the main objectives for the present thesis are as follows:

- Investigation of existing ANN computations, their training methods and architecture with respect to SI problems;
- Development of new INN algorithms, its learning method and training methodology to handle the SI problems;
- Validation of above INN algorithms for identification of the physical parameters (stiffness etc.) of multistory shear buildings;
- Development of new FNN algorithms, its learning method and training methodology;
- Validation of above FNN algorithms for identification of the physical parameters (stiffness etc.) of multistory shear buildings;
- Solving SI problems through cluster of ANNs when data are uncertain;

- ❖ Development of INN and FNN when data are combination of positive and negative such as seismic data with respect to the structural problems;
- ❖ Development of efficient FLNN techniques that can handle system identification problems with less computation.
- ❖ Development of new interval FLNN techniques for uncertain parameter estimation (stiffness etc.) of multistory shear buildings.

1.4 Organization of the Thesis

Present work is based on solution of governing uncertain differential equation of system identification problems by proposing new ANN models which can handle uncertain data. In this investigation, these uncertain data have been considered in interval or fuzzy form. For a given input to the system, rather than solving the inverse vibration problem, the forward problem is solved to generate the solution vector. First the initial (prior) values of the physical parameters (stiffness etc.) of the system are randomized for the numerical experiment and then using these set of physical parameters the responses may be obtained. The responses and the corresponding parameters are used as the input/output in the neural net. Then the physical parameters may be identified, if new response data is supplied as input to the net. Although this is easy to handle if the data/parameter(s) are in crisp form, which is not in the actual case. All the units considered here are consistent. In all test examples the masses are taken in Kgs, stiffnesses in N/m and displacement in cm and m. As such the thesis addresses the above titled problems in a systematic manner in term of ten chapters which are briefly described below:

Overview of this thesis has been presented in Chapter 1. Related literatures on system identification methods along with ANN, INN, FNN and FLNN models are reviewed here. This chapter also contains gaps as well as aims and objectives of the present study.

Chapter 2 begins with the basic concepts of Artificial Neural Network (ANN), network architecture, types of neural network, different training process, activation functions, learning rules etc. Further, this chapter includes general notations and definitions of interval and fuzzy numbers (viz. triangular and trapezoidal fuzzy numbers). Algorithms of Interval and Fuzzy neural network have also been discussed here.

In Chapter 3, SI problems of multistorey shear structure have been solved in terms of interval and fuzzy neural network model and the uncertain structural parameters (viz. stiffness) are estimated from frequency data. The inputs in INN and FNN model are taken as the frequency parameters and outputs as the stiffness parameters. The initial design parameters viz. stiffness and mass and so the frequencies of the said problem are known. It is assumed that only the stiffness is changed and the mass remains the same after span of time. As such, equipments are available to get the present values of the frequencies and using these one may get the present parameter values by the developed mathematical model such as ANN. If sensors are placed to capture the frequency of the floors in interval and fuzzified (uncertain) form then those may be fed into the proposed new ANN model to get the present stiffness parameters. Although to train the new ANN model, set of data are generated here by solving the governing coupled (uncertain) ordinary differential equations numerically beforehand. As such converged ANN model gives the present stiffness parameter values in interval/fuzzified form for each floor. Thus one may predict the health of the structure.

In Chapter 4, forward problem for each time step is solved for a given input to the system, rather than solving the inverse vibration problem. Thus, the solution vector is generated. Again, the initial design parameters viz. stiffness, mass and so the responses of the said problem are known in uncertain form. The initial values of the uncertain physical parameters of the system are used to obtain the interval and fuzzy responses. Responses and the corresponding parameters are used as the input/output in the neural net. Next, the interval neural network and fuzzy neural network model is trained by the proposed interval and fuzzy error back propagation training algorithm. After training of the model, physical parameters may be identified if new maximum response data is supplied as input to the net which are also in interval and fuzzified form. The procedure has been demonstrated for multi-storey shear structures and the structural parameters are identified using the response of the structure subject to initial condition and horizontal displacement which are also in interval and fuzzified form.

The primary aim of Chapter 5 has been to compute response of multi-storey shear structures subject to ground motion data (interval and fuzzy) of Indian earthquakes occurred at Chamoli and Uttarkashi using INN and FNN models. The interval and fuzzy neural network are first trained for a real earthquake data viz. the ground acceleration as input and the numerically generated responses of different floors of multi-storey buildings as output. It may be noted that till date no model exists to handle positive and negative data in the INN and FNN. As such, here the bipolar data in [-1, 1] are converted first to unipolar form that is to [0, 1] by means of a novel transformation for the first time to handle the above training patterns in normalized form. Once the training is done, again the unipolar data are converted back to its bipolar form by using the inverse transformation. The trained INN and FNN architecture are then used to simulate and test the structural responses of different floors for various intensity earthquake data. It is found that the predicted responses given by INN and FNN models are good for practical purposes.

In Chapter 6, identification of structural parameters such as mass, stiffness and damping matrices of shear buildings using single layer neural network has been presented in interval and fuzzified form. To identify the physical parameters in interval and fuzzified form, the governing equations of motion are used systematically in a series (cluster) of INNs and FNNs. The equations of motion is first solved by the classical method to get responses of the consecutive stories and then the equations of motion are modified based on relative responses of consecutive stories in such a way that the new set of equations can be implemented in a cluster of INNs and FNNs. Here, single-layer INNs and FNNs have been used for training each cluster of the INN and FNN such that the converged weights give the uncertain structural parameters.

A single layer functional link neural network with multi-input and multi-output with feed forward neural network model and principle of error back propagation has been used to identify structural parameters in Chapter 7. In the Functional Link neural network model, the hidden layer is excluded by enhancing the input patterns with the help of orthogonal polynomials such as Chebyshev, Legendre and Hermite. In this case, the input is considered as the frequency parameters and the output as the structural parameters. Comparison of results among Multi-layer Neural Network (MNN), Chebyshev Neural Network (ChNN), Legendre Neural Network (LeNN), Hermite Neural Network (HNN) and desired are considered and it is found that Functional Link Neural Network models are more effective than MNN.

Chapter 8 proposes Functional Link Neural Network (FLNN) for structural response prediction of tall buildings due to seismic loads. The ground acceleration data has been taken as input and structural responses of different floors of multi-storey shear buildings have been considered as output. Here, functional expansion block in FLNN has been used along with efficient Chebyshev and Legendre polynomials. Training is done with one earthquake data and testing is done with different intensities of other earthquake data. It is seen that FLNN can very well predict the structural response of different floors of multi-storey shear building subject to earthquake data. Results of FLNN are compared with Multilayer Neural Network (MNN) and it is worth mentioning that the FLNN gives better accuracy and takes less computation time compared to MNN which shows the computational efficiency of FLNN over MNN.

In Chapter 9, a new FLNN viz. interval functional link neural network (IFLNN) has been developed and then uncertain structural parameters of multistorey shear buildings have been identified. The parameters are identified here using response of the structure for both ambient and forced vibration. In the functional link, the Chebyshev and Legendre orthogonal polynomials are taken as intervals.

Based on the present work, Chapter 10 summarizes the main findings and conclusions of the study. Finally, suggestions for future work are also outlined here.

Chapter 2

Preliminaries

This chapter presents basic concepts of Artificial Neural Network (ANN), network architecture, types of neural network, different training methods, activation functions, learning rules etc. Related notations and definitions of interval as well as fuzzy numbers (viz. triangular, trapezoidal fuzzy numbers) are included here. Further, algorithms of Interval and Fuzzy neural network have also been discussed in this chapter. As regards, basics and fundamental concepts of ANN with engineering applications can be found in books like Zurada [186], Fausett [187] and Sivanandam et al. [188] etc. Several excellent books have also been written by other authors such as Jaulin et al. [189], Zimmerman [190], Ross [191], Lee [192], Moore [193], Haghighi et al. [194], Chakraverty [195] and Chakraverty et al. [196-197] which give an extensive review on various aspects of interval, fuzzy set theory and their applications.

2.1 Artificial Neural Network (ANN)

In this section, some important definitions Zurada [186], Fausett [187] and Sivanandam et al. [188] related to ANN are included.

Neural network (NN) is an information processing paradigm which is inspired by the human nervous systems. It is known as artificial neural network (ANN) because the processing is similar to the human brain. An ANN is composed of large number of highly interconnected processing elements called the neurons which works in union to solve different problems. ANN is used in various application problems, such as pattern recognition, data classification, speech recognition, image processing and system identification through different learning procedures. ANN's are a type of artificial intelligence that attempts to imitate the way a human brain works. Artificial neural network is an information-processing system in which the neurons transfer the information. The neurons are connected with each other by connection links known as weights. These weights when multiplied with the input give the net input for any typical neural network. The output of the net is obtained by applying activation functions to the net input.

An artificial neural network is characterized by:

1. Architecture (connection between neurons)
2. Training or learning (determining weights on the connections)
3. Activation functions

2.1.2 Network Architecture

The arrangement of neurons into layers and the connection patterns within and between layers are generally called as the Network architecture. Based on the connections the structures can be classified as below

- **Feed Forward Neural Network**

  In feed forward neural network, the input signal passes into the input layer propagates through each of the hidden layers and finally emerge into the output layer. There is no feed back or back loops i.e. the signals travels in only one direction. In feed forward networks, inputs get associated with the outputs in a straight forward way.

- **Feedback Neural Network**

  In feedback network signals travels in both directions by creating loops in the network. These networks are very powerful and sometimes extremely complicated because they have closed loop connection from neuron back to itself. These networks are dynamic in nature as their state changes continuously until they reach an equilibrium point.

2.1.3 Types of Neural Network

Neural network can be classified in general as single layer and multi-layer network. The single layer and multilayer networks are examples of feedforward network, in which the signal flow from the input units to the output units.

- **Single Layer Neural Network**

  A single layer neural network has one layer of connection weights. The input layer are directly connected to output layer i.e. there is no hidden layer in between the input and output layer. The typical single layer neural network architecture is shown in Figure 2.1.

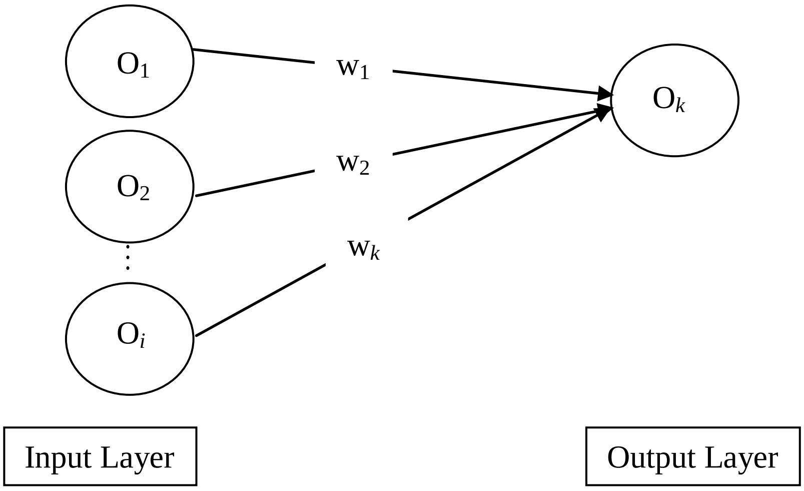

Figure 2.1: A Single Layer Neural Network

- **Multi-Layer Neural Network**

A multilayer neural network is a network with one or more layers of nodes between input and output units. The layer between the input and output units is called the hidden units. Multi-layer neural networks are used to solve more complicated problems when single layer neural network cannot be trained to perform correctly. The multilayer neural network is shown in Figure 2.2.

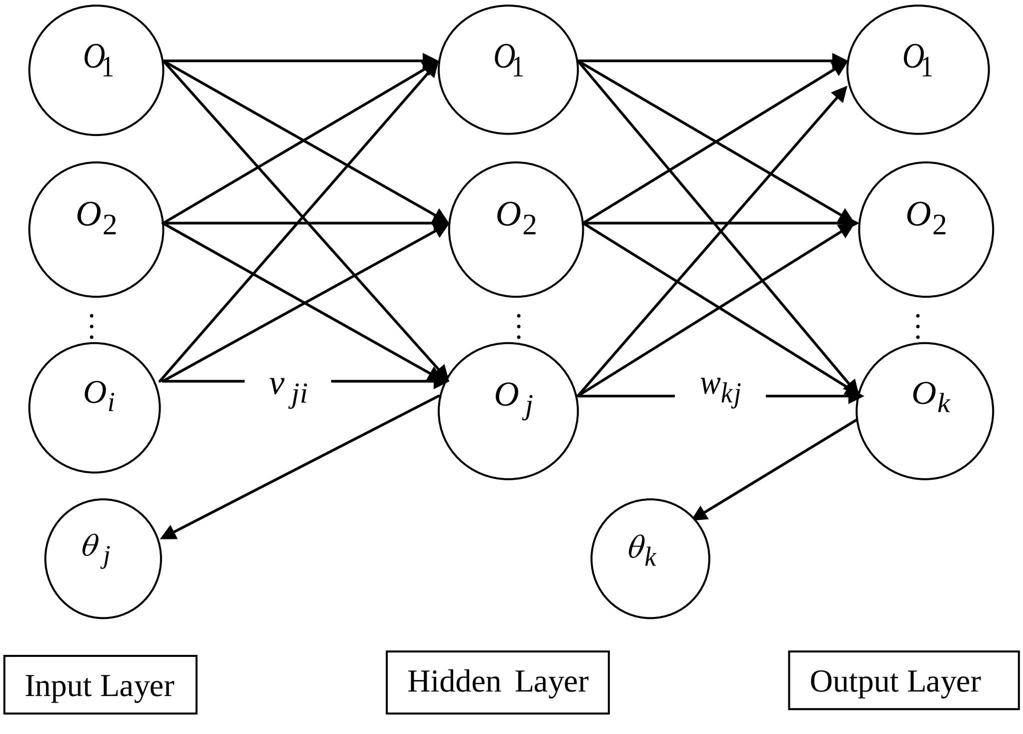

Figure 2.2: A Multi-Layer Neural Network

2.1.4 Different Training Process

The method of setting up of the weights value is known as learning. The process of achieving the expected output by modifying the weights in the connections between network layers is called training. In general there are two types of training, supervised and unsupervised training.

- **Supervised Training**

  Supervised training means training with the help of a teacher. The process of providing the network with a series of sample inputs and then comparing the output with the expected responses is supervised training. The training continues until the network is able to provide the expected response.

- **Unsupervised Training**

  Unsupervised training means training without the help of a teacher. In a neural net if the target output is not known for training the input vectors, then the training method adopted is called as unsupervised training. The net may modify the weight so that the most similar input vector is assigned to the same output unit. Unsupervised training is more complex and difficult to implement.

2.1.5 Activation Functions

An activation function is a function over the weighted sum of input (net) into a neuron. This function over the weighted sum of input (net) gives the output of the neuron. The activation function acts as a squashing function, such that the output of the neural network lies between certain values (usually within [0 , 1], or [-1 , 1]).

For present investigation, we have used unipolar and bipolar sigmoid activation functions which are monotonic and continuously differentiable functions.

- **Unipolar Sigmoid Function**

  Sigmoid functions are usually s-shaped curves. These functions are used in Multilayer neural network. The unipolar sigmoid function is also called logistic function. The unipolar sigmoid function is given as

$$O = f(Net) = \frac{1}{1+e^{-\gamma(Net)}} \tag{2.1}$$

where γ is a positive parameter. The binary sigmoid function is smooth. It has a derivative which is smooth. Its derivative is given by

$$f^{'}(Net) = \gamma \; f(Net)\,[1 - f(Net)] \tag{2.2}$$

- **Bipolar Sigmoid Function**

This function is related to the hyperbolic tangent function. The bipolar sigmoid function is given as

$$O = f(Net) = \frac{1 - e^{-\gamma(Net)}}{1 + e^{-\gamma(Net)}} \tag{2.3}$$

where γ is a positive parameter. The bipolar sigmoid function is also smooth with a smooth derivative given by

$$f^{'}(Net) = \frac{\gamma}{2}\,[1 + f(Net)]\,[1 - f(Net)]. \tag{2.4}$$

2.1.6 Learning Rules of ANN

Learning is the most impressive features of artificial neural network. Learning of artificial neural network is updation of weights and bias with some kind of learning algorithm. It is done to improve the neural network’s performance. There are many different algorithms that can be used when training artificial neural networks, each with their own separate advantages and disadvantages. Various types of learning rules used in ANN are Zurada [186]

- Hebbian Learning Rule
- Perceptron Learning Rule
- Error Back Propagation or Delta Learning Rule
- Widrow-Hoff Learning Rule
- Winner-Take Learning Rule etc.

We have used error back propagation learning algorithm to train the neural network in this thesis and so only this rule has been discussed in a bit detail below.

- **Error Back Propagation or Delta Learning Rule**

Error Back propagation learning algorithm has been introduced by Rumelhart et al. [198]. It is also known as Delta learning rule, Zurada [186] and is one of the most commonly used learning rule. It is valid for continuous activation function and is used in supervised and unsupervised training method.

The simple perceptron can handle linearly separable or linearly independent problems. Taking the partial derivative of error of the network with respect to each of its weights, we can know the flow of error direction in the network. If we take the negative derivative and then proceed to add it to the weights, the error will decrease until it approaches local minima. Then we have to add a negative value to the weight or the reverse if the derivative is negative. Because of these partial derivatives and then applying them to each of the weights, starting from the output layer to hidden layer weights, then the hidden layer to input layer weights, this algorithm is called the back propagation algorithm.

The training of the network involves feeding samples as input vectors, calculation of the error of the output layer, and then adjusting the weights of the network to minimize the error. The average of all the squared errors E for the outputs is computed to make the derivative simpler. After the error is computed, the weights can be updated one by one.

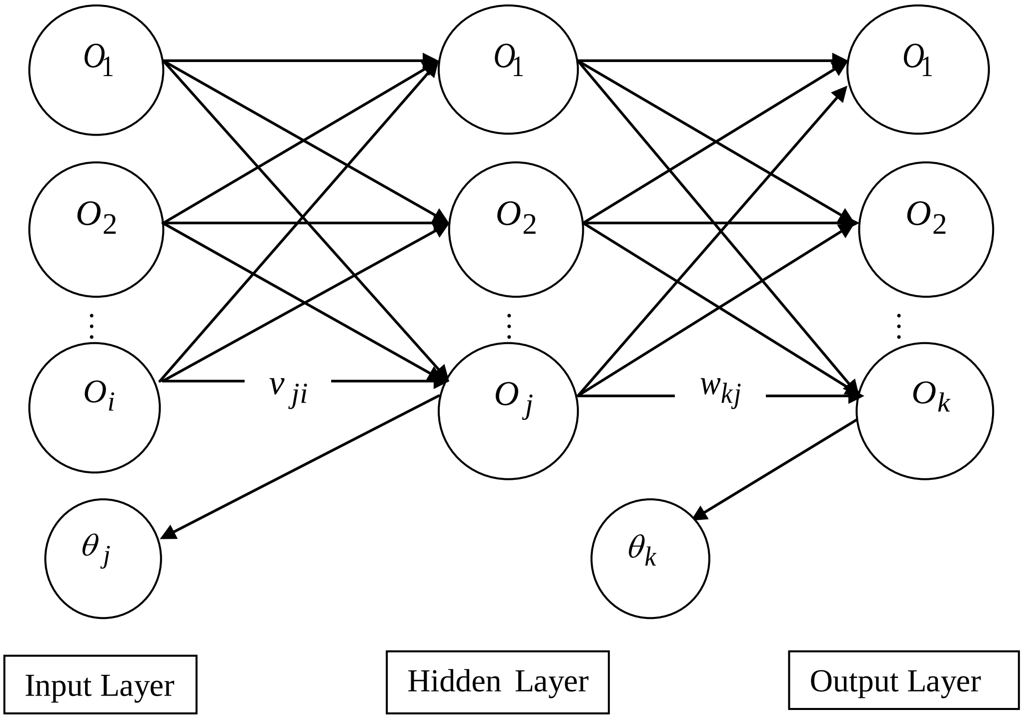

Figure 2.4: Architecture of multi-layer feed forward neural network

Let us consider a multi-layer neural architecture containing i input nodes O_i, j nodes in the hidden layer O_j and k output node O_k. Now by applying feed forward recall with error back propagation learning for above model (Figure 2.4) we have the following algorithm from Zurada [186].

Step 1: Initialize the weights v_{ji} from input to hidden layer and w_{kj} form hidden to output layer. Choose the learning parameter η (lies between 0, 1) and error E. Initially error is taken as E=0.

Step 2: Training steps start here

Outputs of the hidden layer and output layer are computed as below

$$O_j \leftarrow f(v_{ji} O_i) + \theta_j, \qquad j = 1,2,..., n \tag{2.4}$$

$$O_k \leftarrow f(w_{kj} O_j) + \theta_k, \qquad k = 1,2,..., n \tag{2.5}$$

f is the activation function

Step 3: Error value is computed as

$$E = \frac{1}{2}(d_k - O_k)^2 \tag{2.6}$$

Here d_k is the desired output, O_k is output of ANN.

Step 4: The error signal terms of the output and hidden layer are computed as

$$\delta_{ok} = \lfloor (d_k - O_k) f'(w_{kj} O_j) \rfloor \qquad \text{(Error signal of output layer)} \tag{2.7}$$

$$\delta_{oj} = \lfloor (1 - O_j) f'(v_{ji} O_i) \rfloor \delta_{ok} w_{kj} \quad \text{(Error signal of hidden layer)} \tag{2.8}$$

where $O_k = f(w_{kj} O_j),\quad j = 1,2,..., n$ and $k = 1,2,..., n$

Step 5: Compute components of error gradient vector as

$$\frac{\partial E}{\partial v_{ji}} = \delta_{oj} O_i \text{ for } j\text{=1,2,…,}n \text{ and } i\text{=1,…,}n \tag{2.9}$$

(For the particular ANN model Figure 2.4)

$$\frac{\partial E}{\partial w_{kj}} = \delta_{ok} O_j \text{ for } j\text{=1,2,…,}n \text{ and } k\text{=1,…,}n \quad \text{(For Figure 2.4)} \tag{2.10}$$

Step 6: Weights are modified using gradient descent method from input to hidden and from hidden to output layer as

$$v_{ji}^{(New)} = v_{ji}^{(Old)} + \Delta v_{ji} = v_{ji}^{(Old)} + \left(-\eta \frac{\partial E}{\partial v_{ji}} \right) \tag{2.11}$$

$$w_{kj}^{(New)} = w_{kj}^{(Old)} + \Delta w_{kj} = w_{kj}^{(Old)} + \left(-\eta \frac{\partial E}{\partial w_{kj}} \right) \tag{2.12}$$

Similarly the biases θ_j and θ_k are also updated.

The generalized delta learning rule propagates the error back by one layer, allowing the same process to be repeated for every layer.

The concept of interval along with its interval computations, fuzzy numbers and different types of fuzzy numbers are defined below which may help the readers for better understanding. These basic concepts may be found in few books like Jaulin et al. [189], Zimmerman [190], Ross [191], Lee [192], Moore [193], Chakraverty [195] and Chakraverty et al. [196-197].

2.2 Basic Definitions

2.2.1 Interval

An interval $\tilde{x}$ is denoted by $[\underline{x}, \overline{x}]$ on the set of real numbers *R* given by

$$\tilde{x} = [\underline{x}, \overline{x}] = \left\{x \in R : \underline{x} \le x \le \overline{x}\right\}. \tag{2.13}$$

Here we have only considered closed intervals throughout this thesis, although there exists various other types of intervals such as open, half open intervals etc. Here, $\underline{x}$ and $\overline{x}$ are known as the left and right endpoints of the interval $\tilde{x}$ in the above expression (2.13) respectively.

Let us now consider two arbitrary intervals $\tilde{x} = \left[\underline{x}, \overline{x}\right]$ and $\tilde{y} = \left[\underline{y}, \overline{y}\right]$. These two intervals are said to be equal if they are in the same set. Mathematically it only happens when corresponding end points are equal. Hence one may write

$$\tilde{x} = \tilde{y} \text{ if and only if } \underline{x} = \underline{y} \text{ and } \overline{x} = \overline{y}. \tag{2.14}$$

For the above two arbitrary intervals $\tilde{x} = \left[\underline{x}, \overline{x}\right]$ and $\tilde{y} = \left[\underline{y}, \overline{y}\right]$, interval arithmetic operations such as addition (+), subtraction (-), multiplication $(\times)$ and division (/) are defined as follows:

$$\tilde{x} + \tilde{y} = \left[\underline{x} + \underline{y},\ \overline{x} + \overline{y}\right], \tag{2.15}$$

$$\tilde{x} - \tilde{y} = \left[\underline{x} - \overline{y},\ \overline{x} - \underline{y}\right], \tag{2.16}$$

$$\tilde{x} \times \tilde{y} = \left[\min S, \max S\right], \text{ where } \mathrm{S} = \left\{\underline{x} \times \underline{y}, \underline{x} \times \overline{y}, \overline{x} \times \underline{y}, \overline{x} \times \overline{y}\right\}, \tag{2.17}$$

$$\text{and } \tilde{x} / \tilde{y} = \left[\underline{x}, \overline{x}\right] \times \left[\frac{1}{\overline{y}}, \frac{1}{\underline{y}}\right] \text{ if } 0 \notin \tilde{y} \tag{2.18}$$

Now if k is a real number and $\tilde{x} = \left[\underline{x}, \overline{x}\right]$ is an interval then the multiplication are given by

$$k\tilde{x} = \begin{cases} \left[k\overline{x}, k\underline{x}\right], k < 0, \\ \left[k\underline{x}, k\overline{x}\right], k \geq 0. \end{cases} \tag{2.19}$$

2.2.3 Fuzzy Set

If X is a collection of objects denoted generically by *x*, then a fuzzy set $\hat{U}$ in X is a set of ordered pairs:

$$\hat{U} = \left\{\left(x, \mu_{\hat{U}}(x) \middle| x \in X\right)\right\} \tag{2.20}$$

where, $\mu_{\hat{U}}$ is called the membership function

2.2.4 Fuzzy Number

A fuzzy number $\hat{U}$ is convex normalized fuzzy set $\hat{U}$ of the real line R such that

$$\left\{\mu_{\hat{U}}(x) : R \to [0,1], \forall\, x \in R\right\} \tag{2.21}$$

where, $\mu_{\hat{U}}$ is called the membership function and it is piecewise continuous.

There exists variety of fuzzy numbers. But in this study we have used only the triangular and trapezoidal fuzzy numbers. So, we define these two fuzzy numbers below.

2.2.5 Triangular Fuzzy Number (TFN)

A triangular fuzzy number $\hat{U}$ is a convex normalized fuzzy set $\hat{U}$ of the real line *R* such that

- There exists exactly one $x_0 \in R$ with $\mu_{\hat{U}}(x_0) = 1$ (x_0 is called the mean value of *U*), where $\mu_{\hat{U}}$ is called the membership function of the fuzzy set.
- $\mu_{\hat{U}}(x)$ is piecewise continuous.

Let us consider an arbitrary triangular fuzzy number $\hat{U} = (a, b, c)$ as shown in Figure 2.5. The membership function $\mu_{\hat{U}}$ of $\hat{U}$ is defined as follows

$$\mu_{\hat{U}}(x) = \begin{cases} 0, & x \leq a \\ \dfrac{x-a}{b-a}, & a \leq x \leq b \\ \dfrac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x \geq c. \end{cases} \tag{2.22}$$

The triangular fuzzy number $\hat{U} = (a,b,c)$ can be represented with an ordered pair of functions through h-cut approach viz.

$$\left[\lfloor \underline{u} \rfloor_h , \lfloor \overline{u} \rfloor_h \right] = \left[(b-a)h + a, -(c-b)h + c \right] \text{ where, } h \in [0,1]$$

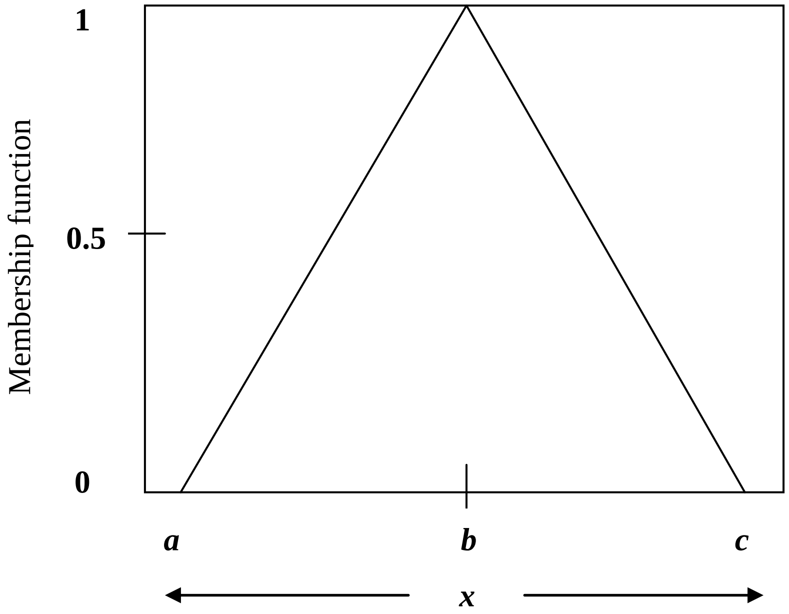

Figure 2.5: Triangular Fuzzy Number

2.2.6 Trapezoidal Fuzzy Number (TrFN)

We now introduce an arbitrary trapezoidal fuzzy number $\hat{U} = (a,b,c,d)$ as given in Figure 2.6. The membership function $\mu_{\hat{U}}$ of $\hat{U}$ will be interpreted as follows

$$\mu_{\hat{U}}(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \\ 0, & x \ge c. \end{cases} \tag{2.23}$$

The trapezoidal fuzzy number $\hat{U} = (a,b,c,d)$ can be represented with an ordered pair of functions through h-cut approach as

$$\left[\lfloor \underline{u} \rfloor_h , \lfloor \overline{u} \rfloor_h \right] = \left[(b-a)h + a, -(d-c)h + d \right] \text{ where, } h \in [0,1]$$

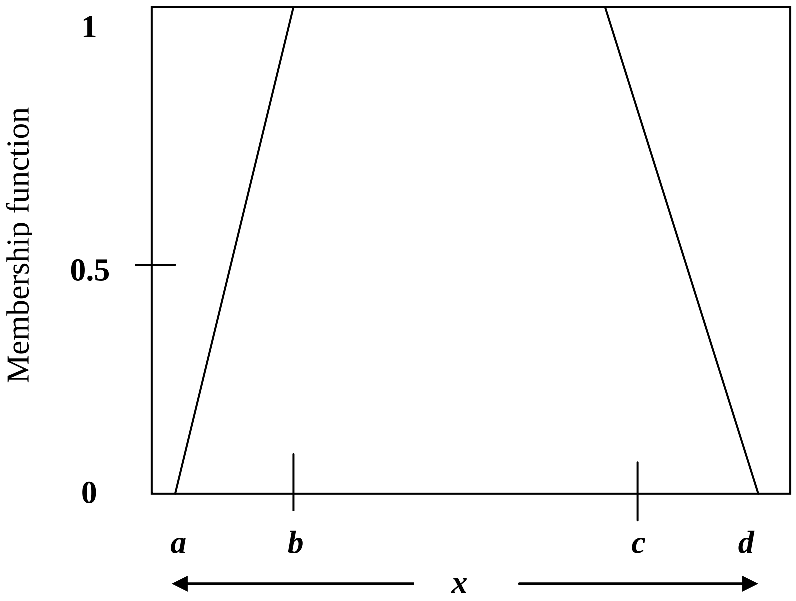

Figure 2.6: Trapezoidal Fuzzy Number

2.3 Algorithm for Interval Neural Network (INN)

A Neural Network is said to be an Interval Neural Network (INN) if at least one of its input, output and weights or all of them are in Interval form. The multilayer interval neural network has three layers. The first layer is considered to be input layer and the last layer is the output layer and in between the input and output layers, there may be more than one hidden layers. It is well known that neural network basically depends upon the type of processing elements or nodes, the network topology and the learning algorithm. Each layer will contain a number of neurons or nodes (processing elements) depending upon the problem. The processing elements are connected to each other by adjustable weights. The input/output behaviour of the network changes if the weights are changed. So, the weights of the net may be chosen in such a way so as to achieve a desired output. To satisfy this goal, error back propagation training algorithm for adjusting the weights has been developed. The error back propagation training algorithm (EBPTA) of interval neural network is similar to the traditional artificial neural network. In INN as the input, output and the weights are in interval therefore the training or learning algorithm is computed based on the interval computation. The interval weights and interval biases are also calculated based on above interval computations. The error back propagation training algorithm (EBPTA) of interval neural network (INN) is described below as in Ishibuchi and Tanaka [101], Ishibuchi et al. [121] and Ishibuchi et al. [102].

A multilayer neural network has been considered where the input, output, weights and biases have values in interval form. Here, $\tilde{O}_i, \tilde{O}_j, \tilde{O}_k$ are considered to be the input, hidden and output units. The target vector corresponding to the interval output vector is given as $\tilde{T}_k$. The weights between input and hidden layers are denoted by $\tilde{v}_{ji} = \left[\underline{v}_{ji}, \overline{v}_{ji}\right]$, the weights between hidden and output layers are denoted by $\tilde{w}_{kj} = \left[\underline{w}_{kj}, \overline{w}_{kj}\right]$ and $\tilde{\theta}_j$ and $\tilde{\theta}_k$ are interval biases. The input-output relation of each unit of the interval neural network is calculated using interval computation.

The inputs $\tilde{O}_i$ in the input units are written as

Input units: $\tilde{O}_i = \left[\left[\underline{O}_i\right], \left[\overline{O}_i\right]\right] = \left[\tilde{x}_i\right] = \left[\left[\underline{x}_i\right], \left[\overline{x}_i\right]\right], \quad i = 1,2,\ldots,n$

The output $\tilde{O}_j$ of the j-th hidden units in the hidden layer is calculated as

$$\left[\tilde{O}_j\right] = \left[\left[\underline{O}_j\right], \left[\overline{O}_j\right]\right] = f\left(\left[\tilde{Net}_j\right]\right) = \left[f\left(\left[\underline{Net}_j\right]\right), f\left(\left[\overline{Net}_j\right]\right)\right], \quad j = 1,2,\ldots,n$$

$$\left[\tilde{Net}_j\right] = \sum_{i=1}^{n} \left[\tilde{v}_{ji}\right] . \left[\tilde{O}_i\right] + \left[\tilde{\theta}_j\right], \quad j = 1,2,\ldots,n$$

Hidden units: $$\left[\underline{Net}_j\right] = \sum_{\substack{i=1 \\ \underline{v}_{ji} \ge 0}}^{n} \left[\underline{v}_{ji}\right] \cdot \left[\underline{O}_i\right] + \sum_{\substack{i=1 \\ \underline{v}_{ji} < 0}}^{n} \left[\underline{v}_{ji}\right] \cdot \left[\overline{O}_i\right] + \left[\underline{\theta}_j\right], \quad j = 1,2,\ldots,n$$

$$\left[\overline{Net}_j\right] = \sum_{\substack{i=1 \\ \overline{v}_{ji} \ge 0}}^{n} \left[\overline{v}_{ji}\right] \cdot \left[\overline{O}_i\right] + \sum_{\substack{i=1 \\ \overline{v}_{ji} < 0}}^{n} \left[\overline{v}_{ji}\right] \cdot \left[\underline{O}_i\right] + \left[\overline{\theta}_j\right], \quad j = 1,2,\ldots,n$$

The output $\tilde{O}_k$ from output units is then evaluated as

$$\left[\tilde{O}_j\right] = \left[\left[\underline{O}_j\right], \left[\overline{O}_j\right]\right] = f\left(\left[\tilde{Net}_j\right]\right) = \left[f\left(\left[\underline{Net}_j\right]\right), f\left(\left[\overline{Net}_j\right]\right)\right], \quad j = 1,2,\ldots,n$$

$$\left[\tilde{Net}_j\right] = \sum_{i=1}^{n} \left[\tilde{v}_{ji}\right] . \left[\tilde{O}_i\right] + \left[\tilde{\theta}_j\right], \quad j = 1,2,\ldots,n$$

Output units: $$\left[\underline{Net}_j\right] = \sum_{\substack{i=1 \\ \underline{v}_{ji} \ge 0}}^{n} \left[\underline{v}_{ji}\right] \cdot \left[\underline{O}_i\right] + \sum_{\substack{i=1 \\ \underline{v}_{ji} < 0}}^{n} \left[\underline{v}_{ji}\right] \cdot \left[\overline{O}_i\right] + \left[\underline{\theta}_j\right], \quad j = 1,2,\ldots,n$$

$$\left[\overline{Net}_j\right] = \sum_{\substack{i=1 \\ \overline{v}_{ji} \ge 0}}^{n} \left[\overline{v}_{ji}\right] \cdot \left[\overline{O}_i\right] + \sum_{\substack{i=1 \\ \overline{v}_{ji} < 0}}^{n} \left[\overline{v}_{ji}\right] \cdot \left[\underline{O}_i\right] + \left[\overline{\theta}_j\right], \quad j = 1,2,\ldots,n$$

Here f is the unipolar activation function defined by $f\left(\tilde{Net}\right) = \dfrac{1}{\left[1 + \exp\left(-\gamma\,\tilde{Net}\right)\right]}$

The cost function is computed using interval arithmetic as

$$\tilde{E} = \underline{E} + \overline{E} = \sum_{k=1}^{n}\left\{\frac{([\underline{T}_k]-[\underline{O}_k])^2}{2} + \frac{([\overline{T}_k]-[\overline{O}_k])^2}{2}\right\}$$

From the cost function a learning rule can be derived for the interval weights $\tilde{w}_{kj}$ between the output and the hidden layer. The interval weights are updated as,

$$\tilde{w}_{kj}^{(New)} = \left[\underline{w}_{kj}^{(New)}, \overline{w}_{kj}^{(New)}\right] = \left[\underline{w}_{kj}^{(Old)}, \overline{w}_{kj}^{(Old)}\right] + \left[\Delta\underline{w}_{kj}, \Delta\overline{w}_{kj}\right]$$

where change in weights are calculated as

$$\Delta\tilde{w}_{kj} = \left[\Delta\underline{w}_{kj}, \Delta\overline{w}_{kj}\right] = \left[-\eta\frac{\partial E}{\partial\tilde{w}_{kj}}\right] = \left[-\eta\frac{\partial E}{\partial\underline{w}_{kj}}, -\eta\frac{\partial E}{\partial\overline{w}_{kj}}\right]$$

Consequently, output layer weights $\tilde{v}_{ji}$ between the hidden layer and the input layer are adjusted as,

$$\tilde{v}_{ji}^{(New)} = \left[\underline{v}_{ji}^{(New)}, \overline{v}_{ji}^{(New)}\right] = \left[\underline{v}_{ji}^{(Old)}, \overline{v}_{ji}^{(Old)}\right] + \left[\Delta\underline{v}_{ji}, \Delta\overline{v}_{ji}\right]$$

where change in weights are calculated as

$$\Delta\tilde{v}_{ji} = \left[\Delta\underline{v}_{ji}, \Delta\overline{v}_{ji}\right] = \left[-\eta\frac{\partial E}{\partial\tilde{v}_{ji}}\right] = \left[-\eta\frac{\partial E}{\partial\underline{v}_{ji}}, -\eta\frac{\partial E}{\partial\overline{v}_{ji}}\right]$$

and η is the learning constant. The complete derivation of weight updation has been described below.

2.3.1 Derivation of the Learning Algorithm for Interval Case

2.3.1.1 Calculation of $\frac{\partial E}{\partial\underline{w}_{kj}}$:

I. If $\underline{w}_{kj} \ge 0$ then

$$\frac{\partial E}{\partial\underline{w}_{kj}} = \frac{\partial\underline{E}}{\partial\underline{w}_{kj}} = \frac{\partial}{\partial\underline{O}_k}\left\{\frac{1}{2}(\underline{T}_k-\underline{O}_k)^2\right\}\cdot\frac{\partial\underline{O}_k}{\partial[\underline{Net}_k]}\cdot\frac{\partial[\underline{Net}_k]}{\partial\underline{w}_{kj}} = -\underline{\delta}_k\cdot[\underline{O}_j]$$

where

$$\underline{\delta}_k = -\frac{\partial E}{\partial[\underline{Net}_k]} = ([\underline{T}_k]-[\underline{O}_k])\cdot[\underline{O}_k]\cdot(1-[\underline{O}_k])$$

II. If $\underline{w}_{kj} < 0$ then

$$\frac{\partial E}{\partial \underline{w}_{kj}} = \frac{\partial \underline{E}}{\partial \underline{w}_{kj}} = -\underline{\delta}_k \cdot \left[\overline{O}_j\right]$$

2.3.1.2 Calculation of $\frac{\partial E}{\partial \overline{w}_{kj}}$:

I. If $\overline{w}_{kj} \geq 0$ then

$$\frac{\partial E}{\partial \overline{w}_{kj}} = \frac{\partial \overline{E}}{\partial \overline{w}_{kj}} = \frac{\partial}{\partial \overline{O}_k}\left\{\frac{1}{2}\left(\overline{T}_k - \overline{O}_k\right)^2\right\} \cdot \frac{\partial \overline{O}_k}{\partial\left[\overline{Net}_k\right]} \cdot \frac{\partial\left[\overline{Net}_k\right]}{\partial \overline{w}_{kj}} = -\overline{\delta}_k \cdot \left[\overline{O}_j\right]$$

where

$$\overline{\delta}_k = -\frac{\partial E}{\partial\left[\overline{Net}_k\right]} = \left(\left[\overline{T}_k\right] - \left[\overline{O}_k\right]\right) \cdot \left[\overline{O}_k\right] \cdot \left(1 - \left[\overline{O}_k\right]\right)$$

II. If $\overline{w}_{kj} < 0$ then

$$\frac{\partial E}{\partial \overline{w}_{kj}} = \frac{\partial \overline{E}}{\partial \overline{w}_{kj}} = -\overline{\delta}_k \cdot \left[\underline{O}_j\right]$$

2.3.1.3 Calculation of $\frac{\partial E}{\partial \underline{v}_{ji}}$:

I. If $\overline{v}_{ji} \geq \underline{v}_{ji} \geq 0$ then

$$\frac{\partial E}{\partial \underline{v}_{ji}} = \frac{\partial \underline{E}}{\partial \underline{v}_{ji}} = -\delta^A \cdot \left[\underline{O}_j\right] \cdot \left(1 - \left[\underline{O}_j\right]\right) \cdot \left[\underline{O}_i\right]$$

where

$$\delta^A = \sum_{\substack{k=1 \\ \underline{w}_{kj} \geq 0}}^{n} \underline{\delta}_k \cdot \underline{w}_{kj} + \sum_{\substack{k=1 \\ \overline{w}_{kj} < 0}}^{n} \overline{\delta}_k \cdot \underline{w}_{kj}$$

II. If $\underline{v}_{ji} \leq \overline{v}_{ji} < 0$ then

$$\frac{\partial E}{\partial \underline{v}_{ji}} = \frac{\partial \overline{E}}{\partial \underline{v}_{ji}} = -\delta^A \cdot \left[\underline{O}_j\right] \cdot \left(1 - \left[\underline{O}_j\right]\right) \cdot \left[\overline{O}_i\right]$$

III. If $\underline{v}_{ji} < 0 \leq \overline{v}_{ji}$ then

$$\frac{\partial E}{\partial \underline{v}_{ji}} = 0$$

2.3.1.4 Calculation of $\frac{\partial E}{\partial \bar{v}_{ji}}$:

I. If $\bar{v}_{ji} \ge \underline{v}_{ji} \ge 0$ then

$$\frac{\partial E}{\partial \bar{v}_{ji}} = \frac{\partial \bar{E}}{\partial \bar{v}_{ji}} = -\delta^B \cdot \left[\bar{O}_j\right] \cdot \left(1 - \left[\bar{O}_j\right]\right) \cdot \left[\bar{O}_i\right]$$

where

$$\delta^B = \sum_{\substack{k=1 \\ \underline{w}_{kj}<0}}^{n} \underline{\delta}_k \cdot \underline{w}_{kj} + \sum_{\substack{k=1 \\ \bar{w}_{kj}\ge 0}}^{n} \bar{\delta}_k \cdot \bar{w}_{kj}$$

II. If $\underline{v}_{ji} \le \bar{v}_{ji} < 0$ then

$$\frac{\partial E}{\partial \bar{v}_{ji}} = \frac{\partial \underline{E}}{\partial \bar{v}_{ji}} = -\delta^B \cdot \left[\bar{O}_j\right] \cdot \left(1 - \left[\bar{O}_j\right]\right) \cdot \left[\underline{O}_i\right]$$

III. If $\underline{v}_{ji} < 0 \le \bar{v}_{ji}$ then

$$\frac{\partial E}{\partial \bar{v}_{ji}} = \frac{\partial \underline{E}}{\partial \bar{v}_{ji}} + \frac{\partial \bar{E}}{\partial \bar{v}_{ji}} - \delta^B \cdot \left[\bar{O}_j\right] \cdot \left(1 - \left[\bar{O}_j\right]\right) \cdot \left[\underline{O}_i\right] - \delta^B \cdot \left[\bar{O}_j\right] \cdot \left(1 - \left[\bar{O}_j\right]\right) \cdot \left[\bar{O}_i\right]$$

While modifying $\underline{v}_{ji}$, $\bar{v}_{ji}$ and $\underline{w}_{kj}$, $\bar{w}_{kj}$ it is undesirable but possible sometimes that $\underline{v}_{ji} > \bar{v}_{ji}$ and $\underline{w}_{kj} > \bar{w}_{kj}$. In order to cope with this situation, the interval weights from input to hidden layer and from hidden to output layer are determined as,

$$\tilde{v}_{ji}^{(New)} = \left[\min\left\{\underline{v}_{ji}^{(New)}, \bar{v}_{ji}^{(New)}\right\}, \max\left\{\underline{v}_{ji}^{(New)}, \bar{v}_{ji}^{(New)}\right\}\right]$$

$$\tilde{w}_{kj}^{(New)} = \left[\min\left\{\underline{w}_{kj}^{(New)}, \bar{w}_{kj}^{(New)}\right\}, \max\left\{\underline{w}_{kj}^{(New)}, \bar{w}_{kj}^{(New)}\right\}\right]$$

In the similar fashion the interval biases $\tilde{\theta}_j$ and $\tilde{\theta}_k$ are also updated.

2.4 Algorithm for Fuzzy Neural Network (FNN)

A Neural Network is said to be a Fuzzy Neural Network if at least one of its input, output and weights or all have values in fuzzified form. The error back propagation algorithm is also developed to handle fuzziness.

A three layer neural network has been taken where the inputs, hidden and outputs units in fuzzified form are considered as $\hat{O}_i, \hat{O}_j$ and $\hat{O}_k$. The inputs, hidden and outputs in h-level form is written as $\left[\hat{O}_i\right]_h, \left[\hat{O}_j\right]_h$ and $\left[\hat{O}_k\right]_h$. The target vector corresponding to the fuzzy output vector is given as $\hat{T}_k$. But in h-level form the target vector is denoted as $\left[\hat{T}_k\right]_h$. The weights between input and hidden layers are denoted by $\hat{v}_{ji} = \left[\underline{v}_{ji}, v_{ji}c, \overline{v}_{ji}\right]$, the weights between hidden and output layers are denoted by $\hat{w}_{kj} = \left[\underline{w}_{kj}, w_{kj}c, \overline{w}_{kj}\right]$ and $\hat{\theta}_j$ and $\hat{\theta}_k$ are the biases which all are in fuzzified form. The weights and biases in h-level form may be written as $\left[\hat{v}_{ji}\right]_h = \left[\left[\underline{v}_{ji}\right]_h, \left[\overline{v}_{ji}\right]_h\right], \left[\hat{w}_{kj}\right]_h = \left[\left[\underline{w}_{kj}\right]_h, \left[\overline{w}_{kj}\right]_h\right], \left[\hat{\theta}_j\right]_h$ and $\left[\hat{\theta}_k\right]_h$. The input-output relation of each unit of the fuzzified neural network is calculated using fuzzy operations on the h-level sets of the fuzzy inputs, fuzzy weights and fuzzy biases. The algorithm written below has been followed from the papers Ishibuchi et al. [122], Ishibuchi et al. [123] and Ishibuchi et al. [124-125].

The inputs $\left[\hat{O}_i\right]_h$ in the input units are written as

Input units: $\left[\hat{O}_i\right]_h = \left[\left[\underline{O}_i\right]_h,\ \left[\overline{O}_i\right]_h\right], \quad i = 1,2,...,n$

The output $\left[\hat{O}_j\right]_h$ of the j-th hidden units in the hidden layer is calculated as

Hidden units:

$$\left[\hat{O}_j\right]_h = \left[\left[\underline{O}_j\right]_h,\ \left[\overline{O}_j\right]_h\right] = \left[f\left(\left[\underline{Net}_j\right]_h\right), f\left(\left[\overline{Net}_j\right]_h\right)\right] \quad j = 1,2,...,n$$

$$\left[\underline{Net}_j\right]_h = \sum_{\substack{i=1 \\ \left[\underline{v}_{ji}\right]_h \geq 0}}^{n} \left[\underline{v}_{ji}\right]_h \cdot \left[\underline{O}_i\right]_h + \sum_{\substack{i=1 \\ \left[\underline{v}_{ji}\right]_h < 0}}^{n} \left[\underline{v}_{ji}\right]_h \cdot \left[\overline{O}_i\right]_h + \left[\underline{\theta}_j\right]_h, \quad j = 1,2,...,n$$

$$\left[\overline{Net}_j\right]_h = \sum_{\substack{i=1 \\ \left[\overline{v}_{ji}\right]_h \geq 0}}^{n} \left[\overline{v}_{ji}\right]_h \cdot \left[\overline{O}_i\right]_h + \sum_{\substack{i=1 \\ \left[\overline{v}_{ji}\right]_h < 0}}^{n} \left[\overline{v}_{ji}\right]_h \cdot \left[\underline{O}_i\right]_h + \left[\overline{\theta}_j\right]_h, \quad j = 1,2,...,n$$

The output $\left[\tilde{O}_k\right]_h$ from output units is then evaluated as

Output units:

$$\left[\hat{O}_k\right]_h = \left[\left[\underline{O}_k\right]_h, \left[\overline{O}_k\right]_h\right] = \left[f\left(\left[\underline{Net}_k\right]_h\right), f\left(\left[\overline{Net}_k\right]_h\right)\right], \quad k = 1,2,..., n$$

$$\left[\underline{Net}_k\right]_h = \sum_{\substack{j=1 \\ \left[\underline{w}_{kj}\right]_h \geq 0}}^{n} \left[\underline{w}_{kj}\right]_h \cdot \left[\underline{O}_j\right]_h + \sum_{\substack{j=1 \\ \left[\underline{w}_{kj}\right]_h < 0}}^{n} \left[\underline{w}_{kj}\right]_h \cdot \left[\overline{O}_j\right]_h + \left[\underline{\theta}_k\right]_h, \quad k = 1,2,..., n$$

$$\left[\overline{Net}_k\right]_h = \sum_{\substack{j=1 \\ \left[\overline{w}_{kj}\right]_h \geq 0}}^{n} \left[\overline{w}_{kj}\right]_h \cdot \left[\overline{O}_j\right]_h + \sum_{\substack{j=1 \\ \left[\overline{w}_{kj}\right]_h < 0}}^{n} \left[\overline{w}_{ki}\right]_h \cdot \left[\underline{O}_j\right]_h + \left[\overline{\theta}_k\right]_h, \quad k = 1,2,..., n$$

The cost function for *h*-level sets is computed based on fuzzy computations

$$E = \sum_{k=1}^{n} \left\{ \frac{h \cdot \left(\left[\underline{T}_k\right]_h - \left[\underline{O}_k\right]_h\right)^2}{2} + \frac{h \cdot \left(\left[\overline{T}_k\right]_h - \left[\overline{O}_k\right]_h\right)^2}{2} \right\}$$

From the cost function a learning rule can be derived for the fuzzy weights $\left[\hat{w}_{kj}\right]_h$ between the output and the hidden layer. The fuzzy weights are updated as,

$$\left[\hat{w}_{kj}^{(new)}\right]_h = \left[\left[\underline{w}_{kj}^{(new)}\right]_h, \left[\overline{w}_{kj}^{(new)}\right]_h\right] = \left[\left[\underline{w}_{kj}^{(old)}\right]_h, \left[\overline{w}_{kj}^{(old)}\right]_h\right] + \left[\left[\Delta \underline{w}_{kj}\right]_h, \left[\Delta \overline{w}_{kj}\right]_h\right]$$

where change in weights are calculated as

$$\left[\Delta \hat{w}_{kj}\right]_h = \left[[\Delta \underline{w}_{kj}]_h, [\Delta \overline{w}_{kj}]_h\right] = \left[-\eta \frac{\partial E}{\partial \hat{w}_{kj}}\right] = \left[-\eta \frac{\partial E}{\partial [\underline{w}_{kj}]_h}, -\eta \frac{\partial E}{\partial [\overline{w}_{kj}]_h}\right]$$

Consequently, output layer weights $\hat{v}_{ji}$ between the hidden layer and the input layer are adjusted as,

$$\left[\hat{v}_{ji}^{(New)}\right]_h = \left[\left[\underline{v}_{ji}^{(New)}\right]_h, \left[\overline{v}_{ji}^{(New)}\right]_h\right] = \left[\left[\underline{v}_{ji}^{(Old)}\right]_h, \left[\overline{v}_{ji}^{(Old)}\right]_h\right] + \left[\left[\Delta \underline{v}_{ji}\right]_h, \left[\Delta \overline{v}_{ji}\right]_h\right]$$

where change in weights are calculated as

$$\left[\Delta \hat{v}_{ji}\right]_h = \left[\left[\Delta \underline{v}_{ji}\right]_h, \left[\Delta \overline{v}_{ji}\right]_h\right] = \left[-\eta \frac{\partial E}{\partial \hat{v}_{ji}}\right] = \left[-\eta \frac{\partial E}{\partial [\underline{v}_{ji}]_h}, -\eta \frac{\partial E}{\partial [\overline{v}_{ji}]_h}\right]$$

and η is the learning parameter. In the similar fashion the fuzzy biases $\hat{\theta}_j$ and $\hat{\theta}_k$ are also updated. The complete derivation for updating of weights has been described below.

2.4.1 Derivation of the Learning Algorithm for Fuzzy Case

2.4.1.1 Calculation of $\dfrac{\partial E}{\partial[\underline{w}_{kj}]_h}$:

I. If $[\underline{w}_{kj}]_h \geq 0$ then

$$\frac{\partial E}{\partial[\underline{w}_{kj}]_h} = \frac{\partial \underline{E}}{\partial[\underline{w}_{kj}]_h} = \frac{\partial}{\partial[\underline{O}_k]_h}\left\{\frac{1}{2}\left([\underline{T}_k]_h - [\underline{O}_k]_h\right)^2\right\} \cdot \frac{\partial[\underline{O}_k]_h}{\partial[\underline{Net}_k]_h} \cdot \frac{\partial[\underline{Net}_k]_h}{\partial[\underline{w}_{kj}]_h} = -\left[\underline{\delta}_k\right]_h \cdot \left[\underline{O}_j\right]_h$$

where

$$\left[\underline{\delta}_k\right]_h = -\frac{\partial E}{\partial[\underline{Net}_k]_h} = \left([\underline{T}_k]_h - [\underline{O}_k]_h\right) \cdot \left[\underline{O}_k\right]_h \cdot \left(1 - \left[\underline{O}_k\right]_h\right)$$

II. If $[\underline{w}_{kj}]_h < 0$ then

$$\frac{\partial E}{\partial[\underline{w}_{kj}]_h} = \frac{\partial \underline{E}}{\partial[\underline{w}_{kj}]_h} = \frac{\partial \underline{E}}{\partial[\underline{Net}_{pk}]_h} . \frac{\partial[\underline{Net}_k]_h}{\partial[\underline{w}_{kj}]_h} = -\left[\underline{\delta}_k\right]_h \cdot \left[\overline{O}_j\right]_h$$

2.4.1.2 Calculation of $\dfrac{\partial E}{\partial[\overline{w}_{kj}]_h}$:

I. If $[\overline{w}_{kj}]_h \geq 0$ then

$$\frac{\partial E}{\partial[\overline{w}_{kj}]_h} = \frac{\partial \overline{E}}{\partial[\overline{w}_{kj}]_h} = \frac{\partial}{\partial[\overline{O}_k]_h}\left\{\frac{1}{2}\left(\left[\overline{T}_k\right]_h - \left[\overline{O}_k\right]_h\right)^2\right\} \cdot \frac{\partial\left[\overline{O}_k\right]_h}{\partial[\overline{Net}_{pk}]_h} \cdot \frac{\partial[\overline{Net}_k]_h}{\partial[\overline{w}_{kj}]_h} = -\left[\overline{\delta}_k\right]_h \cdot \left[\overline{O}_j\right]_h$$

where

$$\left[\overline{\delta}_k\right]_h = -\frac{\partial E}{\partial[\overline{Net}_k]_h} = h \cdot \left(\left[\overline{T}_k\right]_h - \left[\overline{O}_k\right]_h\right) \cdot \left[\overline{O}_k\right]_h \cdot \left(1 - \left[\overline{O}_k\right]_h\right)$$

II. If $[\overline{w}_{kj}]_h < 0$ then

$$\frac{\partial E}{\partial \overline{w}_{kj}} = \frac{\partial \overline{E}}{\partial[\overline{w}_{kj}]_h} = \frac{\partial E}{\partial[\overline{Net}_k]_h} \cdot \frac{\partial[\overline{Net}_k]_h}{\partial[\overline{w}_{kj}]_h} = -\left[\overline{\delta}_k\right]_h \cdot \left[\underline{O}_j\right]_h$$

2.4.1.3 Calculation of $\dfrac{\partial E}{\partial[\underline{v}_{ji}]_h}$:

I. If $[\underline{v}_{ji}]_h \geq 0$ then

$$\frac{\partial E}{\partial[\underline{v}_{ji}]_h} = \frac{\partial \underline{E}}{\partial[\underline{v}_{ji}]_h} = -[\delta^A]_h \cdot \left[\underline{O}_j\right]_h \cdot \left(1 - \left[\underline{O}_j\right]_h\right) \cdot \left[\underline{O}_i\right]_h$$

where

$$[\delta^A]_h = \sum_{\substack{k=1 \\ [\underline{w}_{kj}]_h \geq 0}}^{n} \left[\underline{\delta}_k\right]_h \cdot [\underline{w}_{kj}]_h + \sum_{\substack{k=1 \\ [\overline{w}_{kj}]_h < 0}}^{n} \left[\overline{\delta}_k\right]_h \cdot [\overline{w}_{kj}]_h$$

II. If $[\underline{v}_{ji}]_h < 0$ then

$$\frac{\partial E}{\partial [\underline{v}_{ji}]_h} = \frac{\partial \overline{E}}{\partial [\underline{v}_{ji}]_h} = -[\delta^A]_h \cdot \left[\underline{O}_j\right]_h \cdot \left(1 - \left[\underline{O}_j\right]_h\right) \cdot \left[\overline{O}_i\right]_h$$

2.4.1.4 Calculation of $\frac{\partial E}{\partial [\overline{v}_{ji}]_h}$:

I. If $[\overline{v}_{ji}]_h \geq 0$ then

$$\frac{\partial E}{\partial [\overline{v}_{ji}]_h} = \frac{\partial \overline{E}}{\partial [\overline{v}_{ji}]_h} = -\left[\delta^B\right]_h \cdot \left[\overline{O}_j\right]_h \cdot \left(1 - \left[\overline{O}_j\right]_h\right) \cdot \left[\overline{O}_i\right]_h$$

where

$$\left[\delta^B\right]_h = \sum_{\substack{k=1 \\ [\underline{w}_{kj}]_h < 0}}^{n} \left[\underline{\delta}_k\right]_h \cdot [\underline{w}_{kj}]_h + \sum_{\substack{k=1 \\ [\overline{w}kj]_h \geq 0}}^{n} \left[\overline{\delta}_k\right]_h \cdot [\overline{w}_{kj}]_h$$

II. If $[\overline{v}_{ji}]_h < 0$ then

$$\frac{\partial E}{\partial [\overline{v}_{ji}]_h} = \frac{\partial \underline{E}}{\partial [\overline{v}_{ji}]_h} = -\left[\delta^B\right]_h \cdot \left[\overline{O}_j\right]_h \cdot \left(1 - \left[\overline{O}_j\right]_h\right) \cdot \left[\underline{O}_i\right]_h$$

Chapter 3

System Identification from Frequency Data Using Interval and Fuzzy Neural Network

In this chapter, we have proposed identification methodologies for multistorey shear buildings using the powerful technique of Artificial Neural Network (ANN) models which can handle interval and fuzzified data. Here, inputs are considered as the frequency parameters and outputs as the stiffness parameters in interval and fuzzified form. Examples have been given for various cases such as two, five and ten storey shear structure for crisp case, five and ten storey shear structure with interval data and finally one and two storey with fuzzified data. Testing has been done for five storey shear structure with interval data to validate the present methods.

3.1 Analysis and Modelling For Interval Case

System identification refers to the branch of numerical analysis which uses the experimental input and output data to develop mathematical models of systems which finally identify the parameters. The floor masses for this methodology are assumed to be $[\underline{m}_1, \overline{m}_1], [\underline{m}_2, \overline{m}_2], \ldots, [\underline{m}_n, \overline{m}_n]$ and the stiffness $[\underline{k}_1, \overline{k}_1], [\underline{k}_2, \overline{k}_2], \ldots, [\underline{k}_n, \overline{k}_n]$ are the structural parameters which are to be identified. It may be seen that all the mass and stiffness parameters are taken in interval form. As such here for each mass m_i we have $\underline{m}_i$ as the left value and $\overline{m}_i$ as the right value of the interval. Similarly for the stiffness parameter for each k_i we have $\underline{k}_i$ as the left value and $\overline{k}_i$ as the right value of the interval. The interval n-storey shear structure is shown in Figure 3.1. Corresponding dynamic equation of motion for n-storey (supposed as n degrees of freedom) shear structure without damping may be written as

$$\{\tilde{M}\}\{\ddot{\tilde{X}}\} + \{\tilde{K}\}\{\tilde{X}\} = \{\tilde{0}\} \tag{3.1}$$

where, $\{\ddot{\tilde{x}}\} = \{\underline{\ddot{x}}, \overline{\ddot{x}}\}$, $\{\tilde{x}\} = [\underline{x}, \overline{x}]$

$\{\tilde{M}\} = [\underline{M}, \overline{M}]$ is $n \times n$ mass matrix of the structure, and is given by

$$\{\tilde{M}\} = \begin{bmatrix} [\underline{m}_1, \overline{m}_1] & 0 & \cdots & \cdots & 0 \\ 0 & [\underline{m}_2, \overline{m}_2] & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & 0 & [\underline{m}_{n-1}, \overline{m}_{n-1}] & 0 \\ 0 & \cdots & \cdots & 0 & [\underline{m}_n, \overline{m}_n] \end{bmatrix};$$

$\{\tilde{K}\} = [\underline{K}, \overline{K}]$ is $n \times n$ stiffness matrix of the structure and may be written as

$$\{\tilde{K}\} = \begin{bmatrix} [\underline{k}_1, \overline{k}_1] + [\underline{k}_2, \overline{k}_2] & -[\underline{k}_2, \overline{k}_2] & 0 & \cdots & 0 \\ -[\underline{k}_2, \overline{k}_2] & [\underline{k}_2, \overline{k}_2] + [\underline{k}_3, \overline{k}_3] & -[\underline{k}_3, \overline{k}_3] & \cdots & 0 \\ \cdots & \cdots & & & \cdots \\ 0 & \cdots & -[\underline{k}_{n-1}, \overline{k}_{n-1}] & [\underline{k}_{n-1}, \overline{k}_{n-1}] + [\underline{k}_n, \overline{k}_n] & [\underline{k}_n, \overline{k}_n] \\ 0 & & \cdots & -[\underline{k}_n, \overline{k}_n] & [\underline{k}_n, \overline{k}_n] \end{bmatrix}$$

and $\{\tilde{X}\} = \{\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n\}^T$ is the displacement vector.

We have first solve the above free vibration equation for vibration characteristics namely, for frequency and mode shapes of the said structural system in order to get the stiffness parameters in interval form. Accordingly putting $\{\tilde{X}\} = \{\tilde{\phi}\}e^{i(\tilde{\omega})t}$ in free vibration equation (3.1) Chopra [199], we get

$$\left(\{\tilde{K}\} - \{\tilde{M}\}[\tilde{\omega}]^2\right)\{\tilde{\phi}\} = \{\tilde{0}\} \tag{3.2}$$

where $\{\tilde{\omega}\}^2 = [\underline{\omega}, \overline{\omega}]^2 = [\underline{\lambda}, \overline{\lambda}]$ are eigenvalues or the natural frequency and $\{\tilde{\phi}\}$ are mode shapes of the structure, respectively. These interval natural frequencies are evaluated from the free vibration equation using interval computation described below.

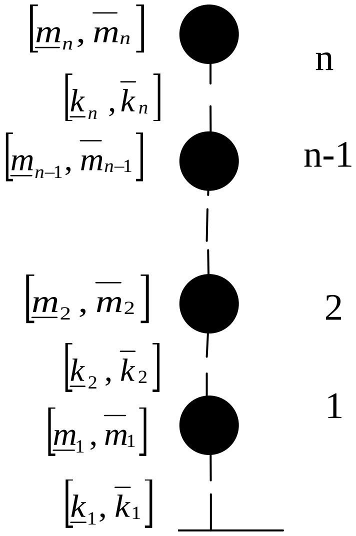
$[\underline{m}_n, \overline{m}_n]$
$[\underline{k}_n, \overline{k}_n]$
$[\underline{m}_{n-1}, \overline{m}_{n-1}]$
$[\underline{m}_2, \overline{m}_2]$
$[\underline{k}_2, \overline{k}_2]$
$[\underline{m}_1, \overline{m}_1]$
$[\underline{k}_1, \overline{k}_1]$
n
n-1
2
1

Figure 3.1: Multi-storey shear structure with n-levels having interval structural parameters

3.2 Interval Neural Network Model

This section describes structure of multi-layer interval neural network model. The training and learning algorithm have already been discussed in Sec. 2.7. Present interval neural network model is given in Figure 3.2. $\tilde{O}_i$, $\tilde{O}_j$ and $\tilde{O}_k$ are taken as input, hidden and output layers. In Figure 3.2, the inputs $\tilde{O}_i = \tilde{\lambda}_i = \left[\underline{\lambda}_i, \overline{\lambda}_i\right]$ are frequencies in interval and the outputs $\tilde{O}_k = \tilde{k}_k = \left[\underline{k}_k, \overline{k}_k\right]$ are stiffness parameters in interval form. The weights between input and hidden layers are denoted by $\tilde{v}_{ji} = \left[\underline{v}_{ji}, \overline{v}_{ji}\right]$ and the weights between hidden and output layers are denoted by $\tilde{w}_{kj} = \left[\underline{w}_{kj}, \overline{w}_{kj}\right]$ which are all in intervals. The total input to the jth hidden unit in the second layer can be calculated as

$$\tilde{U}_j = \left[\underline{U}_j, \overline{U}_j\right] = \left[\underline{v}_{ji}, \overline{v}_{ji}\right] \cdot \left[\underline{O}_i, \overline{O}_i\right] + \left[\underline{\theta}_j, \overline{\theta}_j\right] \tag{3.3}$$

where $\left[\underline{\theta}_j, \overline{\theta}_j\right]$ are the bias weights of the hidden layer. The right hand side of the above equation is computed by interval multiplication and interval addition defined in section Sec. 2.2. The output of the hidden unit is evaluated as

$$\tilde{O}_j = \left[\underline{O}_j, \overline{O}_j\right] = \left[f\left(\underline{U}_j\right), f\left(\overline{U}_j\right)\right] \tag{3.4}$$

where f is the unipolar activation function defined by $f\left(\tilde{Net}\right) = \dfrac{1}{1+\exp\left(-\gamma\,\tilde{Net}\right)}$. The total input from hidden to the output unit is calculated as

$$\tilde{Y}_k = \left[\underline{Y}_k, \overline{Y}_k\right] = \left[\underline{w}_{kj}, \overline{w}_{kj}\right] \cdot \left[\underline{O}_j, \overline{O}_j\right] + \left[\underline{\theta}_k, \overline{\theta}_k\right] \tag{3.5}$$

Again the right hand side of the above equation involves interval multiplication and interval addition, where $\left[\underline{\theta}_k, \overline{\theta}_k\right]$ are the bias weights of the output layer. Finally, the response of the net is given as

$$\tilde{O}_k = \left[\underline{O}_k, \overline{O}_k\right] = \left[f\left(\underline{Y}_k\right), f\left(\overline{Y}_k\right)\right] \tag{3.6}$$

The error value is computed as

$$E = \sum_{k=1}^{n} \left\{ \frac{1}{2}\left(\underline{d}_k - \underline{O}_k\right)^2 + \frac{1}{2}\left(\overline{d}_k - \overline{O}_k\right)^2 \right\} \tag{3.7}$$

From the cost function (3.7) a learning rule can be derived for the interval weights $\left(\tilde{v}_{ji}\right)$ between the hidden and input layer. The interval weights are updated as

$$\tilde{v}_{ji}^{(New)} = \left[\underline{v}_{ji}^{(New)}, \overline{v}_{ji}^{(New)}\right] = \left[\underline{v}_{ji}^{(Old)}, \overline{v}_{ji}^{(Old)}\right] + \left[\Delta\underline{v}_{ji}, \Delta\overline{v}_{ji}\right] \quad j=1,..,n \quad \text{and} \quad i=1,2,..,n \tag{3.8}$$

where change in weights are calculated as

$$\Delta\tilde{v}_{ji} = \left[\Delta\underline{v}_{ji}, \Delta\overline{v}_{ji}\right] = \left[-\eta\frac{\partial E}{\partial \underline{v}_{ji}}, -\eta\frac{\partial E}{\partial \overline{v}_{ji}}\right] \quad j=1,..,n \text{ and } i=1,2,..,n \tag{3.9}$$

Consequently output layer weights $\left(\tilde{w}_{kj}\right)$ between the output layer and the hidden layer are adjusted as

$$\tilde{w}_{kj}^{(New)} = \left[\underline{w}_{kj}^{(New)}, \overline{w}_{kj}^{(New)}\right] = \left[\underline{w}_{kj}^{(Old)}, \overline{w}_{kj}^{(Old)}\right] + \left[\Delta\underline{w}_{kj}, \Delta\overline{w}_{kj}\right] \quad k=1,..,n \quad \text{and} \quad j=1,2,..,n \tag{3.10}$$

where change in weights are calculated as

$$\Delta\tilde{w}_{kj} = \left[\Delta\underline{w}_{kj}, \Delta\overline{w}_{kj}\right] = \left[-\eta\frac{\partial E}{\partial \underline{w}_{kj}}, -\eta\frac{\partial E}{\partial \overline{w}_{kj}}\right] \quad k=1,..,n \text{ and } j=1,2,..,n \tag{3.11}$$

η is the learning constant. In the similar fashion the interval biases $\tilde{\theta}_j$ and $\tilde{\theta}_k$ are also updated.

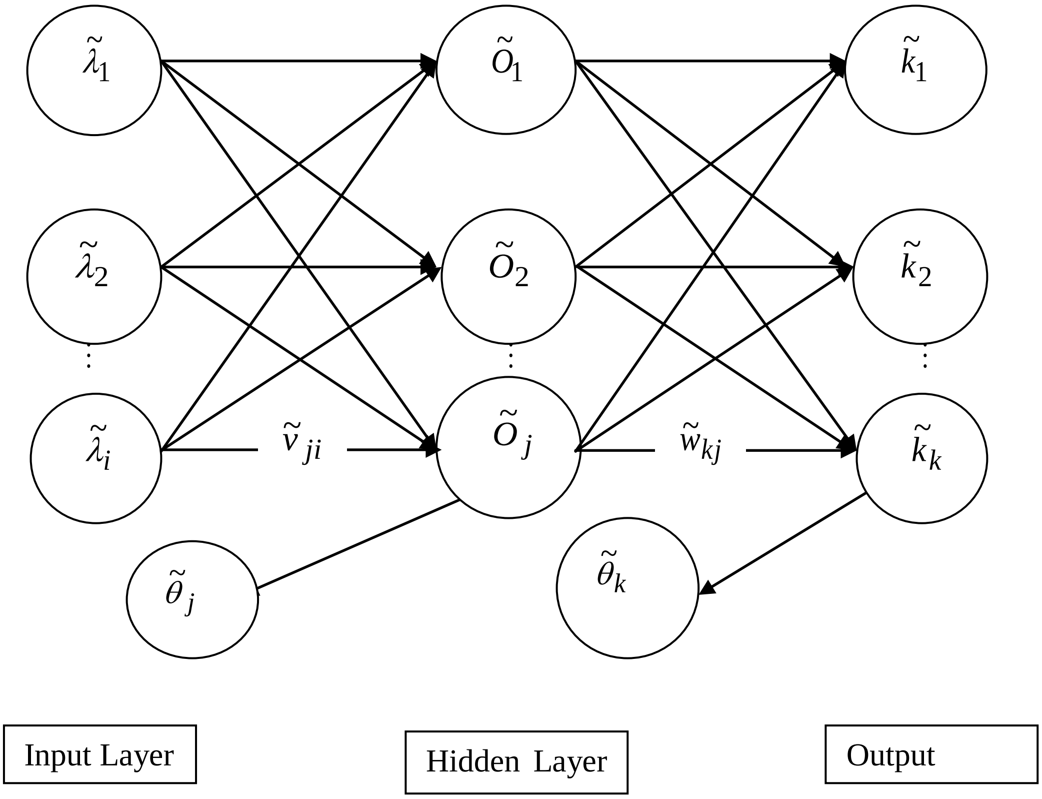

Figure 3.2: Multi-Layer Interval Neural Network Model

3.3 Analysis and Modelling For Fuzzy Case

The floor masses for this methodology are assumed to be $[\underline{m}_1, m_1c, \overline{m}_1]$, $[\underline{m}_2, m_2c, \overline{m}_2]$, ..., $[\underline{m}_n, m_nc, \overline{m}_n]$ and the stiffness $[\underline{k}_1, k_1c, \overline{k}_1]$, $[\underline{k}_2, k_2c, \overline{k}_2]$, ..., $[\underline{k}_n, k_nc, \overline{k}_n]$ are the structural parameters which are to be identified. The fuzzy n-storey shear structure is shown in Figure 3.3. The governing equations of motion for n-storey shear structure in fuzzified form are converted to h-level form and may be written as:

$$\{\hat{M}\}_h \{\hat{\ddot{X}}\}_h + \{\hat{K}\}_h \{\hat{X}\}_h = \{\hat{0}\} \tag{3.12}$$

where , $\{\hat{\ddot{x}}\}_h = \{[\underline{\ddot{x}}]_h , [\overline{\ddot{x}}]_h\}$, $\{\hat{x}\}_h = [[\underline{x}]_h , [\overline{x}]_h]$

$\{\hat{M}\}_h = [[\underline{M}]_h , [\overline{M}]_h]$ is $n \times n$ mass matrix of the structure in h-level form and is given by

$$\left\{\hat{M}\right\}_h = \begin{bmatrix} \left[[\underline{m}_1]_h, [\overline{m}_1]_h\right] & 0 & \cdots & \cdots & 0 \\ 0 & \left[[\underline{m}_2]_h, [\overline{m}_2]_h\right] & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & 0 & \left[[\underline{m}_{n-1}]_h, [\overline{m}_{n-1}]_h\right] & 0 \\ 0 & \cdots & \cdots & 0 & \left[[\underline{m}_n]_h, [\overline{m}_n]_h\right] \end{bmatrix}$$

$\left\{\hat{K}\right\}_h = \left[[\underline{K}]_h, [\overline{K}]_h\right]$ *is* $n \times n$ stiffness matrix of the structure in h-level form and may be written as

$$\left\{\hat{K}\right\}_h = \begin{bmatrix} [\hat{k}_1]_h + [\hat{k}_2]_h & -[\hat{k}_2]_h & 0 & \cdots & 0 \\ -[\hat{k}_2]_h & [\hat{k}_2]_h + [\hat{k}_3]_h & -[\hat{k}_3]_h & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & -[\hat{k}_{n-1}]_h & [\hat{k}_{n-1}]_h + [\hat{k}_n]_h & [\hat{k}_n]_h \\ 0 & \cdots & \cdots & -[\hat{k}_n]_h & [\hat{k}_n]_h \end{bmatrix}$$

and $\left\{\hat{X}\right\}_h = \left\{[\hat{x}_1]_h, [\hat{x}_2]_h, \cdots, [\hat{x}_n]_h\right\}^T$ are the displacement in h-level form .

The above free vibration equation is solved first in order to get the stiffness parameters in h-level form. Accordingly putting $\left\{\hat{X}\right\}_h = \left\{\hat{\phi}\right\}_h e^{i(\hat{\omega})_h t}$ in free vibration equation (3.12), we get

$$\left(\left\{\hat{K}\right\}_h - \left\{\hat{M}\right\}_h [\,\hat{\omega}\,]_h{}^2\right)\left\{\hat{\phi}\right\}_h = \{\,0\,\}_h \tag{3.13}$$

where $\{\hat{\omega}\}_h{}^2 = \left[[\underline{\omega}]_h, [\overline{\omega}]_h\right]^2 = \left[[\underline{\lambda}]_h, [\overline{\lambda}]_h\right]$ are eigen values or the natural frequency and $\{\,\hat{\phi}\,\}_h$ are mode shapes of the structure, respectively.

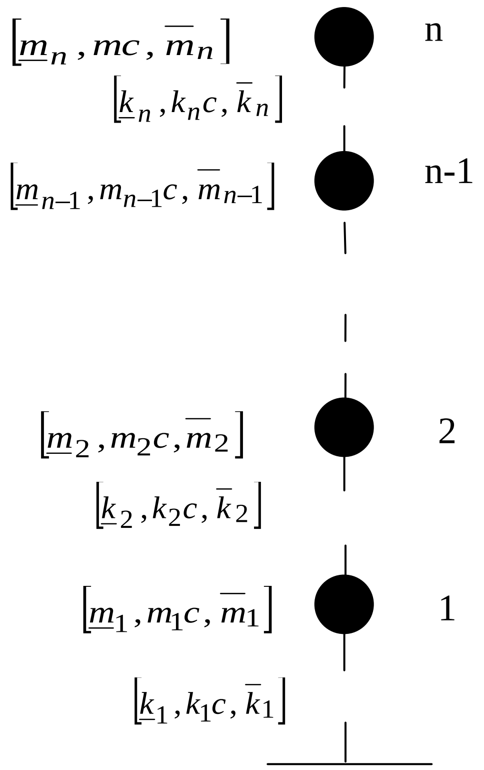
$[\underline{m}_n , mc , \overline{m}_n]$
n
$[\underline{k}_n , k_n c , \overline{k}_n]$
$[\underline{m}_{n-1} , m_{n-1} c , \overline{m}_{n-1}]$
n-1
$[\underline{m}_2 , m_2 c , \overline{m}_2]$
2
$[\underline{k}_2 , k_2 c , \overline{k}_2]$
$[\underline{m}_1 , m_1 c , \overline{m}_1]$
1
$[\underline{k}_1 , k_1 c , \overline{k}_1]$

Figure 3.3: Multi-storey shear structure with n- levels having fuzzy structural parameters

3.4 Fuzzy Neural Network Model

This section describes structure of multi-layer fuzzy neural network model. The training and learning algorithm have already been discussed in Sec. 2.8. Present fuzzy neural network model is given in Figure 3.4. In Figure 3.4, $[\hat{O}_i]_h$, $[\hat{O}_j]_h$ and $[\hat{O}_k]_h$ are taken as input, hidden and output layers. The weights between input and hidden layers are denoted by $[\hat{v}_{ji}]_h = [[\underline{v}_{ji}]_h, [\overline{v}_{ji}]_h]$ and the weights between hidden and output layers are denoted by $[\hat{w}_{kj}]_h = [[\underline{w}_{kj}]_h, [\overline{w}_{kj}]_h]$ which are all in fuzzified form and are taken as *h*-level set. Inputs $[\hat{O}_i]_h = [\hat{\lambda}_i]_h = [[\underline{\lambda}_i]_h, [\overline{\lambda}_i]_h]$ are the frequencies and outputs $[\hat{O}_k]_h = [\hat{k}_k]_h = [[\underline{k}_k]_h, [\overline{k}_k]_h]$ are the stiffness parameters which are in fuzzified form but are converted in h-level form. The total input to the jth hidden unit in the second layer can be calculated as

$$[\hat{U}_j]_h = [[\underline{U}_j]_h, [\overline{U}_j]_h] = [[\underline{v}_{ji}]_h, [\overline{v}_{ji}]_h] \cdot [[\underline{O}_i]_h, [\overline{O}_i]_h] + [[\underline{\theta}_j]_h, [\overline{\theta}_j]_h] \tag{3.14}$$

where $\left\lfloor \left\lfloor \underline{\theta}_j \right\rfloor_h, \left[\overline{\theta}_j\right]_h \right\rfloor$ are the bias weights of the hidden layer. The right hand side of the above equation is computed by interval multiplication and interval addition based on h-level set. The output of the hidden unit is evaluated as

$$\left\lfloor \hat{O}_j \right\rfloor_h = \left\lfloor \left\lfloor \underline{O}_j \right\rfloor_h, \left[\overline{O}_j\right]_h \right\rfloor = \left\lfloor \left\lfloor f\left(\underline{U}_j\right)\right]_h, \left[f\left(\overline{U}_j\right)\right]_h \right\rfloor \tag{3.15}$$

where f is the unipolar activation function defined by $f\left(Ne\hat{t}\right) = \dfrac{1}{1+\exp\left(-\gamma\, Ne\hat{t}\right)}$. The total input from hidden to the output unit in h-level form is calculated as

$$\left\lfloor \hat{Y}_k \right\rfloor_h = \left[\left[\underline{Y}_k\right]_h, \left[\overline{Y}_k\right]_h\right] = \left\lfloor \left\lfloor \underline{w}_{kj} \right\rfloor_h, \left[\overline{w}_{kj}\right]_h \right\rfloor \cdot \left\lfloor \left\lfloor \underline{O}_j \right\rfloor_h, \left[\overline{O}_j\right]_h \right\rfloor + \left[\left[\underline{\theta}_k\right]_h, \left[\overline{\theta}_k\right]_h\right] \tag{3.16}$$

Again the right hand side of the above equation involves interval multiplication and interval addition based on h-level set, where $\left\lfloor \left[\underline{\theta}_k\right]_h, \left[\overline{\theta}_k\right]_h \right\rfloor$ are the bias weights of the output layer. Finally, the response of the net is given as

$$\left\lfloor \hat{O}_k \right\rfloor_h = \left[\left[\underline{O}_k\right]_h, \left[\overline{O}_k\right]_h\right] = \left[\left[f\left(\underline{Y}_k\right)\right]_h, \left[f\left(\overline{Y}_k\right)\right]_h\right] \tag{3.17}$$

The error value is computed as

$$E = \sum_{k=1}^{n} \left\{ \frac{h}{2}\left(\left[\underline{d}_k\right]_h - \left[\underline{O}_k\right]_h\right)^2 + \frac{h}{2}\left(\left[\overline{d}_k\right]_h - \left[\overline{O}_k\right]_h\right)^2 \right\} \tag{3.18}$$

From the cost function (3.18) a learning rule can be derived for the weights $\left(\hat{v}_{ji}\right)_h$ between the hidden and input layer. The h-level weights are updated as

$$\left[\hat{v}_{ji}^{(New)}\right]_h = \left[\left[\underline{v}_{ji}^{(New)}\right]_h, \left[\overline{v}_{ji}^{(New)}\right]_h\right] = \left[\left[\underline{v}_{ji}^{(Old)}\right]_h, \left[\overline{v}_{ji}^{(Old)}\right]_h\right] + \left[\left[\Delta\underline{v}_{ji}\right]_h, \left[\Delta\overline{v}_{ji}\right]_h\right] \tag{3.19}$$

j=1,..,n and *i=1,2,..,n*

where change in weights are calculated as

$$\left[\Delta\hat{v}_{ji}\right]_h = \left[\left[\Delta\underline{v}_{ji}\right]_h, \left[\Delta\overline{v}_{ji}\right]_h\right] = \left[-\eta \frac{\partial E}{\partial\left[\underline{v}_{ji}\right]_h}, -\eta \frac{\partial E}{\partial\left[\overline{v}_{ji}\right]_h} \right] \qquad j=1,..,n \qquad \text{and} \tag{3.20}$$

i=1,2,..,n

Consequently output layer weights $\left(\hat{w}_{kj}\right)_h$ between the output layer and the hidden layer are adjusted as

$$\left[\hat{w}_{kj}^{(New)}\right]_h = \left[\left[\underline{w}_{kj}^{(New)}\right]_h, \left[\overline{w}_{kj}^{(New)}\right]_h\right] = \left[\left[\underline{w}_{kj}^{(Old)}\right]_h, \left[\overline{w}_{kj}^{(Old)}\right]_h\right] + \left[\left[\Delta \underline{w}_{kj}\right]_h, \left[\Delta \overline{w}_{kj}\right]_h\right] \quad (3.21)$$

k=1,..,n and *j=1,2,..,n*

where change in weights are calculated as

$$\left[\Delta \hat{w}_{kj}\right]_h = \left[\left[\Delta \underline{w}_{kj}\right]_h, \left[\Delta \overline{w}_{kj}\right]_h\right] = \left[-\eta \frac{\partial E}{\partial \left[\underline{w}_{kj}\right]_h}, -\eta \frac{\partial E}{\partial \left[\overline{w}_{kj}\right]_h} \right] \quad (3.22)$$

k=1,..,n and *j=1,2,..,n*

η is the learning constant. In the similar fashion the fuzzy biases $\left[\hat{\theta}_j\right]_h$ and $\left[\hat{\theta}_k\right]_h$ are also updated.

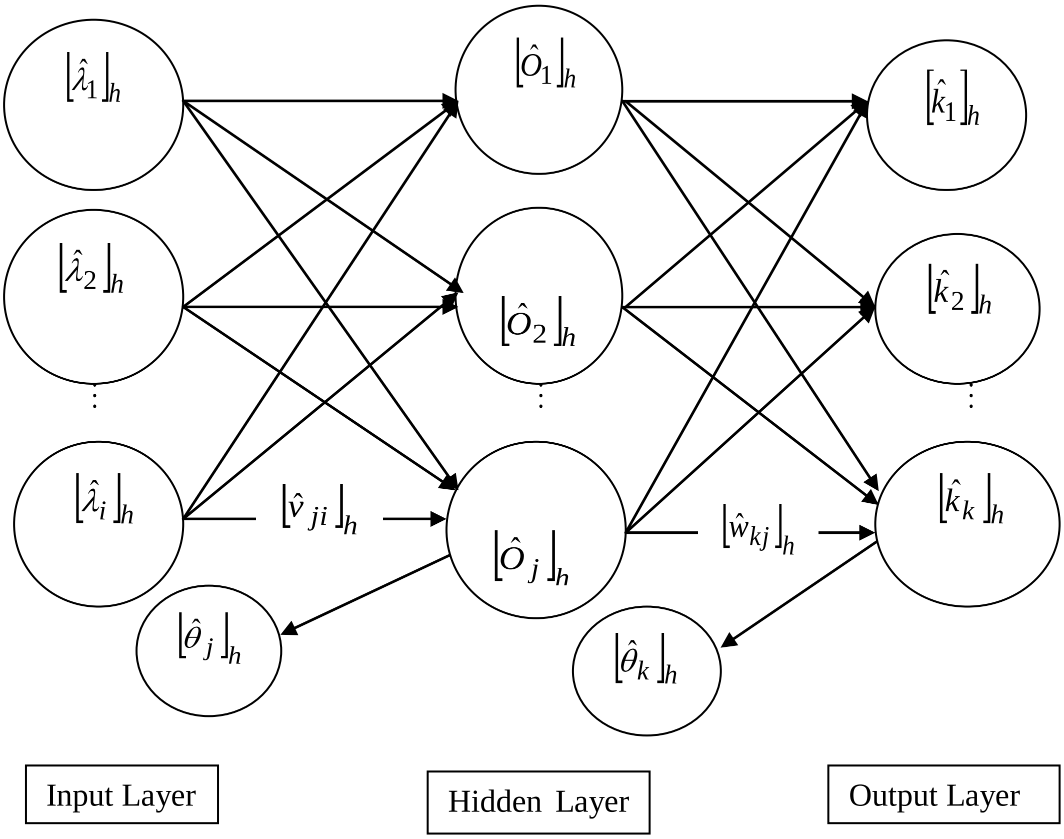
$\left[\hat{\lambda}_1\right]_h$
$\left[\hat{\lambda}_2\right]_h$
$\left[\hat{\lambda}_i\right]_h$
$\left[\hat{O}_1\right]_h$
$\left[\hat{O}_2\right]_h$
$\left[\hat{O}_j\right]_h$
$\left[\hat{k}_1\right]_h$
$\left[\hat{k}_2\right]_h$
$\left[\hat{k}_k\right]_h$
$\left[\hat{v}_{ji}\right]_h$
$\left[\hat{w}_{kj}\right]_h$
$\left[\hat{\theta}_j\right]_h$
$\left[\hat{\theta}_k\right]_h$
Input Layer
Hidden Layer
Output Layer

Figure 3.4: Multi-Layer Fuzzy Neural Network Model

3.5 Case Studies

3.5.1 Crisp Case

Examples of two, five and ten storey shear structure have been considered. It may be noted that in further examples in interval and fuzzy cases, different multistorey shear structures are considered to simulate and validate the methodologies. The inputs are taken as the crisp frequency values and the outputs are the stiffness parameters which are also in crisp form.

Example 1. Two storey shear buildings:

An example of a two storey shear structure is considered where the masses are $m_1 = m_2 = 36000$and the initial crisp stiffness parameters have their values as $k_1 = 100000$ and $k_2 = 20000$. From these initial values of crisp stiffness parameter we have generated 40 sets of data for both stiffness and frequency. The 40 training pattern is used in this case for training. Here the input layer contains 2 input neurons and output layer contain 2 output neurons. Various numbers of hidden nodes are considered and the program was executed. After few runs it was seen that 5 hidden nodes are sufficient to get the desirable result. As such with accuracy of 0.001, the desired and ANN results for ten data among them are summarized in Table 3.1.

Table 3.1: Comparison between desired and ANN value of k_1 and k_2 (crisp) for two storey shear structure

Data No.	k_1 (ANN)	k_1 (Des)	k_2 (ANN)	k_2 (Des)
1	183141.0632	181472.3686	21148.1455	21576.1308
2	190868.2441	190579.1937	28879.8781	29705.9278
3	112821.4965	112698.6816	28837.8503	29571.6695
4	190721.7835	191337.5856	25562.316	24853.7565
5	164741.1345	163235.9246	28164.8835	28002.8047
6	110890.1358	109754.0405	21234.1519	21418.8634
7	125986.1574	127849.8219	24837.6067	24217.6128
8	154938.2043	154688.1519	28732.9869	29157.3553
9	192911.6695	195750.6835	28071.4429	27922.0733
10	193284.5597	196488.8535	28828.9278	29594.9243

Example 2. Five storey shear buildings:

The masses for five storey shear structure is taken as $m_1 = ... = m_5 = 36000$ and the stiffness parameters are within the range $k_1 = [100000\ 200000]$, $k_2 = [50000\ 100000]$, $k_3 = [40000\ 60000]$, $k_4 = [30000\ 50000]$, $k_5 = [20000\ 30000]$. A comparison between the desired and ANN values have been presented in Table 3.2. This table has been plotted in Figure 3.5.

Table 3.2: Comparison between desired and ANN values of k_1, k_2, k_3, k_4, k_5 for a five storey shear structure

Data No.	k_1(ANN)	k_1(Des)	k_2(ANN)	k_2(Des)	k_3(ANN)	k_3(Des)	k_4(ANN)	k_4(Des)	k_5(ANN)	k_5(Des)
1	181277.738	181472.368	57722.680	57880.654	53111.561	53114.814	44531.813	44120.921	24304.530	24387.443
2	195883.722	190579.193	98497.336	98529.639	40890.778	40714.233	30510.588	30636.656	24013.411	23815.584
3	110889.324	112698.681	97540.287	97858.347	56939.133	56982.586	35467.070	35538.459	27905.072	27655.167
4	191334.636	191337.585	74367.090	74268.782	58151.367	58679.865	30921.744	30923.427	27758.983	27951.999
5	162671.148	163235.924	90080.929	90014.023	53507.075	53574.703	31851.263	31942.635	21775.050	21868.726
6	109757.877	109754.040	57034.339	57094.316	55073.904	55154.802	46360.940	46469.156	24899.561	24897.644
7	127671.411	127849.821	71003.128	71088.064	54999.758	54862.649	43688.961	43896.572	24554.259	24455.862
8	154429.361	154688.151	98500.007	95786.776	48031.314	47844.540	36250.004	36341.989	27719.138	26463.130
9	196399.16	195750.683	89751.902	89610.366	53100.037	53109.557	48498.010	49004.441	27950.536	27093.648
10	193722.466	196488.853	98520.881	97974.621	43228.929	43423.733	30822.026	30688.921	26813.963	27546.866

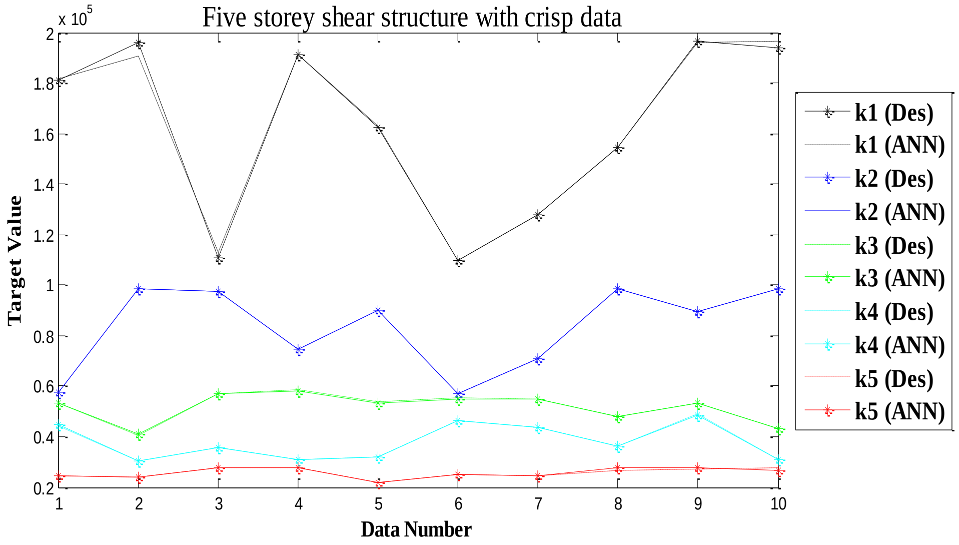

Figure 3.5: Comparison between desired and ANN values of K for a five storey shear structure

Example 3. Ten storey shear buildings:

For ten storey shear structure constant masses have been considered and stiffness parameters are in the range $k_1 = [100000\ 200000]$, $k_2 = ... = k_{10} = [20000\ 30000]$. The desired and ANN values for k_1 to k_5 and k_6 to k_{10} compared in Tables 3.3(a) and 3.3(b) respectively.

Table 3.3(a): Comparison between desired and ANN values of k_1, k_2, k_3, k_4, k_5 for a ten storey shear structure.

Data No.	k_1(ANN)	k_1(Des)	k_2(ANN)	k_2(Des)	k_3(ANN)	k_3(Des)	k_4(ANN)	k_4(Des)	k_5(ANN)	k_5(Des)
1	116900.656	114999.725	23626.293	23947.074	25255.905	24299.214	29078.698	29493.039	27766.141	28842.810
2	137413.465	135922.821	22400.131	21970.538	22076.765	22160.189	28671.607	29898.721	23928.679	23185.242
3	174166.557	171165.670	26751.824	27587.662	28439.483	28089.902	28056.190	27636.733	28909.507	29349.790
4	185745.634	187147.651	28203.597	29952.159	23288.599	23565.089	25414.296	25588.205	24175.838	24794.845
5	131376.808	132868.961	22531.439	21865.714	20887.901	20732.434	22490.503	21838.429	21747.449	22317.916
6	174687.251	165011.802	25813.586	27811.452	25015.101	25909.914	25615.732	24979.488	24612.157	23962.902
7	185115.365	197483.614	24841.291	21957.9798	27352.970	29101.878	25184.177	25178.456	27414.622	27050.774
8	107439.578	107596.736	28408.161	29923.589	22079.074	21937.659	29361.573	29942.430	25283.601	25585.590
9	164872.295	158701.916	25498.986	28022.615	26269.187	24323.677	28505.576	28548.516	27421.786	27566.307
10	134496.319	136428.686	23984.669	23091.364	26528.542	27288.638	20735.487	20391.844	26590.029	26789.410

Table 3.3(b): Comparison between desired and ANN values of $k_6, k_7, k_8, k_9, k_{10}$ for a ten storey shear structure

Data No.	k_6(ANN)	k_6(Des)	k_7(ANN)	k_7(Des)	k_7(ANN)	k_8(Des)	k_9(ANN)	k_9(Des)	k_{10}(ANN)	k_{10}(Des)
1	21170.097	20899.506	25777.665	25605.595	28204.348	29899.502	27503.417	25859.870	27163.546	25814.464
2	22417.136	20549.741	27621.649	28654.385	27612.505	28451.781	28303.999	29823.032	22093.350	22094.050
3	29672.968	29638.701	25998.821	27124.148	22438.230	21982.217	26315.554	26153.251	27684.854	29019.908
4	19962.258	19656.563	21156.258	20166.747	22269.519	21950.715	23818.782	23766.110	27240.574	27020.664
5	20161.80	20514.482	25387.609	28009.208	23922.463	23268.396	27519.842	28771.817	23219.641	23774.551
6	22307.093	23043.489	21527.116	21425.093	27311.644	28803.378	27670.745	27848.524	27234.798	27349.559
7	27804.445	25801.918	24442.816	24784.744	25319.165	24711.018	25870.655	24649.542	27640.802	29541.027
8	25509.234	25309.644	22738.136	22568.353	23936.881	24039.693	28924.111	28139.769	25172.674	25428.131
9	28341.851	29012.080	25380.192	23690.916	23071.182	21792.314	28034.575	28984.441	26519.439	25401.058
10	27911.291	29624.314	26475.855	24319.806	21306.363	21696.088	25327.536	24074.557	24485.256	23343.294

3.5.2 Interval Case

For interval case five and ten storey have been considered.

Example 1. Five storey shear buildings:

Two problems have been solved for five storey shear structure. The masses are kept constant for each story for both the problems. Hence we assume that the structure have the masses as $\tilde{m}_1 = \tilde{m}_2 = \tilde{m}_3 = \tilde{m}_4 = \tilde{m}_5 = 36000$. The initial interval stiffness parameters used to train the first problem have values as $\tilde{k}_1 = [\underline{k}_1, \bar{k}_1] = [100000, 200000]$ and $\tilde{k}_2 = [\underline{k}_2, \bar{k}_2] = \ldots = \tilde{k}_5 = [\underline{k}_5, \bar{k}_5] = [20000, 30000]$. For the second problem, the initial interval stiffness are taken as $\tilde{k}_1 = \left[\underline{k}_1, \overline{k}_1\right] = \left[100000, 200000\right]$, $\tilde{k}_2 = \left[\underline{k}_2, \overline{k}_2\right] = \left[50000, 100000\right]$ and $\tilde{k}_3 = \left[\underline{k}_3, \overline{k}_3\right] = \ldots = \tilde{k}_5 = \left[\underline{k}_5, \overline{k}_5\right] = \left[20000, 80000\right]$. In this case the input layer contains 5 input neurons and the output layer also contains 5 output neurons. Again as in crisp case, we generate 60 sets of interval data of stiffness parameters and frequency parameters. Again various numbers of hidden nodes are taken to optimize the number of hidden nodes as per the required accuracy. Finally 10 hidden nodes are found to be sufficient so as to get an accuracy of 0.001. After training, 10 trained data among 60 are incorporated for comparison of the desired and INN values for first problem in Tables 3.4(a) and 3.4(b). Due to insufficient space, the data for only first and second storey of five storey shear structure have been incorporated in Table 3.4(a). Similarly the data for fourth and fifth storey have been presented in Table 3.4(b). For second problem comparison between the desired and INN values for 10 data among 60 data have been plotted in Figures 3.6(a) to 3.6(e).

Table 3.4(a): Comparison between desired and INN value of $\underline{k}_1$, $\overline{k}_1$ and $\underline{k}_2$, $\overline{k}_2$ for a five storey shear structure

Data No.	$\underline{k}_1$ (INN)	$\underline{k}_1$ (Des)	$\overline{k}_1$ (INN)	$\overline{k}_1$ (Des)	$\underline{k}_2$ (INN)	$\underline{k}_2$ (Des)	$\overline{k}_2$ (INN)	$\overline{k}_2$ (Des)
1	128030	127850	141610	144560	20470	20840	27800	27570
2	158930	158530	178080	179220	21740	20120	25780	25500
3	167610	167870	185880	189090	20860	21660	22230	24020
4	169450	170600	185680	184070	24240	24170	24860	24510
5	103750	103180	125250	125430	21370	20500	20980	20840
6	103990	104620	122070	124350	27190	29130	29100	29450
7	110810	109710	186720	192930	21400	21520	25600	24910
8	132800	135000	188070	182350	23650	24890	28220	28260
9	164650	165510	192840	195720	22120	21820	25010	25310
10	121700	117120	127180	125750	22970	22400	27340	27480

Table 3.4(b): Comparison between desired and INN value of $\underline{k}_4$, $\overline{k}_4$ and $\underline{k}_5$, $\overline{k}_5$ for a five storey shear structure

Data No.	$\underline{k}_4$ (INN)	$\underline{k}_4$ (Des)	$\overline{k}_4$ (INN)	$\overline{k}_4$ (Des)	$\underline{k}_5$ (INN)	$\underline{k}_5$ (Des)	$\overline{k}_5$ (INN)	$\overline{k}_5$ (Des)
1	25220	24890	26830	26180	24530	24610	25680	25270
2	24290	23670	25160	25000	22870	23850	24930	24900
3	20890	20990	25990	25220	23490	22690	28500	28240
4	20760	21070	27090	26600	23030	23010	29100	29060
5	25320	25190	26340	26540	27140	27010	28670	28800
6	26210	26490	28220	27790	22490	22610	25240	25390
7	26460	27150	29180	28000	26720	25940	27090	26980
8	25100	24540	29000	29040	20450	20230	25580	26670
9	25100	25210	28500	28870	20400	20150	23330	21910
10	21690	21500	27510	27210	20710	21080	28840	29830

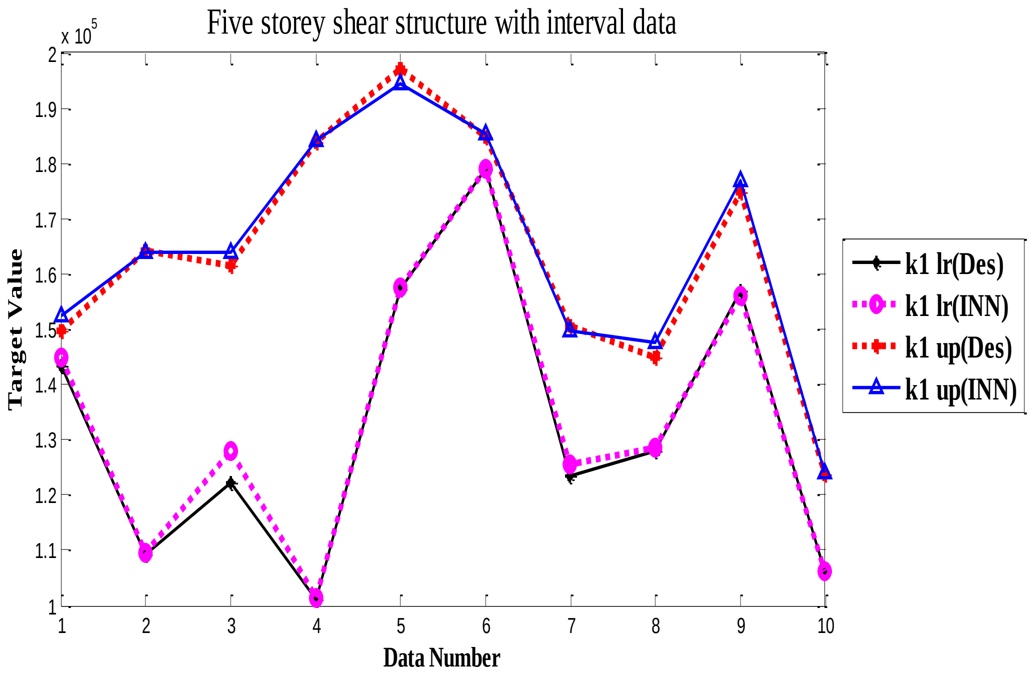

Figure 3.6(a): Comparison between desired and INN value of $\underline{k}_1$ and $\overline{k}_1$ for a five storey shear structure

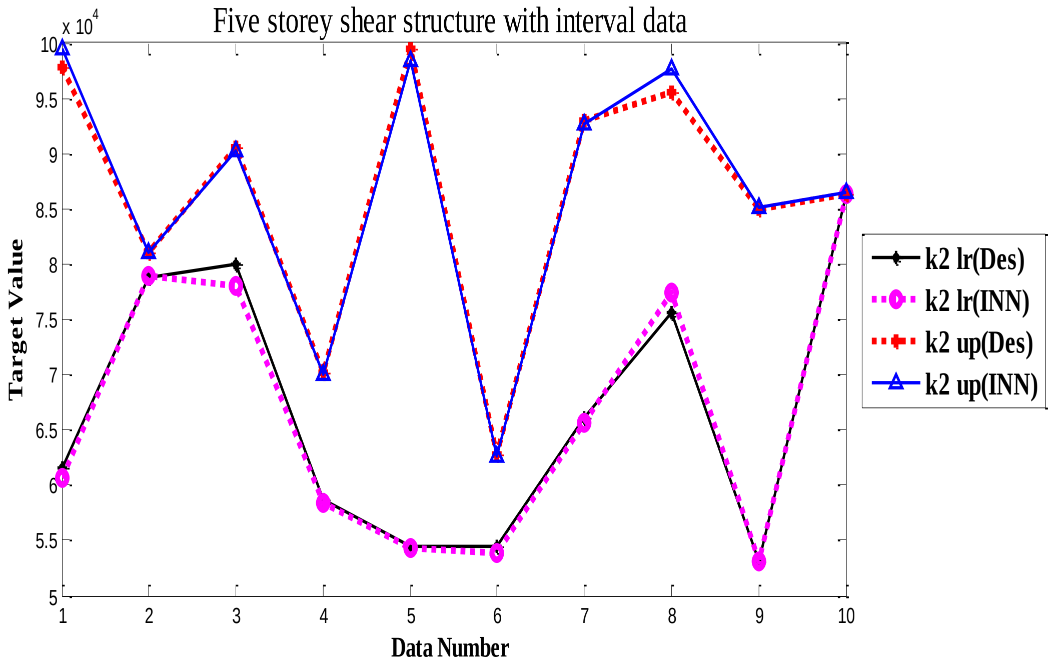

Figure 3.6(b): Comparison between desired and INN value of $\underline{k}_2$ and $\overline{k}_2$ for a five storey shear structure

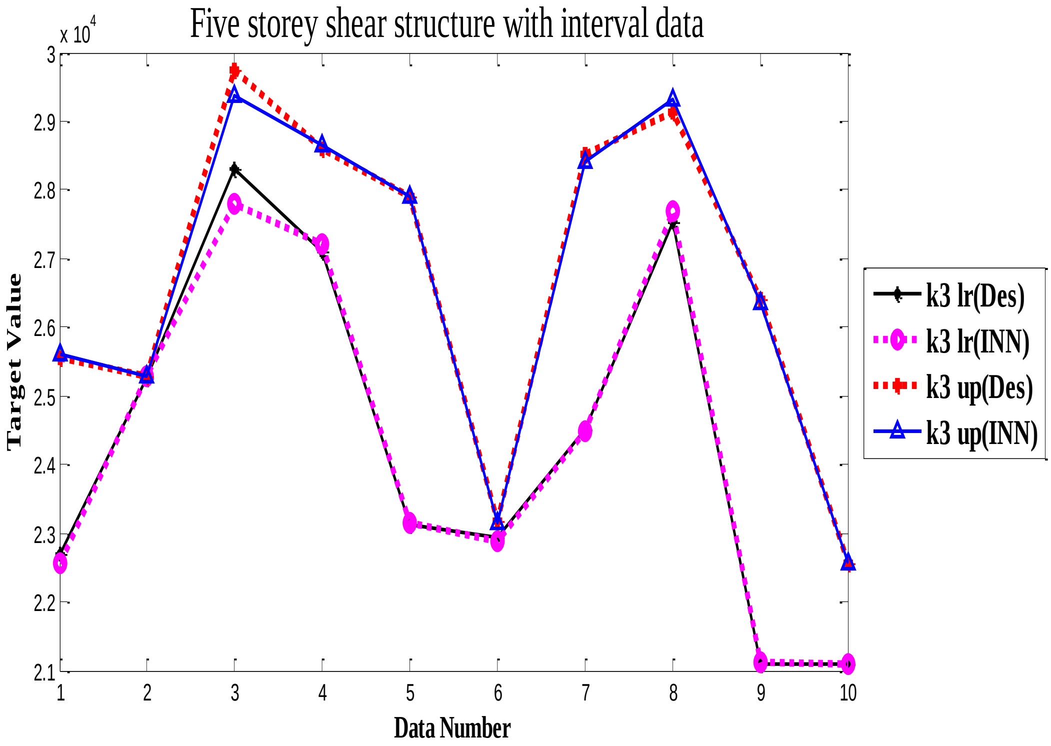

Figure 3.6(c): Comparison between desired and INN value of $\underline{k}_3$ and $\overline{k}_3$ for a five storey shear structure

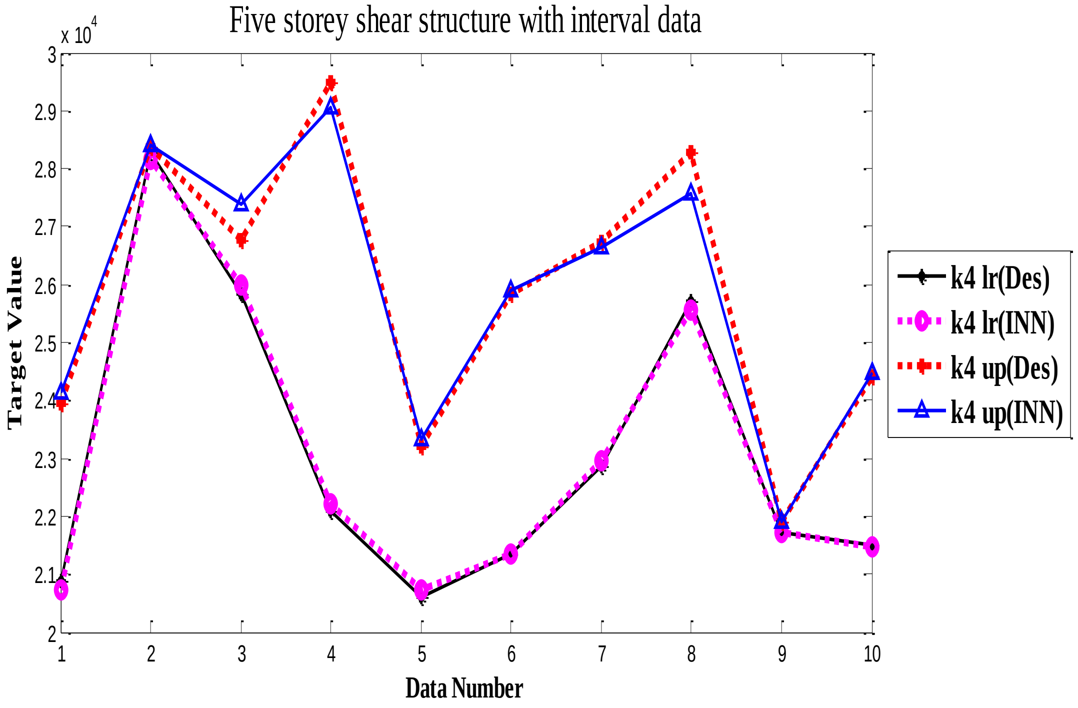

Figure 3.6(d): Comparison between desired and INN value of $\underline{k}_4$ and $\overline{k}_4$ for a five storey shear structure

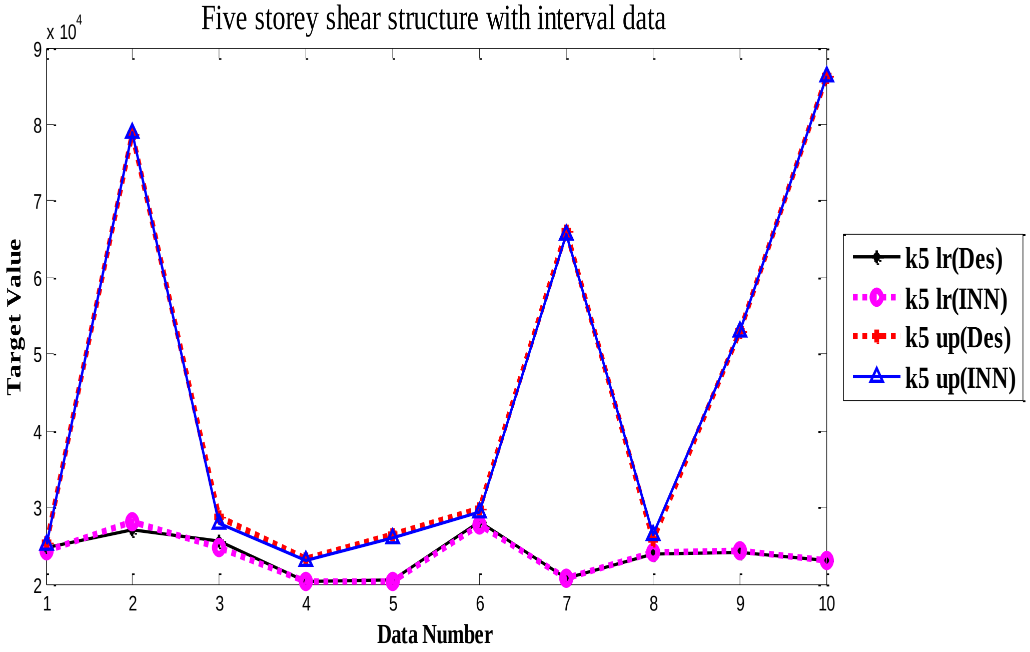

Figure 3.6(e): Comparison between desired and INN value of $\underline{k}_5$ and $\overline{k}_5$ for a five storey shear structure

Example 2. Ten storey shear buildings:

Finally, ten storey shear structure have been considered, where the masses for each stories are taken to be constant as $\tilde{m}_1 = \tilde{m}_2 = \ldots = \tilde{m}_{10} = 36000$and the initial interval stiffness parameter are considered to be as $\tilde{k}_1 = [\underline{k}_1, \overline{k}_1] = [100000, 200000]$ and $\tilde{k}_2 = [\underline{k}_2, \overline{k}_2] = \ldots = \tilde{k}_{10} = [\underline{k}_{10}, \overline{k}_{10}] = [20000, 30000]$. In this case, the input as well as output layer contains 10 neurons each and 100 data sets are trained with 15 hidden nodes in a single hidden layer so as to get an accuracy of 0.001. Comparison between the desired and INN values for 10 data among 100 sets of data have been incorporated in Tables 3.5(a) and 3.5(b). Table 3.5(a) gives the data for fifth and seventh storey of a ten storey shear structure. The data for ninth and tenth storey of the same structure have been tabulated in Table 3.5(b).

Table 3.5(a): Comparison between desired and INN value of $\underline{k}_5$, $\overline{k}_5$ and $\underline{k}_7$, $\overline{k}_7$for a ten storey shear structure

Data No.	$\underline{k}_5$ (INN)	$\underline{k}_5$ (Des)	$\overline{k}_5$ (INN)	$\overline{k}_5$ (Des)	$\underline{k}_7$ (INN)	$\underline{k}_7$ (Des)	$\overline{k}_7$ (INN)	$\overline{k}_7$ (Des)
1	22680	24390	25110	26320	24070	24030	24930	24700
2	20140	20780	22940	23110	22830	22800	23640	23700
3	19000	20350	24240	24840	25380	25910	27490	27660
4	26730	27250	29050	29760	22000	21320	28310	28680
5	21710	23510	28320	29870	23330	23100	26620	26990
6	22240	22110	25470	25200	25290	24770	28380	27360
7	20900	21530	25430	25620	21590	21400	28600	28960
8	25320	28050	27500	28660	23230	22910	27740	27860
9	24260	22990	28500	26490	23680	23650	24010	24360
10	23000	22300	24770	24550	22100	21890	24490	24550

Table 3.5(b): Comparison between desired and INN value of $\underline{k}_9$, $\overline{k}_9$ and $\underline{k}_{10}$, $\overline{k}_{10}$ for a ten storey shear structure

Data No.	$\underline{k}_9$ (INN)	$\underline{k}_9$ (Des)	$\overline{k}_9$ (INN)	$\overline{k}_9$ (Des)	$\underline{k}_{10}$ (INN)	$\underline{k}_{10}$ (Des)	$\overline{k}_{10}$ (INN)	$\overline{k}_{10}$ (Des)
1	25410	24470	29020	26860	26250	25540	28370	27520
2	22080	21710	26400	26520	23050	23130	27470	27680
3	24340	24170	25150	24950	22070	22240	24910	24810
4	22620	22450	24790	23930	22360	22520	29880	29820
5	22990	21830	23760	24010	20690	20170	22440	21660
6	21160	20490	28300	29090	21950	21930	29880	26930
7	22810	22390	24760	24390	27190	27790	29880	29390
8	25890	25170	25530	25830	23650	24030	29880	28720
9	24810	25050	28630	28560	19610	20000	22600	22140
10	26980	27470	28020	28120	27030	27140	29500	29880

3.5.3 Testing for Interval Case

An example of a five storey shear structure for comparing the testing values (which are not used in the training) of desired and ANN values in interval form is considered. These test values are fed into the neural network along with the stored converged weights to generate corresponding stiffness parameters. The stiffness parameters used for testing have the lower and upper value as $\tilde{k}_1 = \left[\underline{k}_1, \overline{k}_1\right] = \left[100000, 200000\right]$ and $\tilde{k}_2 = \left[\underline{k}_2, \overline{k}_2\right] = \ldots = \tilde{k}_5 = \left[\underline{k}_5, \overline{k}_5\right] = \left[20000, 30000\right]$. Comparison between the test values of desired and INN for 10 data are incorporated in Tables 3.6(a) and 3.6(b). In this case the data for first and second storey of a five storey shear structure are included in Table 3.6(a) and the data for fourth and fifth storey of the same structure have been tabulated in Table 3.6(b).

Table 3.6(a): Comparison between the testing values of desired and INN of $\underline{k}_1, \overline{k}_1$ and $\underline{k}_2$, $\overline{k}_2$ for five storey shear structure

Data No.	$\underline{k}_1$ (INN)	$\underline{k}_1$ (Des)	$\overline{k}_1$ (INN)	$\overline{k}_1$ (Des)	$\underline{k}_2$ (INN)	$\underline{k}_2$ (Des)	$\overline{k}_2$ (INN)	$\overline{k}_2$ (Des)
1	102110	101540	140770	140390	21660	21890	24540	24360
2	108890	104300	114840	109650	22780	24470	25110	26870
3	109720	113200	112450	116900	22940	21840	24600	23060
4	168020	164910	190680	194210	23990	23680	26190	25090
5	171590	173170	192360	195610	26030	25110	26470	26260
6	163760	157520	167350	164770	26620	27800	28060	28180
7	110100	105980	147190	145090	21810	20810	28030	27950
8	121690	123480	152580	154700	24450	26440	28210	29290
9	128940	129630	136210	135320	24880	23790	28700	27760
10	167020	174470	182930	182120	23950	24870	27760	28120

Table 3.6(b): Comparison between the testing values of desired and INN of $\underline{k}_4, \overline{k}_4$ and $\underline{k}_5$ $\overline{k}_5$ for five storey shear structure

Data No.	$\underline{k}_4$ (INN)	$\underline{k}_4$ (Des)	$\overline{k}_4$ (INN)	$\overline{k}_4$ (Des)	$\underline{k}_5$ (INN)	$\underline{k}_5$ (Des)	$\overline{k}_5$ (INN)	$\overline{k}_5$ (Des)
1	22310	21850	27570	27110	20560	20290	22470	22320
2	21650	22220	28300	29050	26240	24890	28650	29290
3	21620	21170	28390	29800	26240	26240	28150	27300
4	23480	22970	25190	24390	24700	24890	28040	26790
5	21670	21110	23170	23190	24380	23960	26860	25790
6	23080	22580	24220	24240	22860	22370	25060	23670
7	23690	24090	26090	25080	24320	24590	29260	29880
8	21450	20860	24220	25950	21080	20380	28800	29630
9	23480	22620	24740	22620	25720	25470	29510	28850
10	25340	26030	26310	28010	25420	25210	28630	29130

3.5.4 Fuzzy Case

In this case one and two storey shear structures have been taken. Two examples for each case have been solved.

Example 1. One storey shear buildings:

First example for one storey shear structure with masses $\hat{M} = 36000$ and the stiffness parameters lies within the range $\underline{K}$ = $[100000 \quad 200000]$, Kc = $[100010 \quad 200010]$, $\overline{K}$ = $[100020 \quad 200020]$ is solved. A comparison between desired and the ANN values have been incorporated in Table 3.7. This table has been plotted in Figure 3.7. In the second example masses $\hat{M} = 36000$ and the stiffness parameter varying within the range $\underline{K} =$ $[50000 \quad 100000]$, $Kc =$ $[50010 \quad 100010]$ $\overline{K} =$ $[50020 \quad 100020]$ are considered. Comparison between the desired and the FNN values are tabulated in Table 3.8 and are plotted in Figure 3.8.

Table 3.7: Comparison between desired and FNN values of $\hat{K}$ for a single storey shear structure

Data No.	$\underline{K}$ (FNN)	$\underline{K}$ (Des)	Kc (FNN)	Kc (Des)	$\overline{K}$ (FNN)	$\overline{K}$ (Des)
1	124932.9319	124189.1286	135932.8213	135095.2381	191905.7285	190281.611
2	135763.4956	140411.2146	152464.2896	151324.954	191726.9669	194488.719
3	110291.5736	109665.4525	141591.1084	140180.8034	148514.9196	149096.4092
4	107394.0584	107596.6692	115134.7117	113217.3293	149119.797	148935.2638
5	124430.5911	123991.6154	134455.7107	133781.941	192947.0787	194225.0591
6	112574.5401	112331.8935	188853.0283	190015.3846	194542.4439	195633.454
7	120988.7913	118390.7788	138166.8698	136934.6781	157619.3944	157540.8595
8	108960.7756	105997.9543	113526.3083	111130.2755	126504.2017	123995.2526
9	124700.9011	123497.9913	143556.5197	141726.7069	179985.5092	178035.2068
10	105227.009	104965.443	135988.057	135335.8571	140612.991	138983.8837

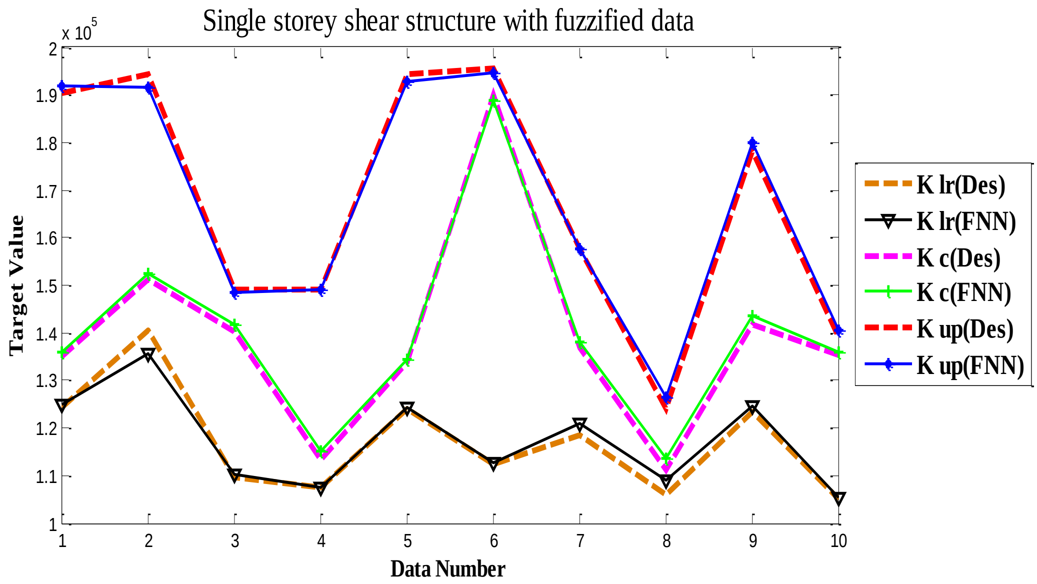

Figure 3.7: Comparison between desired and FNN values of $\hat{K}$ for a single storey shear structure

Table 3.8: Comparison between desired and FNN values of $\hat{K}$ for a single storey shear structure

Data No.	$\underline{k}\ (FNN)$	$\underline{k}\ (Des)$	$kc\ (FNN)$	$kc\ (Des)$	$\overline{k}\ (FNN)$	$\overline{k}\ (Des)$
1	62365.0341	62104.5643	67644.4378	67547.619	95458.8463	95145.8055
2	67613.8959	70215.6073	75826.931	75662.477	95893.1612	97249.3595
3	55005.0087	54842.7263	70556.3936	70090.4017	74368.7997	74553.2046
4	53128.7186	53798.3346	56947.542	56618.6646	74261.1768	74472.6319
5	62246.1626	61995.8077	66881.2194	66895.9705	96278.4523	97122.5295
6	55957.2602	56165.9467	94391.1219	95012.6923	97271.6645	97826.727
7	60056.7259	59195.3894	68223.7328	68472.3391	78385.7211	78780.4298
8	53587.3374	53008.9771	55747.5298	55570.1378	61968.3882	61997.6263
9	62126.6388	61758.9957	71196.546	70863.3535	89991.9346	89022.6034
10	52238.7913	52482.7215	67702.5259	67677.9286	69679.1387	69496.9418

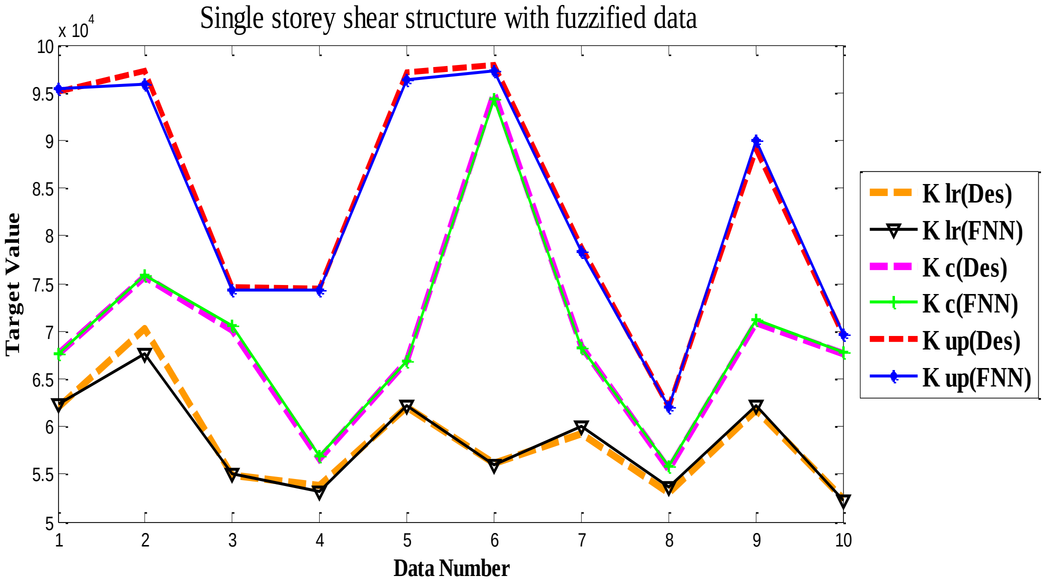

Figure 3.8: Comparison between desired and FNN values of $\hat{K}$ for a single storey shear structure

Example 2. Two storey shear buildings:

The first example for two storey shear structure is considered where the masses are $\hat{m}_1 = \hat{m}_2 = 36000$ and the stiffness parameters varying within the range $\underline{k}_1 = [100000 \quad 200000]$, $k_1c = [100010 \quad 200010]$, $\overline{k}_1 = [100020 \quad 200020]$ and $\underline{k}_2 = [20000 \quad 30000]$, $k_2c = [20010 \quad 30010]$, $\overline{k}_2 = [20020 \quad 30020]$. The desired and FNN values have been compared in Tables 3.9(a) and 3.9(b). These tables have also been shown in Figures 3.9(a) and 3.9(b). In the second example a two storey shear structure is implemented with masses $\hat{m}_1 = \hat{m}_2 = 36000$ and the stiffness parameters having the range $\underline{k}_1 = [50000 \quad 100000], k_1c = [50010 \quad 100010], \overline{k}_1 = [50020 \quad 100020]$ and $\underline{k}_2 = [20000 \quad 30000]$, $k_2c = [20010 \quad 30010]$, $\overline{k}_2 = [20020 \quad 30020]$. Comparison between the desired and FNN values are again incorporated in Tables 3.10(a) and 3.10(b). These tables are plotted in Figures 3.10(a) and 3.10(b).

Table 3.9(a): Comparison between desired and FNN values of $\hat{k}_1$ for a double storey shear structure

Data No.	$\underline{k_1}\ (FNN)$	$\underline{k_1}\ (Des)$	$k_1c\ (FNN)$	$k_1c\ (Des)$	$\overline{k_1}\ (FNN)$	$\overline{k_1}\ (Des)$
1	113258.5422	113317.1008	133858.2555	133969.3413	162741.2326	162807.3359
2	116526.66	117338.8613	128622.4646	129208.408	193447.2534	195183.0465
3	139119.0131	139093.7802	143501.0894	143175.117	191801.5675	192053.204
4	101316.7594	101558.7126	106512.8803	105287.6998	183040.5812	183137.9743
5	173554.7972	173805.8096	180269.037	180336.4392	198199.4731	198416.3724
6	105985.8857	106047.1179	117592.2043	116726.841	127688.8972	126931.9426
7	110472.1362	110631.6345	140132.9188	139925.7771	142461.6007	142303.5615
8	137120.3743	137250.974	152856.9037	152687.5831	155005.6994	154807.0901
9	119783.849	119821.8403	142480.6438	141679.9468	195824.6624	194293.6984
10	141708.5217	141794.4104	148654.6797	148978.7638	165370.4902	165685.9891

Table 3.9(b): Comparison between desired and FNN values of $\hat{k}_2$ for a double storey shear structure

Data No.	$\underline{k_2}\ (FNN)$	$\underline{k_2}\ (Des)$	$k_2c\ (FNN)$	$k_2c\ (Des)$	$\overline{k}_2\ (FNN)$	$\overline{k}_2\ (Des)$
1	58488.0205	58566.0533	91472.5908	92796.1403	99954.016	99152.6233
2	51977.6459	51640.041	65679.199	65072.7474	85856.2114	82258.2268
3	69213.7504	68833.6105	77764.8854	78069.9896	85939.2478	85054.9378
4	60124.2902	59566.1848	83319.0924	83316.9426	93794.9834	94103.325
5	71317.3378	71432.6496	76748.5343	76956.3233	83930.3883	83468.7652
6	59384.591	59531.6634	73687.8221	74121.1031	85087.5853	84905.276
7	56031.7629	56050.5807	68215.603	68455.8273	83728.3652	83326.3957
8	58849.1438	58906.6227	72741.8224	73046.2969	79796.3231	79495.3742
9	57077.8681	56400.72	61035.9761	61329.384	95110.8992	99091.8975
10	57858.7023	57830.2476	69404.8867	69250.9562	99433.354	99954.0197

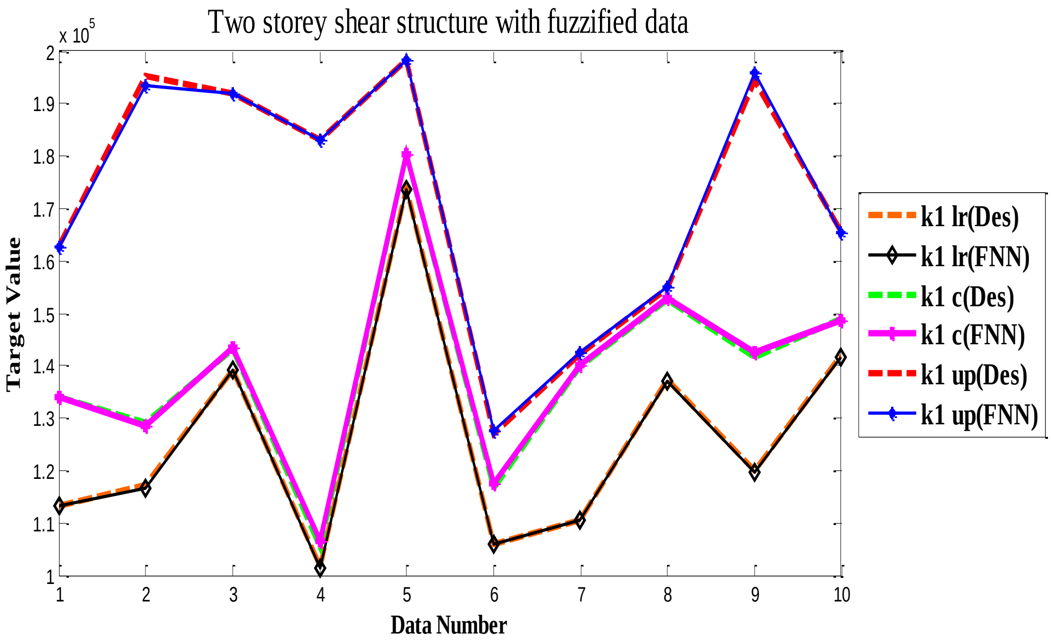

Figure 3.9(a): Comparison between desired and FNN values of $\hat{k}_1$ for a double storey shear structure

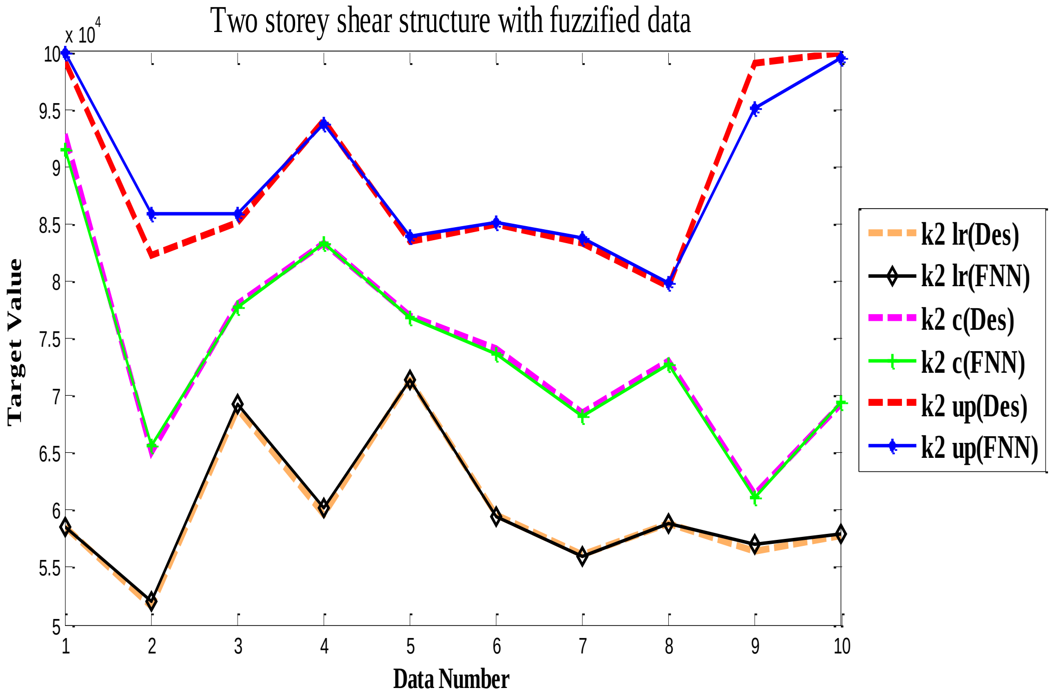

Figure 3.9(b): Comparison between desired and FNN values of $\hat{k}_2$ for a double storey shear structure

Table 3.10(a): Comparison between desired and FNN values of $\hat{k}_1$ for a double storey shear structure

Data No.	$\underline{k_1}\ (FNN)$	$\underline{k_1}\ (Des)$	$k_1c\ (FNN)$	$k_1c\ (Des)$	$\overline{k_1}\ (FNN)$	$\overline{k_1}\ (Des)$
1	62193.4503	62149.2679	65871.2249	66562.894	69219.2648	69672.8181
2	71578.6175	71235.4748	71816.4848	72130.1157	83311.5254	83571.557
3	63514.5116	63533.5212	84447.6191	84399.8043	86708.0123	87062.8972
4	59459.7849	59872.6899	67907.0231	67971.4105	75148.6421	76002.6234
5	67809.323	67385.6336	85655.9802	86827.0037	91460.6917	91106.0592
6	57879.7498	57499.8627	69206.8111	69745.3738	71387.5577	71516.0705
7	77679.4222	79304.6034	83870.4195	84180.7933	93380.175	94408.5477
8	63484.2106	63107.2659	69028.8134	69579.1498	85313.0398	85212.3715
9	52329.6174	52222.7046	72316.9085	72125.2707	88399.7894	88475.7194
10	50895.5951	50988.8812	69532.938	69859.5759	87547.575	87746.6634

Table 3.10(b): Comparison between desired and FNN values of $\hat{k}_2$ for a double storey shear structure

Data No.	$\underline{k_2}\ (FNN)$	$\underline{k_2}\ (Des)$	$k_2c\ (FNN)$	$k_2c\ (Des)$	$\overline{k_2}\ (FNN)$	$\overline{k_2}\ (Des)$
1	27396.6322	27507.0572	27776.8241	27698.5425	28078.4211	28085.141
2	21439.7109	21682.5355	27548.5904	27550.771	29081.9724	28275.8382
3	23717.5546	23773.9554	27948.7301	27919.6303	28617.4807	28629.8048
4	22090.0439	22160.1892	23180.7111	23205.2425	28962.7934	29908.7215
5	25086.5697	25154.2346	25465.6455	25360.6413	27961.1253	27904.0722
6	20724.6386	20919.5068	28645.592	28852.8102	29353.9707	29493.0391
7	20706.4881	21137.0574	23381.0249	23275.6543	25716.8788	25890.2606
8	21056.3736	21382.9255	21574.7879	21557.5235	26851.9289	26712.6437
9	22075.9397	22008.6282	24479.3131	24386.4498	27062.0237	26806.523
10	24070.1411	24079.5484	25009.5776	24971.7702	28009.5803	28335.006

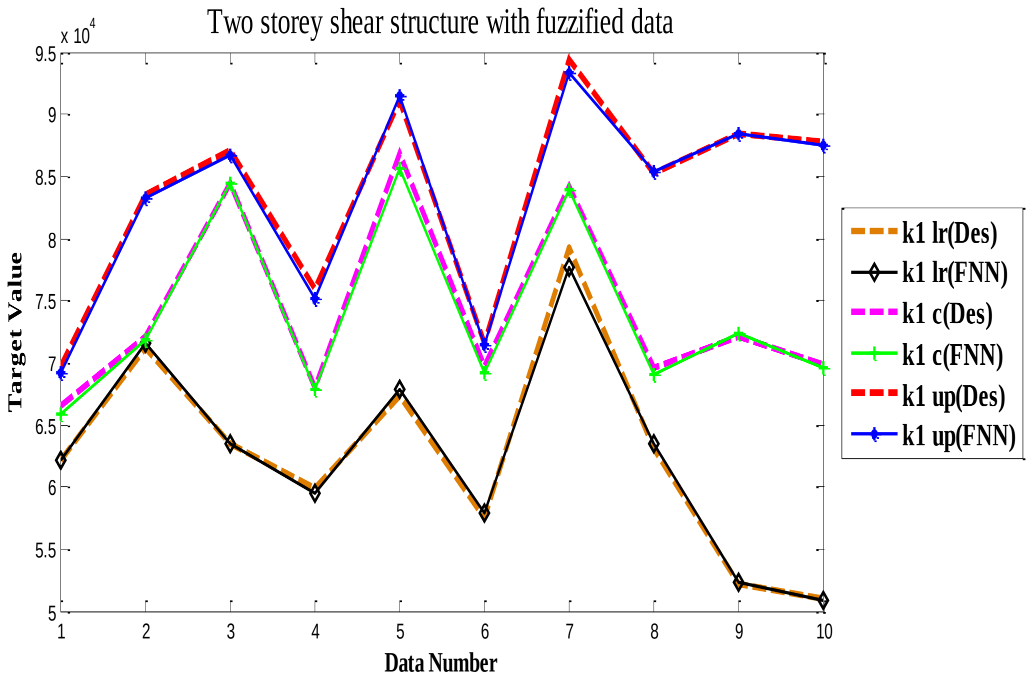

Figure 3.10(a): Comparison between desired and FNN values of $\hat{k}_1$ for a double storey shear structure.

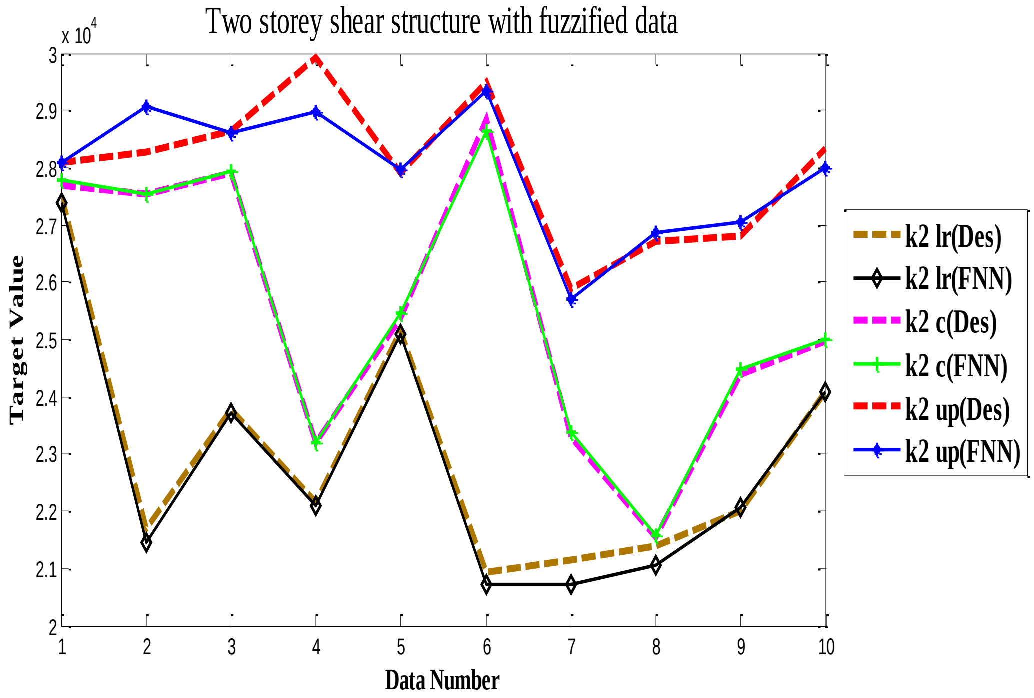

Figure 3.10(b): Comparison between desired and FNN values of $\hat{k}_2$ for a double storey shear structure.

3.5.6 Testing for Fuzzy Case

The training data with the influence of noise for two storey shear structure in TFN form for five sets of data have been presented here. Accordingly Figures 3.11(a) and 3.11(b) refer the fuzzy plot of frequency. Moreover the TFN (Triangular Fuzzy Number) plots of identified stiffness are cited in Figures 3.12(a) and 3.12(b). Also for different alpha values such as $h = 0.3$, $h = 0.5$ and $h = 0.8$, the comparison between the desired and FNN values with another five sets of data have been given in Tables 3.11(a), 3.11(b) and 3.11(c).

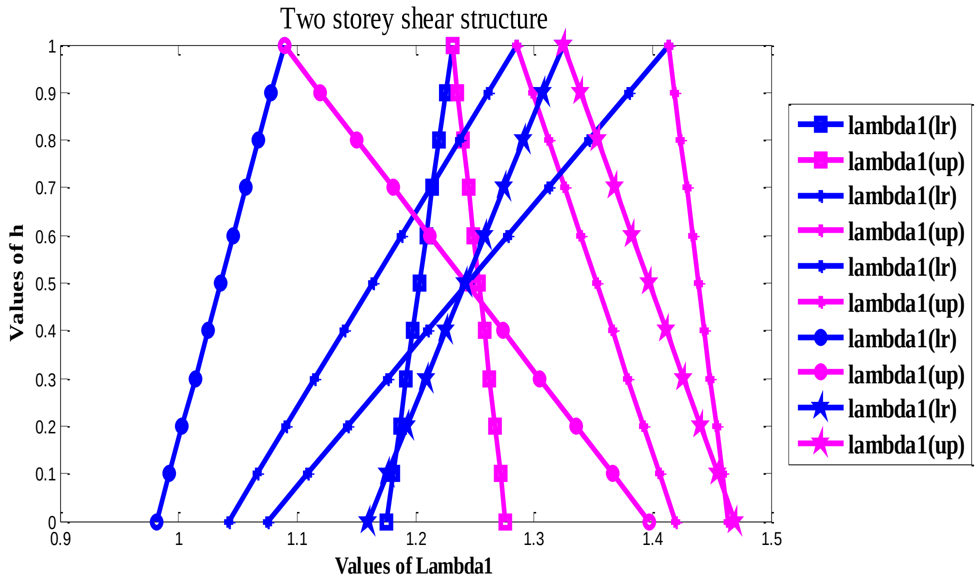

Figure 3.11(a): Comparison of $\underline{\lambda_1}$ and $\overline{\lambda_1}$ with respect to h

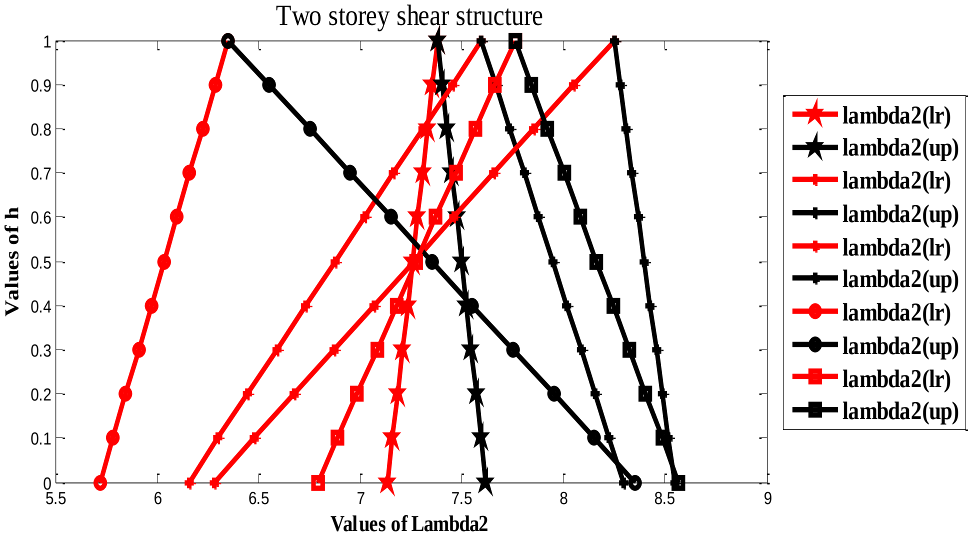

Figure 3.11(b): Comparison of $\underline{\lambda_2}$ and $\overline{\lambda_2}$ with respect to h

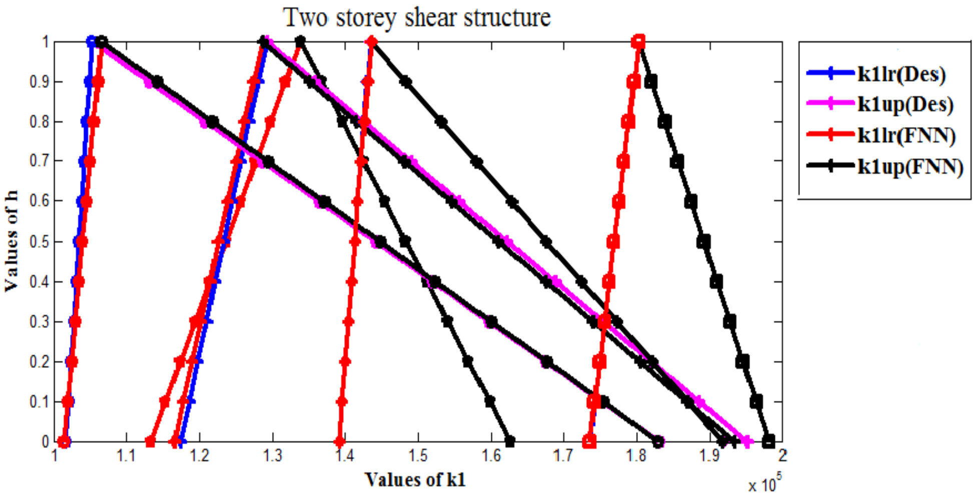

Figure 3.12(a): Comparison of $\underline{k_1}$ and $\overline{k_1}$ with respect to *h*

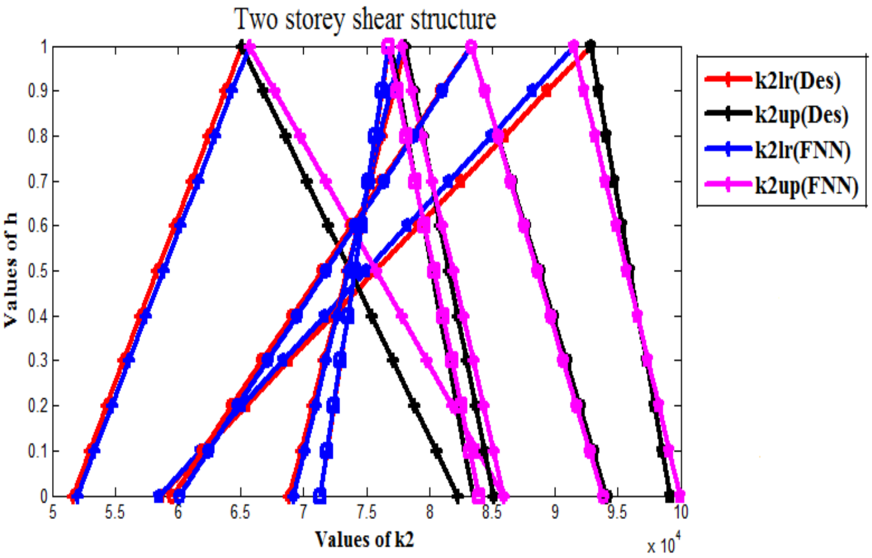

Figure 3.12(b): Comparison of $\underline{k_2}$ and $\overline{k_2}$ with respect to *h*

Table 3.11(a): Comparison between desired and FNN values of $\underline{k_1}$, $\overline{k_1}$ and $\underline{k_2}$, $\overline{k_2}$ for two storey shear structure for *h*=0.3

Data No.	$\underline{k_1}$ (FNN)	$\underline{k_1}$ (Des)	$\overline{k_1}$ (FNN)	$\overline{k_1}$ (Des)	$\underline{k_2}$ (FNN)	$\underline{k_1}$ (Des)	$\overline{k_2}$ (FNN)	$\overline{k_2}$ (Des)
1	109470	109250	124660	123870	63676	63908	81668	81670
2	119370	119420	141760	141590	59687	59772	79075	78865
3	141840	141880	154360	154170	63017	63149	77680	77561
4	126590	126380	179820	178510	58265	57879	84888	87763
5	143790	143950	160360	160670	61323	61256	90425	90743

Table 3.11(b): Comparison between desired and FNN values of $\underline{k}_1$, $\overline{k_1}$ and $\underline{k}_2$, $\overline{k_2}$ for two storey shear structure for *h*=0.5

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	$\underline{k}_2$ (FNN)	$\underline{k}_1$ (Des)	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)
1	111790	111390	122640	121830	66536	66826	79388	79513
2	125300	125280	141300	141110	62124	62253	75972	75891
3	144990	144970	153930	153750	65795	65976	76269	76271
4	131130	130750	169150	167990	59057	58865	78073	80211
5	145180	145390	157010	157330	63632	63541	84419	84602

Table 3.11(c): Comparison between the desired and the FNN values of $\underline{k}_1$, $\overline{k_1}$ and $\underline{k}_2$, $\overline{k_2}$ for two storey shear structure for *h*=0.8

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	$\underline{k}_2$ (FNN)	$\underline{k}_1$ (Des)	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)
1	115270	114590	119610	118770	70827	71203	75968	76278
2	134200	134070	140600	140400	65779	65975	71318	71430
3	149710	149600	153290	153110	69963	70218	74153	74336
4	137940	137310	153150	152200	60244	60344	67851	68882
5	147270	147540	152000	152320	67096	66967	75411	75392

3.6 Conclusion

Here procedure is demonstrated to identify stiffness parameters for multi-storey shear structure using interval and fuzzified data in ANN. Present study considers example problems of one, two, five and ten storey shear structures. At first, identification for two, five and ten storey shear structures have been done with crisp data. Then the procedure is extended for higher stories for interval and fuzzified data. Initial design parameters namely, stiffness and mass and so the design frequency of the said problem is known in term of intervals and fuzzy. Present values of the uncertain (interval and fuzzy) frequencies may be obtained by experiments. One may get the present structural parameter values of the shear structure by the proposed INN and FNN using the uncertain experimental frequencies. In order to train the new INN and FNN model, set of data are generated numerically beforehand. As such converged INN model gives the present stiffness parameter values in interval form for each floor. Thus one may predict the health of the structure when the data are available as uncertain viz. in term of interval and fuzzy.

Corresponding example problems have been solved (as mentioned) and related results are reported to show the reliability and powerfulness of the model. Although it may be interesting to have the damping in the structural model too. Also another challenge is that when we have partial or sparse data for tall storey shear buildings etc. These are important concerns to be investigated for the related problems of system identification using interval.

Chapter 4

System Identification from Response Data Using Interval and Fuzzy Neural Network

This chapter uses the concept of Interval Neural Network (INN) and Fuzzy Neural Network (FNN) modeling for the identification of structural parameters of multi storey shear buildings. First the identification has been done using response of the structure subject to ambient vibration with interval and fuzzy initial condition. Then forced vibration with horizontal displacement in interval and fuzzified form has been used to investigate the identification procedure. The model has been developed to handle the data in interval and fuzzified form for multistory shear structure and the procedure is tested for the identification of the stiffness parameters of simple example problems using the prior values of the design parameters.

4.1 Analysis and Modelling for Interval Case

The floor masses for this application problem are assumed to be $[\underline{m}_1, \overline{m}_1], [\underline{m}_2, \overline{m}_2], \ldots, [\underline{m}_n, \overline{m}_n]$ and the stiffness $[\underline{k}_1, \overline{k}_1], [\underline{k}_2, \overline{k}_2], \ldots, [\underline{k}_n, \overline{k}_n]$ are the structural parameters which are to be identified. It may be seen that all the mass and stiffness parameters are taken in interval form. The interval n-storey shear structure is already shown in Figure 3.1. Corresponding dynamic equation of motion for n-storey (supposed as n degrees of freedom) shear structure without damping may be written as

$$\left[\tilde{M}\right]\left\{\ddot{\tilde{X}}\right\} + \left[\tilde{K}\right]\left\{\tilde{X}\right\} = \left\{\tilde{F}(t)\right\} \tag{4.1}$$

where, $\left[\tilde{M}\right] = \left[\underline{M}, \overline{M}\right]$ and $\left[\tilde{K}\right] = \left[\underline{K}, \overline{K}\right]$ are interval mass and stiffness matrices and $\left\{\tilde{F}(t)\right\} = \left\{\underline{F}(t), \overline{F}(t)\right\}$ is the interval horizontal displacement forcing function.

Let us consider that the initial conditions in interval form are given by Eq. (4.2) and (4.3) as

$$\left\{\tilde{x}(0)\right\} = \left\{\underline{x}(0), \overline{x}(0)\right\} = \left\{\tilde{x}_1(0) \quad \tilde{x}_2(0) \quad \cdots \quad \tilde{x}_n(0)\right\}^T \tag{4.2}$$

$$\left\{\dot{\tilde{x}}(0)\right\} = \left\{\underline{\dot{x}}(0), \overline{\dot{x}}(0)\right\} = \left\{\dot{\tilde{x}}_1(0) \quad \dot{\tilde{x}}_2(0) \quad \cdots \quad \dot{\tilde{x}}_n(0)\right\}^T \tag{4.3}$$

Solution of Eq. (4.1) for free vibration with given interval values of mass and stiffness gives the corresponding interval eigenvalues and eigenvectors. These are denoted respectively by $\tilde{\lambda}_i$ and $\{\tilde{A}\}_i = \{\underline{A}, \overline{A}\}_i$, $i = 1,..., n$ where $\tilde{\omega}_i^2 \left(= \tilde{\lambda}_i\right)$ are the system's interval natural frequency. It may be noted that the free vibration equation will be an interval eigenvalue problem. The interval eigenvalue and vector are obtained then by considering different sets of lower and upper stiffness and mass values. Although there exist different methods to handle interval eigenvalue problems but here the above procedure has been used so that we may handle the inverse of the matrices in crisp form separately as lower and upper value. And that is why now we will replace the '~' from all notations and will consider the case for lower form first and simultaneously for upper form. Hence the modal matrix for lower form $\{\underline{A}\}$ may be written as

$$[\underline{A}] = [\{\underline{A}\}_1 \ \{\underline{A}\}_2 \ \cdots \ \{\underline{A}\}_n] \tag{4.4}$$

Denoting the diagonal matrix made up of the eigenvalues in lower form as $\underline{\lambda}_i$, as $[\underline{\lambda}]_{n\times n}$, a new set of co-ordinates in lower form $\{\underline{y}\}$ related to the co-ordinates $\{\underline{X}\}$ is introduced by the well known transformation

$$\{\underline{X}\} = [\underline{A}]\ \{\underline{y}\} \tag{4.5}$$

If the system (4.1) is subjected to an initial velocity only then substituting Eq. (4.5) in Eq. (4.1) for ambient vibration, the following equation is obtained for the response in lower form as:

$$\{\underline{X}\} = [\underline{A}][\underline{D}][\underline{\omega}]^{-1}\ [\underline{A}]^{-1}\ \{\underline{\dot{x}}(0)\} \tag{4.6}$$

whereas for the horizontal displacement in lower form we have the equation

$$\{\underline{\ddot{y}}\} + [\underline{\lambda}]\{\underline{y}\} = [\underline{P}]^{-1}\ [\underline{A}]^T\ \{\underline{F}(t)\} \tag{4.7}$$

where

$$[\underline{P}] = [\underline{A}]^T\ [\underline{M}][\underline{A}] \tag{4.8}$$

The final response for this case may be expressed in term of the original co-ordinates $\{\underline{X}\}$ after solving Eq. (4.7) for $\underline{y}$ and then putting in Eq. (4.5). In the similar manner we can compute the response for upper form. The training patterns are now trained using Interval Error Back Propagation Training Algorithm (IEBPTA) of generalized delta learning rule.

4.2 Interval Neural Network Model

This section describes structure of multi-layer interval neural network model. The training and learning algorithm have already been discussed in Sec. 2.7, 3.2. Present interval neural network model is given in Figure 4.1. The inputs $\tilde{O}_i = \tilde{X}_i = \left[\underline{X}_i, \overline{X}_i\right]$ are responses in interval and the outputs $\tilde{O}_k = \tilde{k}_k = \left[\underline{k}_k, \overline{k}_k\right]$ are stiffness parameters in interval form.

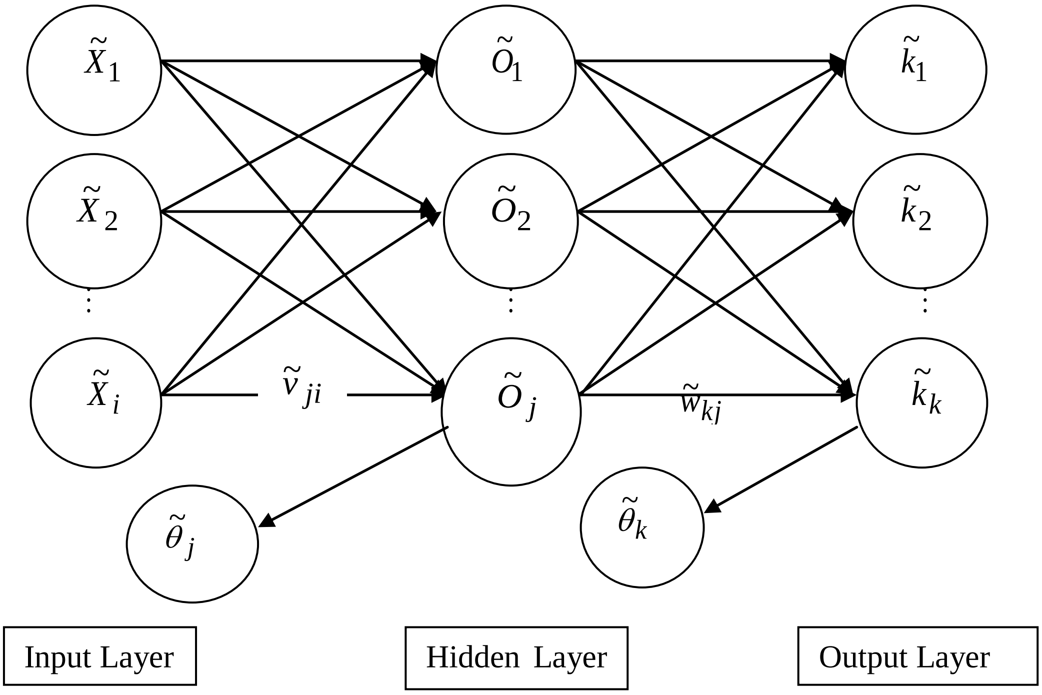

Figure 4.1: Multi-Layer Interval Neural Network Model

4.3 Analysis and Modelling for Fuzzy Case

The fuzzy n-storey shear structure is already shown in Figure 3.3. Corresponding differential equation of motion for n-storey (uncertain) structure without damping is Chakraverty [74]:

$$\left[\hat{M}\right]\left\{\hat{\ddot{X}}\right\} + \left[\hat{K}\right]\left\{\hat{X}\right\} = \left\{\hat{F}(t)\right\} \tag{4.9}$$

where, $\left[\hat{M}\right] = \left[\underline{M}, Mc, \overline{M}\right]$ and $\left[\hat{K}\right] = \left[\underline{K}, Kc, \overline{K}\right]$ are mass and stiffness matrices in fuzzy form and $\left[\hat{F}(t)\right] = \left[\underline{F}(t), Fc(t), \overline{F}(t)\right]$ is the fuzzy horizontal forcing function.

Let us consider that the initial conditions are given as

$$\{\hat{x}(0)\} = \{\hat{x}_1(0)\ \hat{x}_2(0) \cdots \hat{x}_n(0)\}^T$$

$$\{\hat{x}(0)\} = \{\hat{x}_1(0)\ \hat{x}_2(0) \cdots \hat{x}_n(0)\}^T$$

Solution of Eq. (4.9) for free vibration with given fuzzy values of mass and stiffness gives the corresponding fuzzy eigenvalues and eigenvectors. These are denoted respectively by $\hat{\lambda}_i$ and $\{\hat{A}\}_i = \{\underline{A}, Ac, \overline{A}\}_i$, $i = 1 ,..., n$ where $\hat{\omega}_i^2 = (\hat{\lambda}_i)$ are the system natural frequency in fuzzy form. It may be noted that the free vibration equation will be a fuzzy eigen value problem. The fuzzy eigen value and vector are obtained then by considering different sets of lower, centre and upper stiffness and mass values. Here the above procedure has been used so that we may handle the inverse of the matrices in crisp form separately for lower, centre and upper values. And that is why now we will drop '^' from all notations and will consider the case for lower form first and similarly for centre and upper form. Hence the modal matrix for lower form $\{\underline{A}\}$ may be written as

$$[\underline{A}] = [\{\underline{A}\}_1 \quad \{\underline{A}\}_2 \quad \cdots \quad \{\underline{A}\}_n] \tag{4.10}$$

Denoting the diagonal matrix made up of the eigenvalues in lower form as $\underline{\lambda}_i \ (= [\underline{\lambda}])_{n \times n}$, a new set of co-ordinates $\{\underline{y}\}$ related to the co-ordinates $\{\underline{X}\}$ is introduced by the well known transformation

$$\{\underline{X}\} = [\underline{A}]\,\{\underline{y}\} \tag{4.11}$$

If the system (4.9) is subjected to an initial velocity only then substituting Eq. (4.10) in Eq. (4.9) for ambient vibration, the following equation is obtained for the response in lower form

$$\{\underline{X}\} = [\underline{A}][\underline{D}][\underline{\omega}]^{-1}\,[\underline{A}]^{-1}\,\{\underline{\dot{x}}(0)\} \tag{4.12}$$

whereas for the horizontal displacement in lower form we have the equation

$$\{\underline{\ddot{y}}\} + [\underline{\lambda}]\{\underline{y}\} = [\underline{P}]^{-1}\,[\underline{A}]^T\,\{\underline{F}(t)\} \tag{4.13}$$

where

$$[\underline{P}] = [\underline{A}]^T\,[\underline{M}][\underline{A}] \tag{4.14}$$

The final response for this case may be expressed in term of the original co-ordinates $\{\underline{X}\}$ after solving Eq. (4.13) for $\underline{y}$ and then putting in Eq. (4.11). In similar manner we can compute the response for centre and upper form. The training patterns are now trained using Fuzzy Error Back Propagation Training Algorithm (FEBPTA).

4.4 Fuzzy Neural Network Model

This section describes structure of multi-layer fuzzy neural network model. The training and learning algorithm have already been discussed in Sec. 2.8, 3.4. Present fuzzy neural network model is given in Figure 4.2. Inputs $\lfloor \hat{O}_i \rfloor_h = \lfloor \hat{X}_i \rfloor_h = \left[\lfloor \underline{X}_i \rfloor_h, \lfloor \overline{X}_i \rfloor_h \right]$ are the frequencies and outputs $\lfloor \hat{O}_k \rfloor_h = \lfloor \hat{k}_k \rfloor_h = \left[\lfloor \underline{k}_k \rfloor_h, \lfloor \overline{k}_k \rfloor_h \right]$ are the stiffness parameters which are in fuzzified form but are converted in *h*-level form.

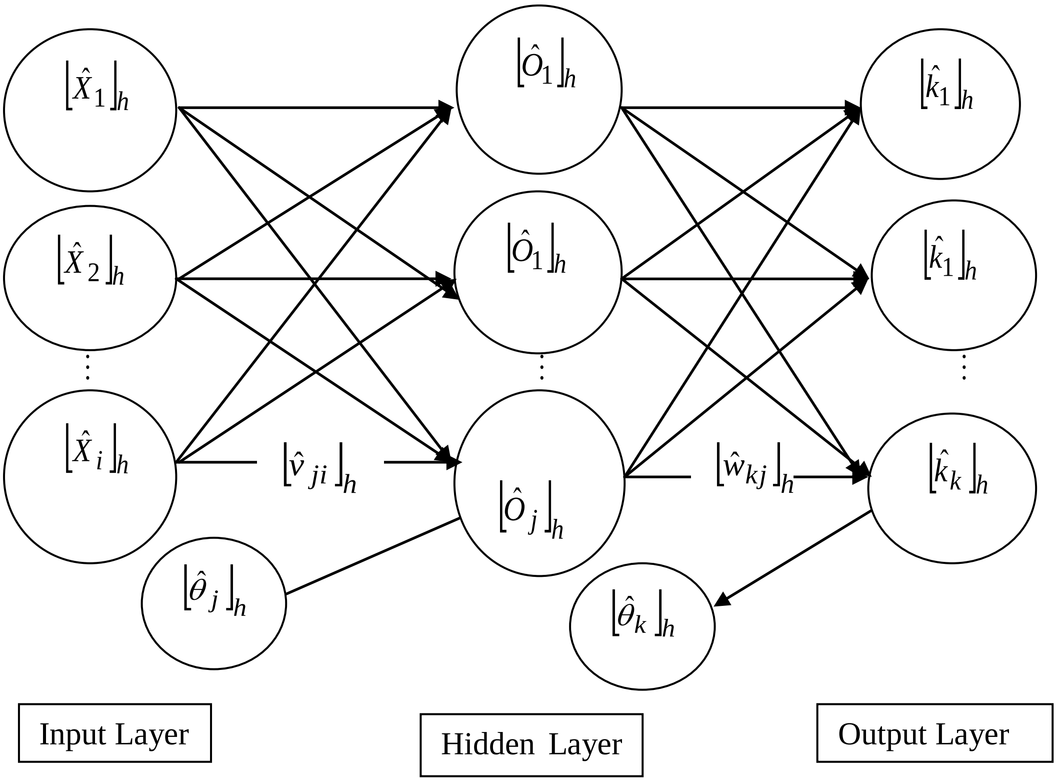
$\lfloor \hat{X}_1 \rfloor_h$
$\lfloor \hat{X}_2 \rfloor_h$
$\lfloor \hat{X}_i \rfloor_h$
$\lfloor \hat{O}_1 \rfloor_h$
$\lfloor \hat{O}_1 \rfloor_h$
$\lfloor \hat{O}_j \rfloor_h$
$\lfloor \hat{k}_1 \rfloor_h$
$\lfloor \hat{k}_1 \rfloor_h$
$\lfloor \hat{k}_k \rfloor_h$
$\lfloor \hat{v}_{ji} \rfloor_h$
$\lfloor \hat{w}_{kj} \rfloor_h$
$\lfloor \hat{\theta}_j \rfloor_h$
$\lfloor \hat{\theta}_k \rfloor_h$
Input Layer
Hidden Layer
Output Layer

Figure 4.2: Multi-Layer Fuzzy Neural Network Model

4.5 Results and Discussion

4.5.1 Interval Case

Although the developed method has been used for different storey shear structure but here only two storey shear structure has been reported to understand the methodology. To investigate the present method numerical experiment has been shown for two-storey lumped mass structure to identify interval stiffness parameters. So, we consider the floor masses for two storey shear structure in interval form as $\left[\underline{m}_1, \overline{m}_1 \right]$ and $\left[\underline{m}_2, \overline{m}_2 \right]$. Similarly the stiffness parameter may also be written in interval form as $\left[\underline{k}_1, \overline{k}_1 \right]$ and $\left[\underline{k}_2, \overline{k}_2 \right]$. For the present investigation, the masses are assumed to be constant i.e., $\underline{m}_1 = \overline{m}_1$, $\underline{m}_2 = \overline{m}_2$. One may note for identifying the interval stiffness parameters we need to have interval responses in the input nodes. In practical application due to error in measurements, we may have the response data in interval form. It is worth mentioning that the response may actually be obtained from some experiments. But here the analyses have been shown by numerical simulation only. In this respect one may see that the procedure is mentioned with constant masses but with interval stiffness parameters. To get the set of data of interval responses and interval stiffness parameters, the problem has to be solved first as forward vibration problem. For this the initial design (structural) parameters in interval form are randomized [74] and training sets of initial interval stiffness parameters are generated. For the above sets of initial interval stiffness parameters, the set of corresponding responses in interval form are generated from Eq. (4.6) for ambient vibration and from Eq. (4.5) for other case (after solving Eq. (4.8) for $\underline{y}$ or $\overline{\mathrm{y}}$). Now the mentioned neural net is trained with the interval responses that are generated from the structural parameters. When the neural net is converged, (or trained) the converged neural weight matrices $\tilde{v}_{ji}$ and $\tilde{w}_{mj}$ for hidden and output layer are stored. In order to get the interval responses for ambient vibration problem, Eq. (4.6) is used and for forced vibration problem Eq. (4.5) and Eq. (4.8) are used. The neural network training is done till a desired accuracy is reached. We will identify the stiffness parameters in interval form using the interval form of the maximum absolute response. The methodology has been discussed by giving the results for following five cases.

Case (i): Ambient vibration: interval response with crisp initial condition.
Case (ii): Ambient vibration: interval response with initial condition in interval form.
Case (iii): Forced vibration: interval response with the forcing function in crisp form.
Case (iv): Forced vibration: interval response with the forcing function in interval form.
Case (v): Interval response for both ambient and forced vibration for testing of the method with the data which are not used in the training.

A set of computer programs have been written and tested for variety of experiments for different cases and it is a gigantic task to incorporate all the results. But few of them are reported to understand the methodology. All the parameters are taken in consistent units and the data for the initial interval stiffness parameters, are considered for the academic illustrations. The input layer will have the maximum absolute interval responses for ambient as well as for forced vibration and output layer contains the corresponding interval stiffness parameters of the system. As such, the input layer will have the nodes as $\left\{ \tilde{X}_1 = \left[\underline{X}_1 \, , \, \overline{X}_1\right] \right.$ and $\left. \tilde{X}_2 = \left[\underline{X}_2 \, , \, \overline{X}_2\right] \right\}$ and output layer will have the nodes as $\left\{ \tilde{k}_1 = \left[\underline{k}_1 , \overline{k}_1\right] \right.$ and $\left. \tilde{k}_2 = \left[\underline{k}_2 \, , \overline{k}_2\right] \right\}$ for two storey shear structure. This neural network architecture is maintained for all the cases.

Examples for case (i):

As mentioned earlier for case (i), the system is subjected first to crisp initial condition expressed by the vector (with zero displacement) as $\left\{ \dot{x}(0) \right\} = \left\{ 10 \quad -10 \right\}^T$. Two examples in case (i) have been solved. For first example, a double storey shear structure is taken where the masses are $\underline{m}_1 = \overline{m}_1 = 1$ and $\underline{m}_2 = \overline{m}_2 = 1$ and the initial stiffness parameters are within the range $\tilde{k}_1 = [1000, 2000]$ and $\tilde{k}_2 = [1000, 2000]$. In second problem the masses are taken to be the same as that of the first one and the stiffness parameters varies within the range $\tilde{k}_1 = [2200, 3200]$ and $\tilde{k}_2 = [1100, 2100]$. From these initial interval stiffness parameters we have generated 40 data for both stiffness and responses in interval form. These 40 numbers of data are used as training patterns. Here the input layer contains 2 (interval) input neurons and output layer contains 2 (interval) output neurons. Various numbers of hidden nodes are considered and the program was executed. After few runs it was seen that 6 hidden nodes are sufficient to get the desirable result. As such for the first problem, with accuracy of 0.001, the desired and ANN results for 10 numbers of data chosen from 40 data have been plotted in Figures 4.3(a) and 4.3(b). For second problem, again 10 data are summarized in Tables 4.1(a) and 4.1(b).

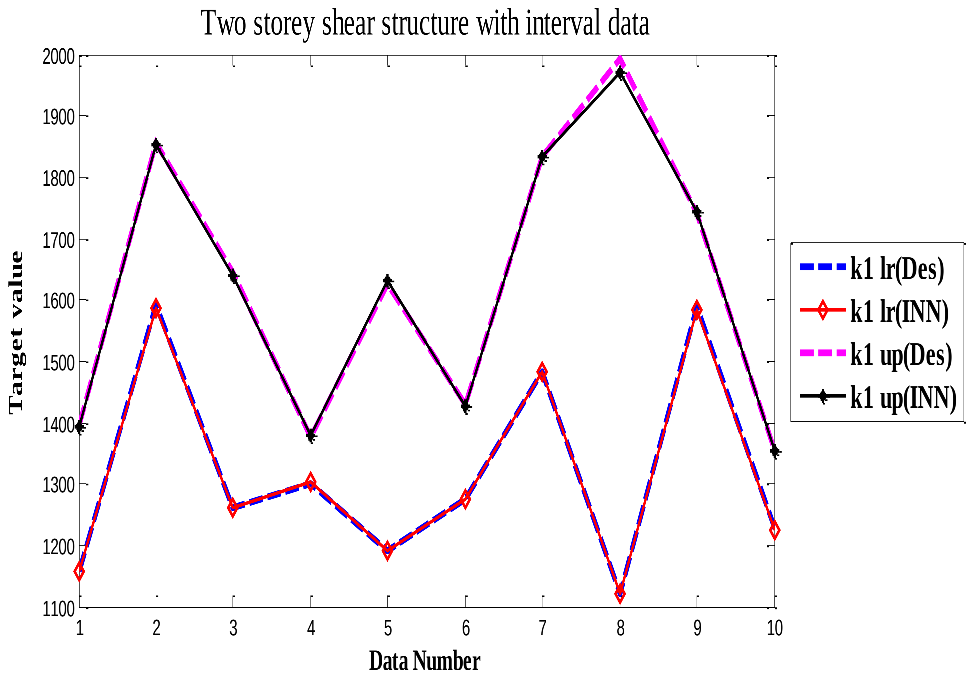

Figure 4.3(a): Comparison of desired and INN value for ambient vibration with crisp initial condition for $\underline{k}_1$, $\overline{k}_1$ for case (i), (first example)

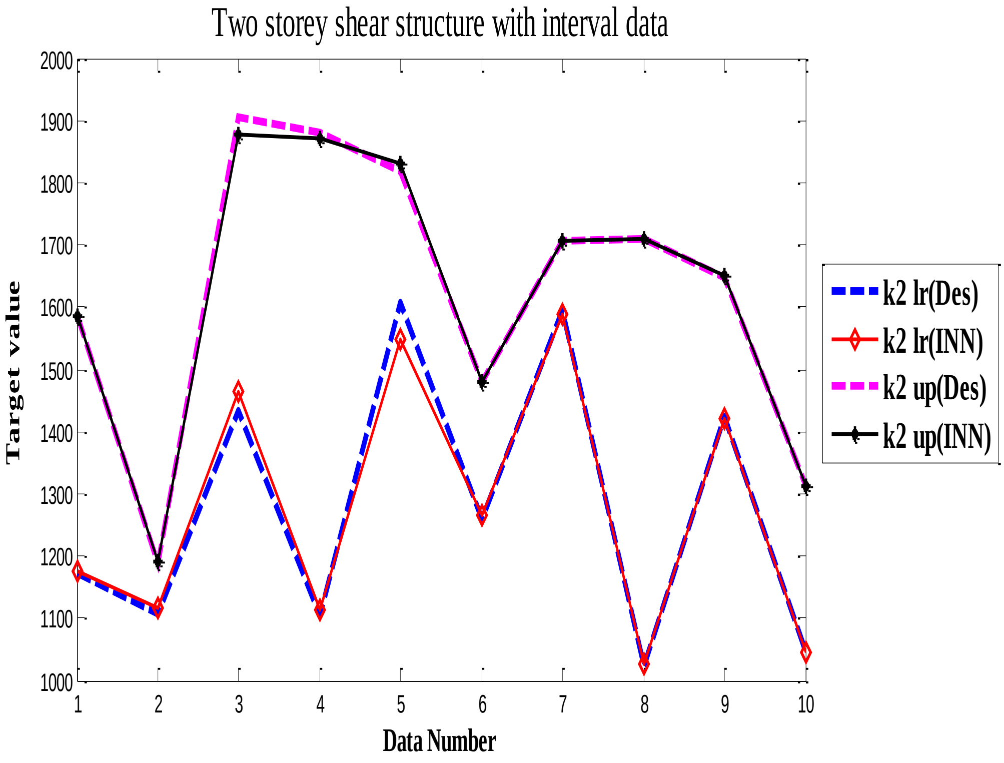

Figure 4.3(b): Comparison of desired and INN value for ambient vibration with crisp initial condition for $\underline{k}_2$, $\overline{k}_2$ for case (i), (first example)

Table 4.1(a): Comparison of desired and INN value for ambient vibration with crisp initial condition for $\underline{k}_1, \overline{k}_1$ for case (i), (second example)

Data No.	$\underline{k}_1$ (INN)	$\underline{k}_1$ (Des)	Deviation %	$\overline{k}_1$ (INN)	$\overline{k}_1$ (Des)	Deviation %
1	2415	2416	0.04	2936	2935	-0.03
2	2300	2297	-0.13	3125	3171	1.45
3	2925	2982	1.91	3018	3067	1.6
4	2290	2286	-0.17	2398	2416	0.75
5	2567	2566	-0.04	2625	2598	-1.04
6	2575	2569	-0.23	2784	2762	-0.8
7	2427	2439	0.49	2844	2885	1.42
8	2803	2798	-0.18	2848	2852	0.14
9	2693	2694	0.04	3036	2989	-1.57
10	2361	2362	0.04	2564	2568	0

Table 4.1(b): Comparison of desired and INN value for ambient vibration with crisp initial condition for $\underline{k}_2, \overline{k}_2$ for case (i), (second example)

Data No.	$\underline{k}_2$ (INN)	$\underline{k}_2$ (Des)	Deviation %	$\overline{k}_2$ (INN)	$\overline{k}_2$ (Des)	Deviation %
1	1401	1402	0.07	1890	1882	-0.43
2	1202	1201	-0.08	1710	1714	0.23
3	1393	1394	0.07	2013	2074	2.94
4	1340	1337	-0.22	1517	1542	1.62
5	1637	1631	-0.37	1845	1805	-2.22
6	1193	1191	-0.17	1878	1868	-0.54
7	1495	1505	0.66	1555	1543	-0.78
8	1210	1205	-0.41	1758	1765	0.4
9	1215	1212	-0.25	1227	1220	-0.57
10	1874	1884	0.53	2004	2044	1.96

Examples for case (ii):

In case (ii) two problems have been solved for two storey shear structure. Here the system is subjected to initial condition expressed by the vector (with zero displacement) in interval form as $\left\{\underline{\dot{x}(0)}\ \ \overline{\dot{x}(0)}\right\} = \left\{(8\ ,\ 10)\ \ \ (-10\ ,-8)\right\}^T$. The masses are kept constant for both the problems and are taken as $\underline{m}_1 = \overline{m}_1 = 1$ and $\underline{m}_2 = \overline{m}_2 = 1$. The initial interval stiffness parameter for first problem is considered as $\tilde{k}_1 = [1000, 2000]$ and $\tilde{k}_2 = [1000, 2000]$ and for the second example the initial interval stiffness are taken as $\tilde{k}_1 = [2200, 3200]$ and $\tilde{k}_2 = [1100, 2100]$. In this case, 50 numbers of data for both responses and structural parameters are generated from these initial interval stiffness parameters. The neural network architecture is similar to case (i). Again various numbers of hidden nodes are taken as per the desired accuracy and finally 8 hidden nodes are found to be sufficient to get an accuracy of 0.001. After training with 50 numbers of data, we incorporate 10 numbers of data for comparison of the desired and INN values for the first problem in Tables 4.2(a) and 4.2(b). For second problem, comparison between the desired and INN values for 10 data chosen from 50 numbers of data are plotted in Figures 4.4(a) and 4.4(b).

Table 4.2(a): Comparison of desired and INN value for ambient vibration with interval initial condition for $\underline{k}_1$, $\overline{k}_1$ for case (ii), (first example)

Data No.	$\underline{k}_1$ (INN)	$\underline{k}_1$ (Des)	Deviation %	$\overline{k}_1$ (INN)	$\overline{k}_1$ (Des)	Deviation %
1	1144	1145	0.09	1251	1250	-0.08
2	1451	1427	-1.68	1836	1853	0.92
3	1063	1060	-0.28	1617	1622	0.31
4	1349	1351	0.15	1953	1913	-2.09
5	1513	1513	0	1930	1955	1.28
6	1420	1402	-1.28	1493	1501	0.53
7	1075	1076	0.09	1496	1499	0.2
8	1223	1240	1.37	1360	1348	-0.89
9	1129	1123	-0.53	1850	1910	3.14
10	1184	1184	0	1378	1379	0.07

Table 4.2(b): Comparison of desired and INN value for ambient vibration with interval initial condition for $\underline{k}_2$, $\overline{k}_2$ for case (ii), (first example)

Data No.	$\underline{k}_2$ (INN)	$\underline{k}_2$ (Des)	Deviation %	$\overline{k}_2$ (INN)	$\overline{k}_2$ (Des)	Deviation %
1	1401	1402	0.07	1890	1882	-0.43
2	1202	1201	-0.08	1710	1714	0.23
3	1393	1394	0.07	2013	2074	2.94
4	1340	1337	-0.22	1517	1542	1.62
5	1637	1631	-0.37	1845	1805	-2.22
6	1193	1191	-0.17	1878	1868	-0.54
7	1495	1505	0.66	1555	1543	-0.78
8	1210	1205	-0.41	1758	1765	0.4
9	1215	1212	-0.25	1227	1220	-0.57
10	1874	1884	0.53	2004	2044	1.96

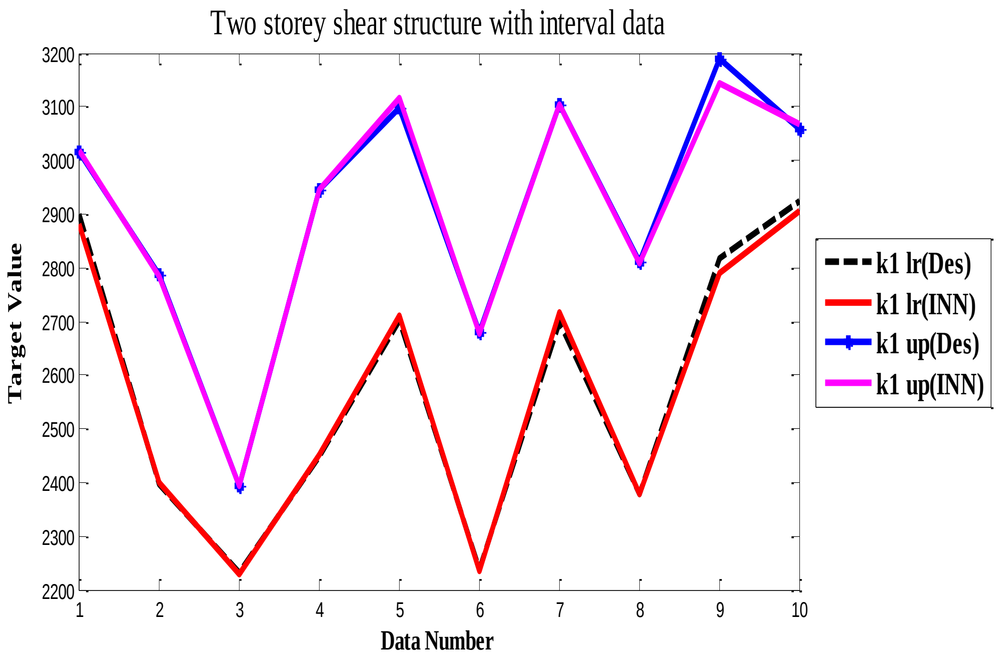

Figure 4.4(a): Comparison of desired and INN value for ambient vibration with interval initial condition for $\underline{k}_1$, $\overline{k}_1$ for case (ii), (second example)

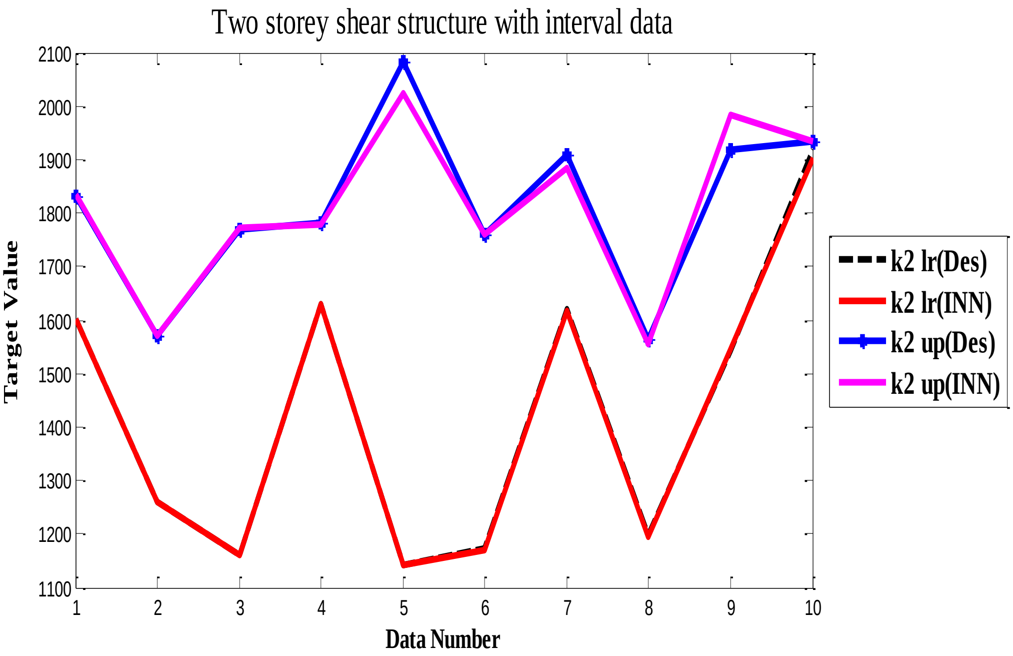

Figure 4.4(b): Comparison of desired and INN value for ambient vibration with interval initial condition for $\underline{k}_2$, $\overline{k}_2$ for case (ii), (second example)

Examples for case (iii):

Similarly for the problem with the considered horizontal displacement function, the identification of interval stiffness from interval responses with zero initial condition and the forcing function in crisp form are considered in case (iii). The forcing function vector in crisp form is defined as $F(t) = \{100\sin(1.6\pi t + \pi) \quad 100\sin(1.6\pi t)\}^T$. Again two problems have been considered for this case. The initial interval stiffness parameter used to train the first problem have values as $\tilde{k}_1 = [2000, 3000]$ and $\tilde{k}_2 = [1000, 2000]$ and for second problem, the initial interval stiffness are considered as $\tilde{k}_1 = [2200, 3200]$ and $\tilde{k}_2 = [1100, 2100]$. The masses are kept constant as that of the above cases. Here, 60 data for both responses and stiffness parameters have been generated using these initial interval structural parameters. These 60 data are used for training with 10 hidden nodes so as to get an accuracy of 0.001. After training, 10 data chosen from 60 data are again plotted in Figures 4.5(a) and 4.5(b) in order to compare the desired and INN values for the first problem. Similarly the results for second example are included in Tables 4.3(a) and 4.3(b).

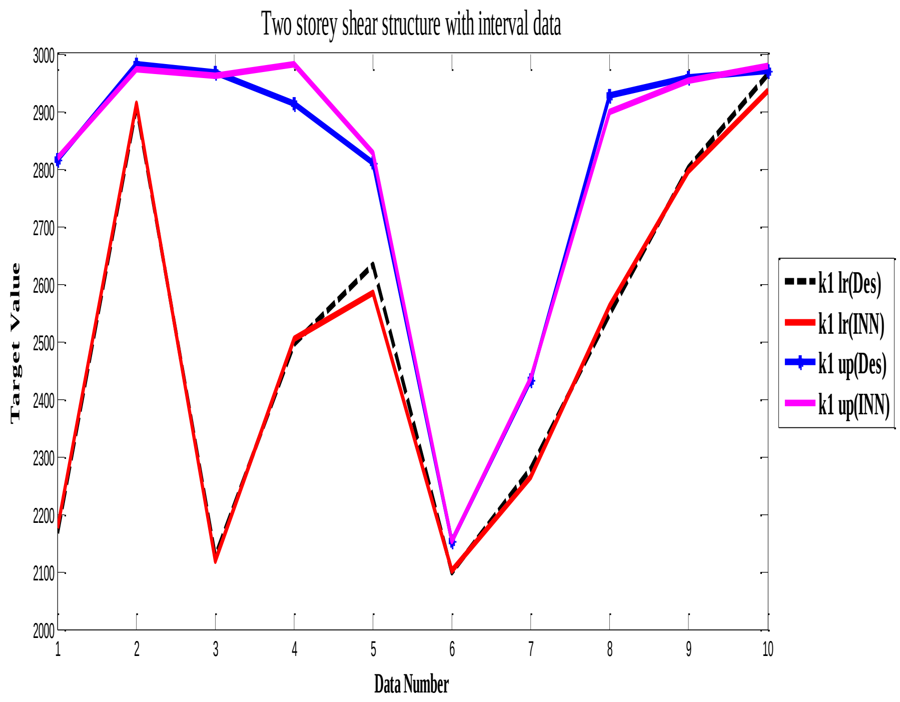

Figure 4.5(a): Comparison of desired and INN value for forced vibration with crisp forcing function for $\underline{k}_1$, $\overline{k}_1$ for case (iii), (first example)

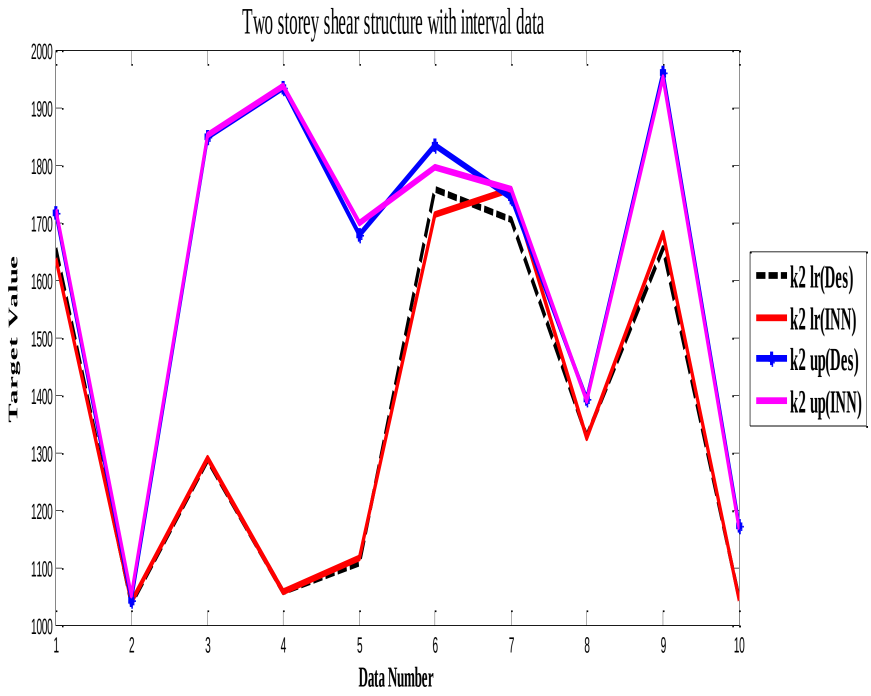

Figure 4.5(b): Comparison of desired and INN value for forced vibration with crisp forcing function for $\underline{k}_2$, $\overline{k}_2$ for case (iii), (first example)

Table 4.3(a): Comparison of desired and INN value for forced vibration with crisp forcing function for $\underline{k}_1$, $\overline{k}_1$ for case (iii), (second example)

Data No.	$\underline{k}_1$ (INN)	$\underline{k}_1$ (Des)	Deviation %	$\overline{k}_1$ (INN)	$\overline{k}_1$ (Des)	Deviation %
1	1070	1070	0	1111	1111	0
2	1242	1245	0.24	1798	1780	-1.01
3	1357	1363	0.44	1401	1390	-0.79
4	1246	1242	-0.32	1812	1831	1.04
5	1022	1025	0.29	1407	1404	-0.21
6	1066	1053	-1.23	1085	1096	1
7	1121	1132	0.97	1183	1179	-0.34
8	1693	1659	-2.05	1898	1942	2.27
9	1729	1742	0.75	1939	1956	0.87
10	1574	1575	0.06	1657	1658	0.06

Table 4.3(b): Comparison of desired and INN value for forced vibration with crisp forcing function for $\underline{k}_2$, $\overline{k}_2$ for case (iii), (second example)

Data No.	$\underline{k}_2$ (INN)	$\underline{k}_2$ (Des)	Deviation %	$\overline{k}_2$ (INN)	$\overline{k}_2$ (Des)	Deviation %
1	1161	1170	0.77	1211	1211	0
2	1330	1345	1.12	1888	1880	-0.43
3	1455	1463	0.55	1494	1490	-0.27
4	1337	1342	0.37	1944	1931	-0.67
5	1123	1125	0.18	1514	1504	-0.66
6	1145	1153	0.69	1195	1196	0.08
7	1220	1232	0.97	1282	1279	-0.23
8	1821	1759	-3.52	2043	2042	-0.05
9	1772	1842	3.8	2018	2056	1.85
10	1684	1675	-0.54	1767	1758	-0.51

Examples for case (iv):

Next, in case (iv) the forcing function vector in interval form with zero initial condition is defined as $\tilde{F}(t)=\{(80\sin(1.6\pi t+\pi)\ ,\ 100\sin(3.2\pi t+\pi))\ \ (80\sin(1.6\pi t)\ ,\ 100\sin(3.2\pi t))\}^T$. Again two problems have been solved considering the masses and stiffnesses as the previous cases. Here 80 data are used to train with 15 hidden nodes so as to get an accuracy of 0.001. Comparison between the desired and INN values for 10 data chosen from 80 numbers of data have been incorporated in Tables 4.4(a) and 4.4(b) for first problem. Similarly the results for second problem have been plotted in Figures 4.6(a) and 4.6(b).

Table 4.4(a): Comparison of desired and INN value for forced vibration with interval forcing function for $\underline{k}_1$, $\overline{k}_1$ for case (iv), (first example)

Data No.	$\underline{k}_1$ (INN)	$\underline{k}_1$ (Des)	Deviation %	$\overline{k}_1$ (INN)	$\overline{k}_1$ (Des)	Deviation %
1	2155	2145	-0.47	2251	2250	-0.04
2	2416	2427	0.45	2879	2853	-0.91
3	2118	2060	-2.82	2625	2622	-0.11
4	2338	2351	0.55	2955	2913	-1.44
5	2505	2513	0.32	2954	2955	0.03
6	2357	2402	1.87	2499	2501	0.08
7	2076	2076	0	2497	2499	0.08
8	2198	2240	1.88	2362	2348	-0.6
9	2157	2123	-1.6	2863	2910	1.62
10	2155	2184	1.33	2370	2379	0.38

Table 4.4(b): Comparison of desired and INN value for forced vibration with interval forcing function for $\underline{k}_2$, $\overline{k}_2$ for case (iv), (first example)

Data No.	$\underline{k}_2$ (INN)	$\underline{k}_2$ (Des)	Deviation %	$\overline{k}_2$ (INN)	$\overline{k}_2$ (Des)	Deviation %
1	1070	1070	0	1105	1111	0.54
2	1240	1245	0.4	1757	1780	1.29
3	1370	1363	-0.51	1370	1390	1.44
4	1244	1242	-0.16	1835	1831	-0.22
5	1029	1025	-0.39	1397	1404	0.5
6	1051	1053	0.19	1101	1096	-0.46
7	1132	1132	0	1180	1179	-0.08
8	1689	1659	-1.81	1913	1942	1.49
9	1714	1742	1.61	1931	1956	1.28
10	1571	1575	0.25	1692	1658	-2.05

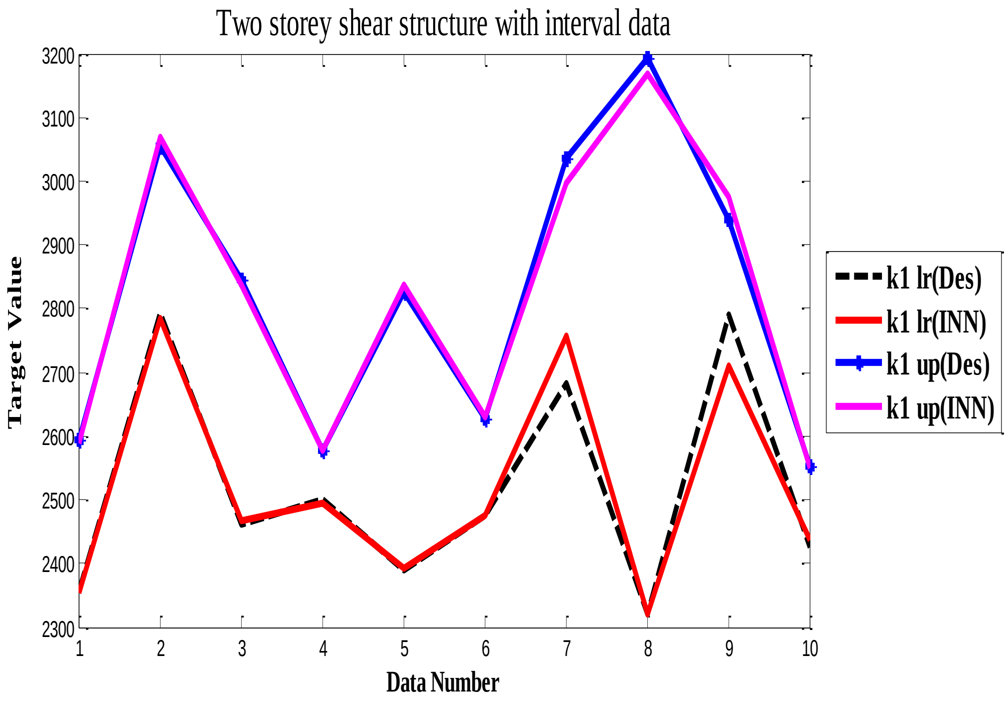

Figure 4.6(a): Comparison of desired and INN value for forced vibration with interval forcing function for $\underline{k}_1$, $\overline{k}_1$ for case (iv), (second example)

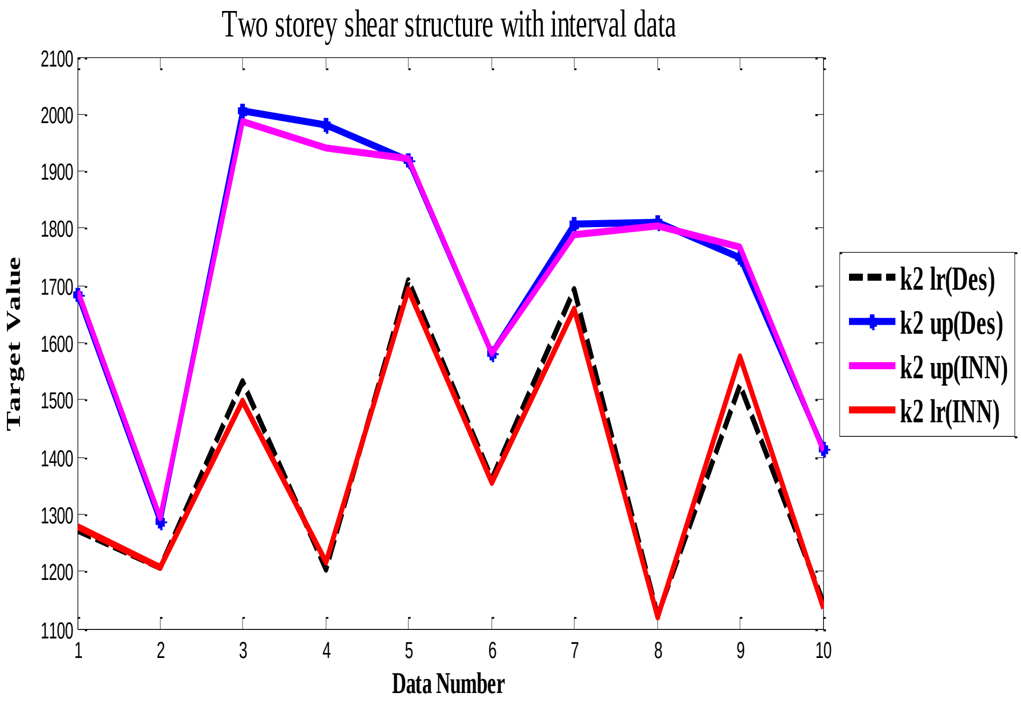

Figure 4.6(b): Comparison of desired and INN value for forced vibration with interval forcing function for $\underline{k}_2$, $\overline{k}_2$ for case (iv), (second example)

Examples for case (v) (Testing case):

Finally in case (v), two examples for testing the data which are not used (seen) during the training are considered for both ambient and forced vibration. These test data are fed into the neural network along with the stored (converged) weights to generate corresponding stiffness parameters. For the first problem, interval response with initial condition in interval form and for second problem, the interval responses with the forcing function in interval form are considered for testing. Here 10 numbers of data are taken for testing using the stored converged weights of training. Comparison between the test values of desired and INN for ambient vibration with the initial condition in interval form for 10 numbers of data are plotted in Figures 4.7(a) and 4.7(b). Again comparison between the test values of desired and INN for forced vibration with the forcing function for 10 data in interval form have been plotted in Figures 4.8(a) and 4.8(b).

It may be seen that the neural results are comparable with the desired and the deviations in percentage between them have also been shown in all the tables.

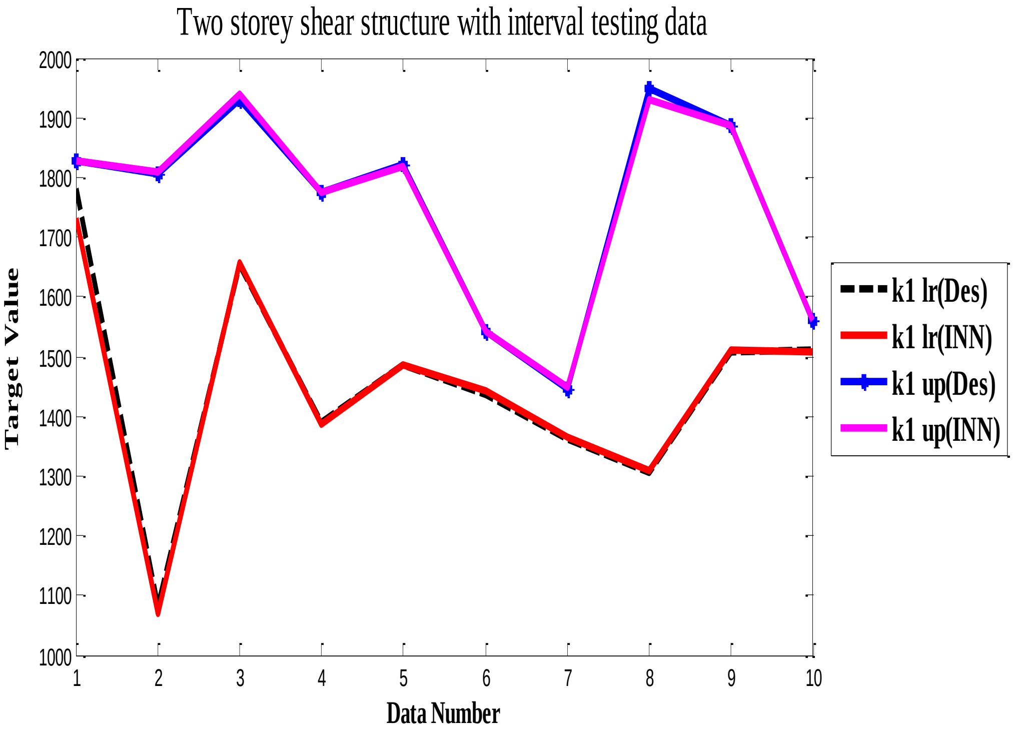

Figure 4.7(a): Comparison of desired and INN value of testing data for ambient vibration with interval initial condition for $\underline{k}_1$, $\overline{k}_1$ for case (v), (first example)

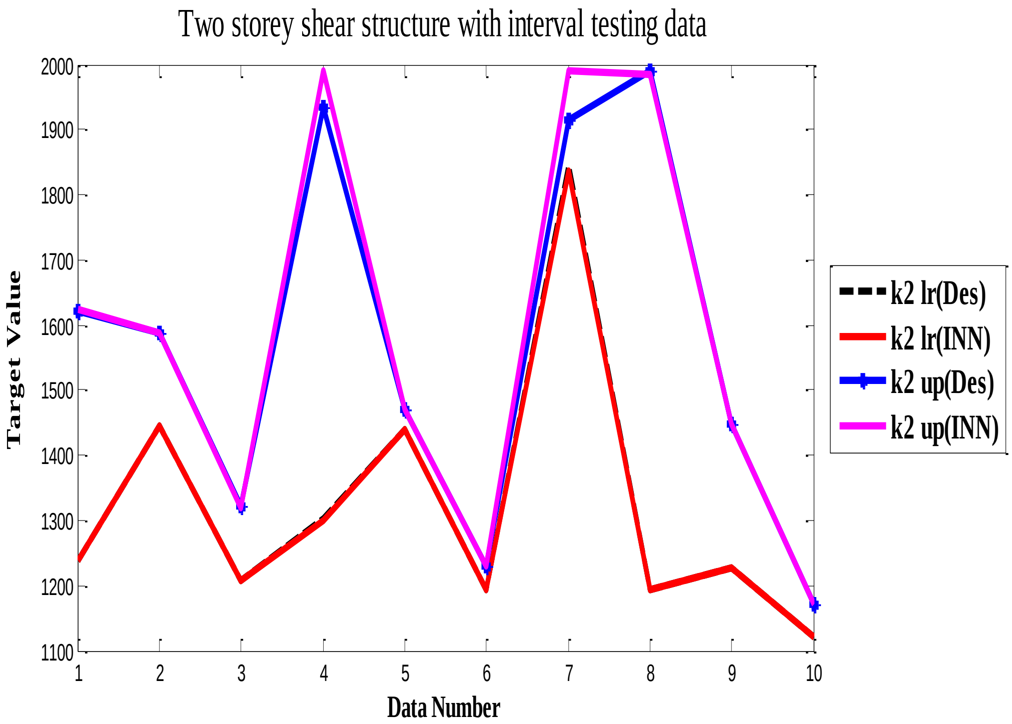

Figure 4.7(b): Comparison of desired and INN value of testing data for ambient vibration with interval initial condition for $\underline{k}_2$, $\overline{k}_2$ for case (v), (first example)

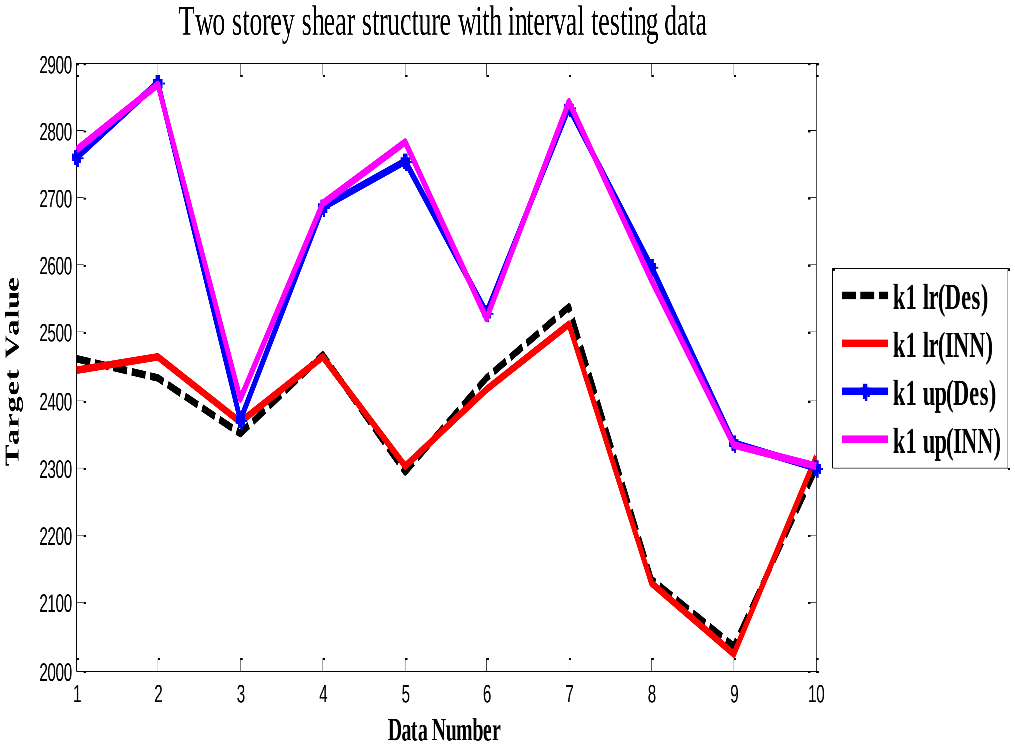

Figure 4.8(a): Comparison of desired and INN value of testing data for forced vibration with interval forcing function for $\underline{k}_1$, $\overline{k}_1$ for case (v), (second example)

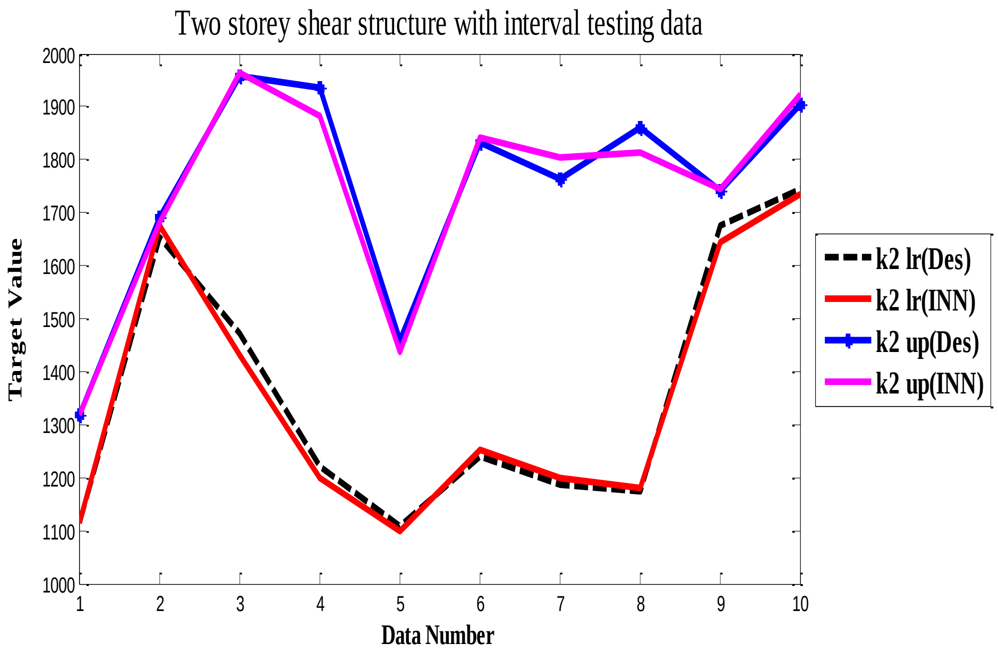

Figure 4.8(b): Comparison of desired and INN value of testing data for forced vibration with interval forcing function for $\underline{k}_2$, $\overline{k}_2$ for case (v), (second example)

4.5.2 Fuzzy Case

Although the developed method may be used for different storey shear structure but here only two storey shear structure has been reported to understand the methodology. To investigate the present method numerical experiment has been shown for two-storey lumped mass structure to identify fuzzified stiffness parameters. So, we consider the floor masses for two storey shear structure in fuzzy form as $\left[\underline{m}_1, m_1c, \overline{m}_1\right]$ and $\left[\underline{m}_2, m_1c, \overline{m}_2\right]$. Similarly the stiffness parameter may also be written in fuzzy form as $\left[\underline{k}_1, k_1c, \overline{k}_1\right]$ and $\left[\underline{k}_2, k_2c, \overline{k}_2\right]$. For the present investigation, the masses are assumed to be constant i.e., $\underline{m}_1 = m_1c = \overline{m}_1$, $\underline{m}_2 = m_2c = \overline{m}_2$. One may note that for identifying the stiffness parameters in fuzzy form we need to have fuzzified responses in the input nodes. In practical application due to error in measurements, we may have the response data in fuzzy form. It is worth mentioning that the response may actually be obtained from some experiments. But here the analyses have been shown by numerical simulation. In this respect one may see that the procedure is mentioned with constant masses but with stiffness parameters in fuzzy form. To get the set of data of responses and stiffness parameters in fuzzified form, the problem has to be solved first as forward vibration problem. For this the initial design (structural) parameters in fuzzified form are randomized as in Chakraverty [74] and training sets of initial fuzzified stiffness parameters are generated. For the above sets of initial fuzzified stiffness parameters, the set of corresponding responses in fuzzy form are generated from Eq. (4.12) for ambient vibration and from Eq. (4.11) for forced vibration case (after solving Eq. (4.14) for $\hat{y}$). Now the mentioned neural net is trained with the fuzzified responses that are generated from the structural parameters. When the neural net is converged (or trained) the converged neural weight matrices $\hat{v}_{ji}$ and $\hat{w}_{jk}$ in Figure 4.2 for hidden and output layers are stored. The neural network training is done till a desired accuracy is achieved. We will identify the stiffness parameters using fuzzy form of the maximum absolute response. The methodology has been discussed by giving the results for the following five cases.

Case (i): Ambient vibration: fuzzy response with crisp initial condition.
Case (ii): Ambient vibration: fuzzy response with initial condition in fuzzy form.
Case (iii): Forced vibration: fuzzy response with the forcing function in crisp form.
Case (iv): Forced vibration: fuzzy response with the forcing function in fuzzy form.
Case (v): Fuzzy response for both ambient and forced vibration for testing of the method with the data which are not used in the training.

All the parameters are taken in consistent units and the data for the initial fuzzified stiffness parameters are considered for the academic illustrations. The input layer will have the maximum absolute fuzzified responses for ambient as well as for forced vibration and output layer contains the corresponding fuzzified stiffness parameters of the system. As such, the input layer will have the nodes as $\{\hat{X}_1\} = \{\underline{X}_1, X_1c, \overline{X}_1\}$ and $\{\hat{X}_2\} = \{\underline{X}_2, X_2c, \overline{X}_2\}$ and output layer will have the nodes as $\{\hat{k}_1\} = \{\underline{k}_1, k_1c, \overline{k}_1\}$ and $\{\hat{k}_2\} = \{\underline{k}_2, k_2c, \overline{k}_2\}$ for two storey shear structure. This neural network architecture is maintained for all the cases.

Examples for case (i):

As mentioned earlier for case (i), the system is subjected first to crisp initial condition expressed by the vector (with zero displacement) as $\{\dot{x}(0)\} = \{10 \quad -10\}^T$. Two examples in case (i) have been solved. For first example, a double storey shear structure is taken where the masses are $\underline{m}_1 = m_1c = \overline{m}_1 = 1$ and $\underline{m}_2 = m_2c = \overline{m}_2 = 1$ and the initial stiffness parameters are within the range $\hat{k}_1 = [2000, 2010, 2020]$ and $\hat{k}_2 = [1000, 1010, 1020]$. In second problem the masses are taken to be the same as that of the first one and the stiffness parameters vary within the range $\hat{k}_1 = [2200, 2210, 2220]$ and $\hat{k}_2 = [1100, 1110, 1120]$. From these initial fuzzy stiffness parameters we have generated 50 sets of data for both stiffness and responses. These 50 training pattern are used in this case for training. Here the input layer contains 2 input neurons and output layer contains 2 output neurons. Various numbers of hidden nodes are considered and the program was executed. After few runs it was seen that 8 hidden nodes are sufficient to get the desired result. As such for the first problem (with accuracy of 0.001), the desired and FNN results for 10 data among all are included in Tables 4.5(a) and 4.5(b). These tables are plotted as TFN (Triangular Fuzzy Number) in Figures 4.9(a) and 4.9(b) for 5 data. For second problem, the data are tabulated in Tables 4.6(a) and 4.6(b) and are plotted as TFN in Figures 4.10(a) and 4.10(b) for 5 data.

Table 4.5(a): Comparison of desired and FNN value for ambient vibration with crisp initial condition for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2167.9821	2196.8551	1.31	2395.5009	2397.2454	0.07	2918.4387	2950.8944	1.09
2	2156.3645	2152.1872	-0.19	2442.3961	2443.9642	0.06	2985.6778	2977.384	-0.27
3	2040.4593	2035.135	-0.26	2055.1361	2060.0188	0.23	2283.5102	2285.322	0.07
4	2389.8985	2431.1123	1.69	2839.5843	2866.7499	0.94	2936.7227	2944.5809	0.26
5	2186.7706	2194.1003	0.33	2242.2977	2243.7704	0.06	2627.6441	2631.1887	0.13
6	2325.1742	2355.0737	1.26	2398.3027	2393.5638	-0.19	2721.1542	2735.7753	0.53
7	2103.1454	2107.5003	0.20	2380.0314	2380.3627	0.01	2996.9129	2997.0033	0.00
8	2304.007	2224.1715	-3.58	2656.3394	2660.1165	0.14	2903.6481	2851.5601	-1.82
9	2186.9704	2200.6169	0.62	2633.3905	2652.4511	0.71	2745.2227	2744.2297	-0.03
10	2069.4259	2065.0511	-0.21	2596.2003	2581.0259	-0.5	2599.6623	2604.9906	0.20

Table 4.5(b): Comparison of desired and FNN value for ambient vibration with crisp initial condition for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1052.5241	1049.992	-0.24	1382.3412	1393.3063	0.07	1749.057	1723.1735	-0.01
2	1346.2086	1347.4376	0.09	1560.4724	1555.6642	-0.03	1630.5815	1627.2792	-0.20
3	1106.8461	1107.0772	0.02	1591.2028	1585.4949	-0.03	1659.9549	1660.6168	0.03
4	1385.9187	1383.8686	-0.14	1538.0404	1540.0517	0.13	1829.7184	1822.0914	-0.41
5	1285.0788	1285.0698	-0.00	1625.378	1627.3465	0.12	1982.1358	2009.1449	0.01
6	1021.6113	1021.6498	0.00	1084.868	1086.9463	0.19	1265.4937	1258.629	0.54
7	1461.8393	1461.6388	-0.01	1901.5472	1910.57	0.47	1953.1503	1959.3984	0.31
8	1037.8535	1038.1775	0.03	1251.0481	1237.7128	-0.01	1774.4738	1800.5587	0.01
9	1714.5838	1703.8386	-0.63	1733.1448	1745.8475	0.72	1861.557	1814.4496	-0.02
10	1772.0203	1803.7365	1.75	1827.4152	1813.1128	-0.78	1940.5288	1996.1042	0.02

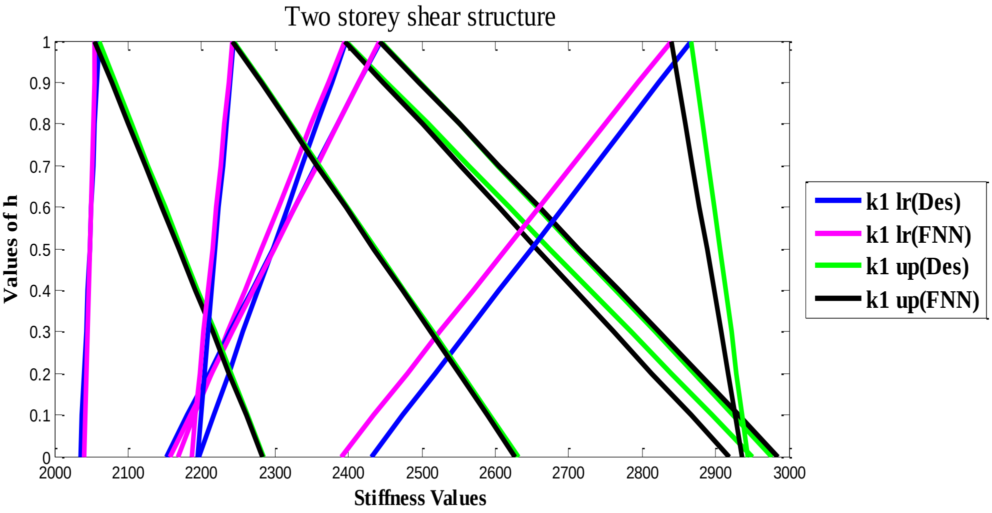

Figure 4.9(a): TFN for ambient vibration with crisp initial condition for $[\underline{k}_1]_h, [\overline{k}_1]_h$

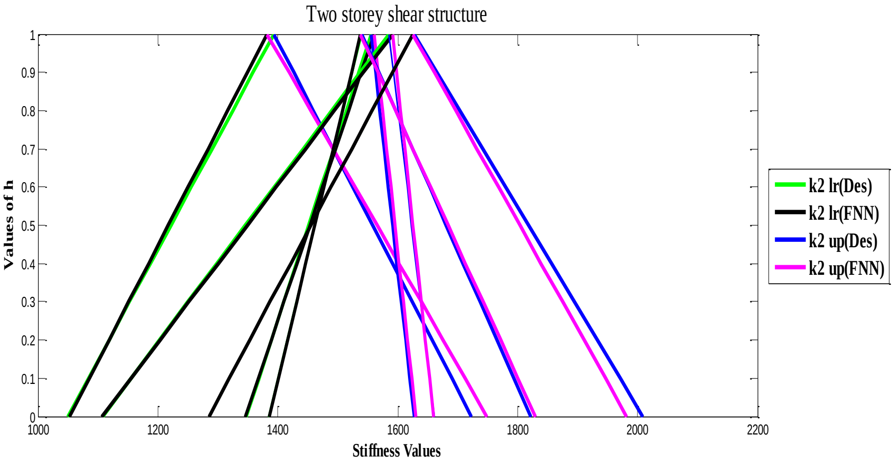

Figure 4.9(b): TFN for ambient vibration with crisp initial condition for $[\underline{k}_2]_h, [\overline{k}_2]_h$

Table 4.6(a): Comparison of desired and FNN value for ambient vibration with crisp initial condition for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2452.0789	2452.7854	0.02	2550.8584	2550.8579	-0.00	2592.76	2593.4564	0.02
2	2643.7267	2644.3095	0.02	2653.0107	2652.4023	-0.02	2868.7481	2871.4311	0.09
3	2491.1864	2490.2704	-0.03	2902.2437	2897.7961	-0.15	2950.3716	2941.2579	-0.30
4	2414.8809	2417.0538	0.08	2569.503	2569.2282	-0.01	2712.4648	2720.0525	0.27
5	2547.8591	2547.7127	-0.00	2924.5316	2946.3401	0.74	3045.8268	3041.7212	-0.13
6	2349.0473	2349.9973	0.04	2604.0824	2604.7075	0.02	2650.1936	2649.9214	-0.01
7	2747.0409	2786.0921	1.40	2900.6723	2893.4159	-0.25	3070.1667	3107.771	1.21
8	2465.533	2462.1453	-0.13	2609.2892	2611.183	0.07	2936.8073	2914.0474	-0.78
9	2245.9988	2244.4541	-0.06	2652.5665	2652.3054	-0.00	2994.1723	2989.1144	-0.16
10	2229.4976	2229.5776	0.00	2617.4106	2616.7915	-0.02	2947.1434	2954.9333	0.26

Table 4.6(b): Comparison of desired and FNN value for ambient vibration with crisp initial condition for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1864.4554	1868.7057	0.22	1879.8603	1878.8543	-0.05	1908.595	1908.5141	-0.00
2	1277.3513	1277.2535	-0.00	1855.9477	1855.0771	-0.04	1943.3948	1945.5838	0.11
3	1477.1244	1477.3955	0.01	1914.1053	1909.963	-0.21	2031.7836	1971.9805	-3.03
4	1316.0031	1316.0189	0.00	1439.241	1438.5242	-0.04	2036.8438	2099.8722	3.00
5	1624.9836	1624.4235	-0.03	1653.2142	1654.0641	0.05	1881.2284	1890.4072	0.48
6	1210.3236	1209.9507	-0.03	1973.5848	1994.281	1.03	2022.2013	2049.3039	1.32
7	1232.0312	1231.7057	-0.02	1428.5537	1427.5654	-0.06	1694.1558	1698.0261	0.22
8	1256.4596	1256.2925	-0.01	1265.1769	1264.7523	-0.03	1782.3289	1771.2644	-0.62
9	1309.4382	1309.8628	0.03	1540.1715	1538.645	-0.09	1808.0597	1798.6523	-0.52
10	1517.1897	1516.9548	-0.01	1614.0209	1615.177	0.07	1922.9195	1933.5006	0.54

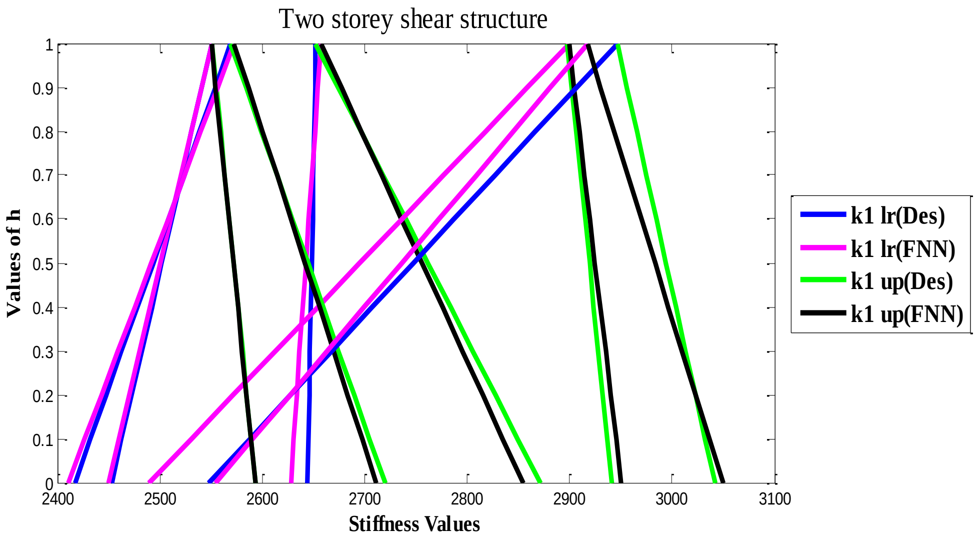

Figure 4.10(a): TFN for ambient vibration with crisp initial condition for $[\underline{k}_1]_h, [\overline{k}1]_h$

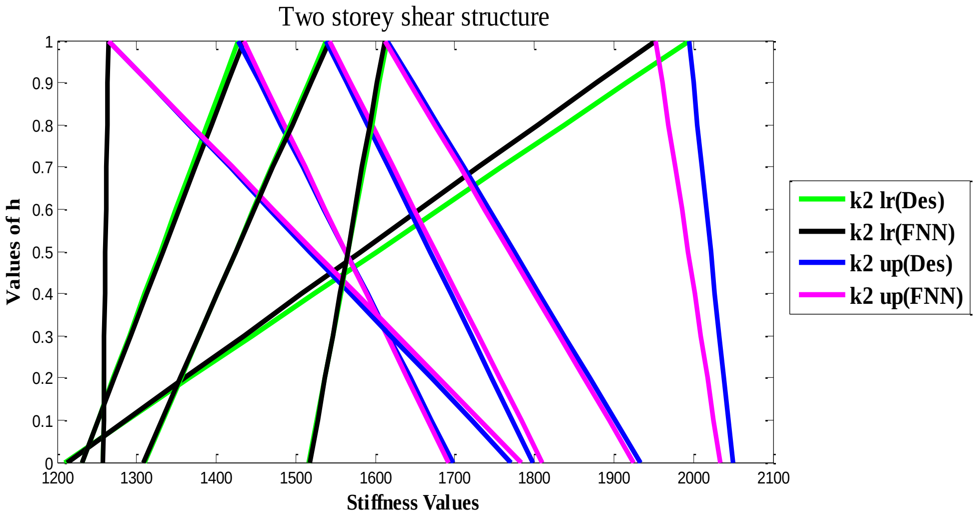

Figure 4.10(b): TFN for ambient vibration with crisp initial condition for $[\underline{k}_2]_h, [\overline{k}2]_h$

Examples for case (ii):

In case (ii), two problems have been solved for two storey shear structure. Here the system is subjected to initial condition expressed by the vector (with zero displacement) in fuzzy form as $\left\{\underline{\dot{x}(0)}, \dot{x}c(0), \overline{\dot{x}(0)}\right\} = \{(8\ ,10,\ 12)\ \ (-12\ ,-10,-8)\}^T$. The masses are kept constant for both the problems and are taken as $\underline{m}_1 = m_1c = \overline{m}1 = 1$ and $\underline{m}_2 = m_2c = \overline{m}2 = 1$. The initial fuzzified stiffness parameter for first problem is considered as $\hat{k}_1 = [2000, 2010, 2020]$ and $\hat{k}_2 = [1000, 1010, 1020]$ and for the second example the initial stiffness in fuzzy form are taken as $\hat{k}_1 = [2200, 2210, 2220]$

and $\hat{k}_2 = [1100, 1110, 1120]$. 60 sets of data for both responses and structural parameters are generated from these initial fuzzy stiffness parameters. In this case the neural network architecture is similar as in case (i). Again various numbers of hidden nodes are taken as per the desired accuracy and finally 10 hidden nodes are found to be sufficient to get an accuracy of 0.001. After training 10 trained data among 60 are incorporated for comparison of the desired and FNN values for the first problem in Tables 4.7(a) and 4.7(b). These tables have been plotted as TFN in Figures 4.11(a) and 4.11(b) for 5 data. For second problem comparison between the desired and FNN values for 10 data among 60 data have been tabulated in Tables 4.8(a) and 4.8(b). These tables are plotted as TFN in Figures 4.12(a) and 4.12(b) for 5 data.

Table 4.7(a): Comparison of desired and FNN value for ambient vibration with fuzzy initial condition for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2712.4432	2720.6895	0.30	2740.2729	2740.9997	0.02	2991.8529	3004.207	0.41
2	2341.1669	2339.3179	-0.07	2884.3239	2912.9671	0.98	2924.7706	2937.651	0.43
3	2055.0991	2053.022	-0.10	2411.9965	2416.0909	0.16	2746.0776	2746.0249	-0.00
4	2193.4089	2193.7959	0.01	2556.6636	2557.716	0.04	2604.2448	2605.5213	0.04
5	2011.5635	2011.4941	-0.00	2907.4145	2947.057	1.34	2954.3946	2972.2081	0.59
6	2135.0845	2135.5559	0.02	2375.4562	2380.7479	0.22	2730.3737	2734.7614	0.16
7	2108.9329	2111.2523	0.10	2466.566	2469.3742	0.11	2627.2043	2628.769	0.05
8	2381.8233	2382.333	0.02	2505.8781	2506.9169	0.04	2852.4095	2851.3387	-0.03
9	2185.5594	2185.5842	0.00	2245.5523	2246.8234	0.05	2302.7568	2304.1591	0.06
10	2065.9854	2065.1411	-0.04	2278.5417	2277.3245	-0.05	2432.1578	2431.3957	-0.03

Table 4.7(b): Comparison of desired and FNN value for ambient vibration with fuzzy initial condition for $\underline{k}_2, k_2c, \overline{k}_2$.

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1001.7499	1001.7441	-0.00	1174.3756	1174.3606	-0.00	1658.16	1658.1989	0.00
2	1393.5595	1390.5963	-0.21	1755.8986	1763.2508	0.41	1866.5971	1850.7432	-0.85
3	1028.7436	1027.1498	-0.15	1241.5602	1242.8138	0.10	1781.5498	1763.7961	-1.00
4	1016.1336	1015.6761	-0.04	1044.8147	1046.2616	0.13	1849.3909	1854.0591	0.25
5	1124.9853	1123.7487	-0.11	1200.824	1201.0798	0.02	1782.1465	1781.8965	-0.01
6	1082.2105	1080.4416	-0.16	1264.1739	1263.8489	-0.02	1300.3199	1299.1074	-0.09
7	1074.5898	1078.0735	0.32	1711.6886	1710.6337	-0.06	1956.9678	2017.1184	2.98
8	1572.4641	1591.2451	1.18	1675.2801	1674.9342	-0.02	1915.8622	1923.0426	0.37
9	1137.2544	1137.8338	0.05	1495.6981	1496.1279	0.02	2015.6705	1954.8971	-3.10
10	1278.0244	1277.9735	-0.00	1492.2765	1492.9389	0.04	1855.291	1844.7672	-0.57

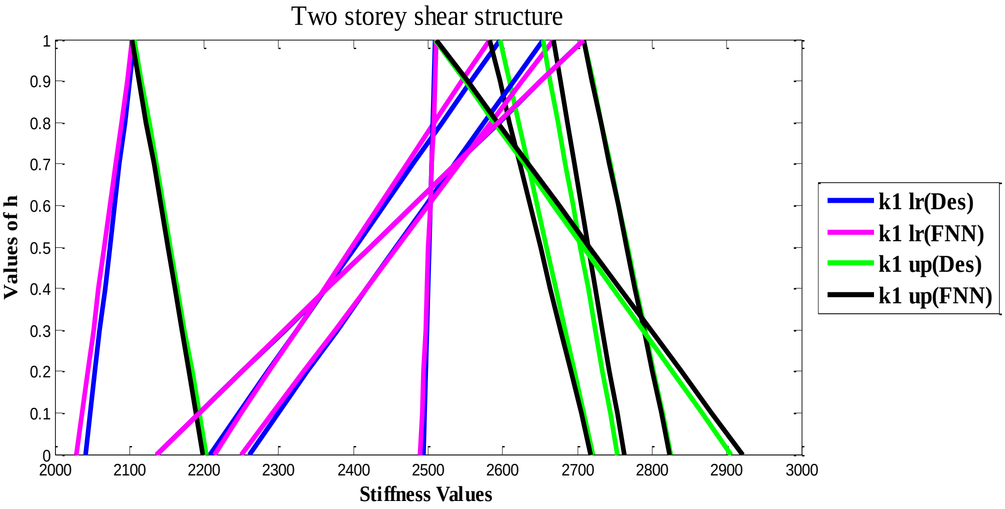

Figure 4.11(a): TFN for ambient vibration with fuzzy initial condition for $[\underline{k}_1]_h, [\overline{k}_1]_h$

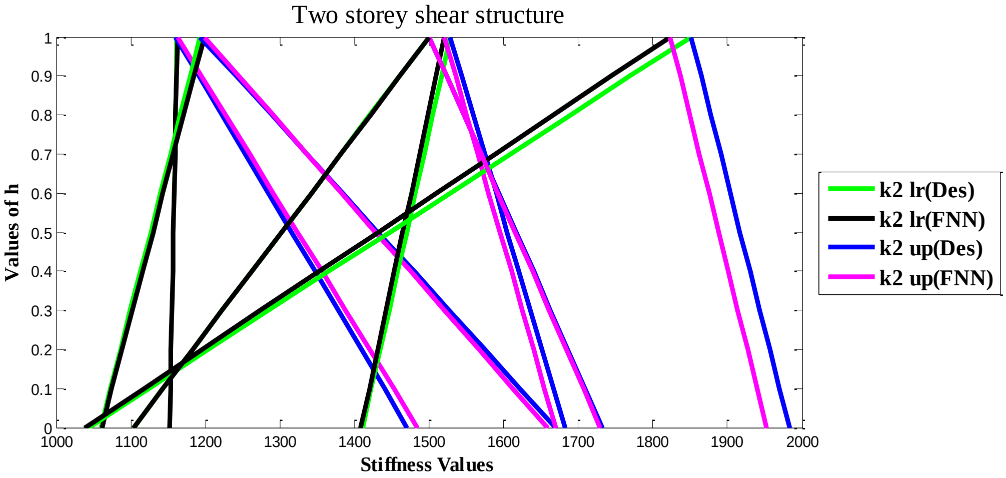

Figure 4.11(b): TFN for ambient vibration with fuzzy initial condition for $[\underline{k}_2]_h, [\overline{k}_2]_h$

Table 4.8(a): Comparison of desired and FNN value for ambient vibration with fuzzy initial condition for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2450.1683	2452.7854	0.10	2550.3655	2550.8579	0.01	2593.4768	2593.4564	-0.00
2	2627.7658	2644.3095	0.62	2657.3922	2652.4023	-0.18	2854.747	2871.4311	0.58
3	2489.5777	2490.2704	0.02	2899.467	2897.7961	-0.05	2950.4275	2941.2579	-0.31
4	2410.4906	2417.0538	0.27	2572.5709	2569.2282	-0.13	2711.6029	2720.0525	0.31
5	2554.0384	2547.7127	-0.24	2917.3193	2946.3401	0.98	3049.7731	3041.7212	-0.26
6	2352.0903	2349.9973	-0.08	2599.3815	2604.7075	0.20	2656.5113	2649.9214	-0.24
7	2680.6273	2786.0921	3.78	2866.7292	2893.4159	0.92	3048.2152	3107.771	1.91
8	2494.5878	2462.1453	-1.31	2600.9786	2611.183	0.39	2970.086	2914.0474	-1.92
9	2244.8626	2244.4541	-0.01	2661.3343	2652.3054	-0.34	2991.2942	2989.1144	-0.07
10	2228.7874	2229.5776	0.03	2620.263	2616.7915	-0.13	2941.8077	2954.9333	0.44

Table 4.8(b): Comparison of desired and FNN value for ambient vibration with fuzzy initial condition for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1859.8134	1868.7057	0.47	1885.5137	1878.8543	-0.35	1908.7286	1908.5141	-0.01
2	1277.6131	1277.2535	-0.02	1858.867	1855.0771	-0.20	1933.3046	1945.5838	0.63
3	1474.2994	1477.3955	0.20	1930.3145	1909.963	-1.06	2037.056	1971.9805	-3.30
4	1316.7153	1316.0189	-0.05	1443.1725	1438.5242	-0.32	2039.701	2099.8722	2.86
5	1629.9065	1624.4235	-0.33	1650.5398	1654.0641	0.21	1879.3819	1890.4072	0.58
6	1211.7234	1209.9507	-0.14	1952.9784	1994.281	2.07	2033.7778	2049.3039	0.75
7	1231.0913	1231.7057	0.04	1434.0897	1427.5654	-0.45	1691.6876	1698.0261	0.37
8	1257.4988	1256.2925	-0.09	1264.1406	1264.7523	0.04	1783.453	1771.2644	-0.68
9	1308.3423	1309.8628	0.11	1541.9572	1538.645	-0.21	1809.8501	1798.6523	-0.62
10	1518.0058	1516.9548	-0.06	1612.0433	1615.177	0.19	1925.4892	1933.5006	0.41

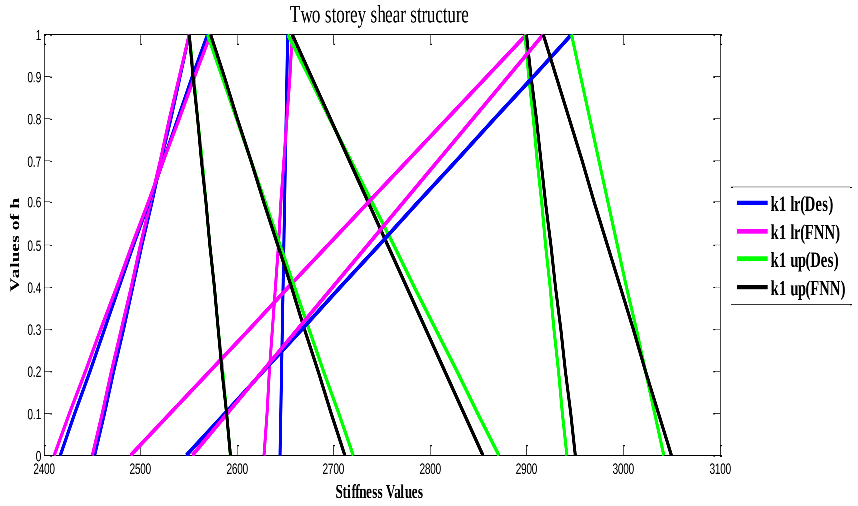

Figure 4.12(a): TFN for ambient vibration with fuzzy initial condition for $[\underline{k}_1]_h, [\overline{k}_1]_h$

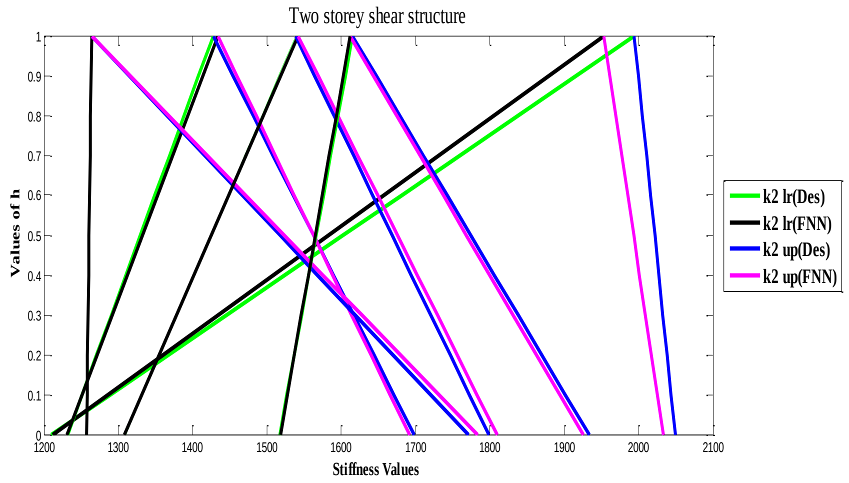

Figure 4.12(b): TFN for ambient vibration with fuzzy initial condition for $[\underline{k}_2]_h, [\overline{k}_2]_h$

Examples for case (iii):

Similarly we consider the problem with the horizontal displacement function. The identification of stiffness in fuzzy form from fuzzified responses with zero initial condition and forcing function in crisp form are considered in case (iii). The forcing function vector in crisp form is defined as $F(t) = \{100\sin(1.6\pi t + \pi) \quad 100\sin(1.6\pi t)\}^T$. Again two problems have been considered for this case. The initial fuzzified stiffness parameter used to train the first problem have values as $\hat{k}_1 = [2000, 2010, 2020]$and $\hat{k}_2 = [1000, 1010, 1020]$ and for second problem, the initial fuzzified stiffness are considered as $\hat{k}_1 = [2200, 2210, 2220]$ and $\hat{k}_2 = [1100, 1110, 1120]$. The masses are kept constant as that of the above cases. Here 80 sets of data for both responses and stiffness parameters have been generated using these initial fuzzified structural parameters. These 80 training pattern are used for training with 12 hidden nodes so as to get an accuracy of 0.001. Again 10 trained data among 80 are included in Tables 4.9(a) and 4.9(b) in order to compare the desired and FNN values for the first problem and are plotted as TFN in Figures 4.13(a) and 4.13(b) again for 5 data. Similarly the results for the second example are tabulated in Tables 4.10(a) and 4.10(b) and are plotted as TFN in Figures 4.14(a) and 4.14(b) for 5 data.

Table 4.9(a): Comparison of desired and FNN value for forced vibration with crisp forcing function for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2136.3878	2136.5531	0.00	2708.0485	2708.7458	0.02	2823.94	2825.4894	0.05
2	2213.4196	2207.8098	-0.25	2582.0387	2596.7215	0.56	2718.408	2721.2275	0.10
3	2028.9696	2040.5409	0.56	2103.9664	2106.7619	0.13	2198.3539	2202.9225	0.20
4	2249.4018	2259.932	0.46	2668.0384	2653.7573	-0.53	2762.1808	2754.0743	-0.29
5	2489.4993	2494.1739	0.18	2511.5788	2510.0224	-0.06	2920.8248	2906.5119	-0.49
6	2054.8604	2048.6742	-0.30	2485.551	2489.9221	0.17	2771.0529	2779.0517	0.28
7	2481.5327	2509.9014	1.13	2744.4885	2715.0371	-1.08	2952.6467	2914.7222	-1.30
8	2203.6758	2187.9271	-0.71	2612.5518	2619.8666	0.27	2888.4125	2903.7206	0.52
9	2626.2226	2627.6664	0.05	2856.1899	2890.9225	1.20	2973.6323	2998.6806	0.83
10	2332.6513	2334.1631	0.06	2734.1007	2732.6945	-0.05	2867.9343	2869.4423	0.05

Table 4.9(b): Comparison of desired and FNN value for forced vibration with crisp forcing function for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1103.3733	1103.4698	0.00	1500.7006	1500.4716	-0.01	1729.8432	1732.4396	0.14
2	1152.0276	1153.171	0.09	1162.5774	1159.8654	-0.23	1485.1404	1471.0884	-0.95
3	1061.0412	1059.6189	-0.13	1198.3835	1193.3886	-0.41	1659.074	1669.6053	0.63
4	1408.6154	1410.9378	0.16	1520.9071	1528.5949	0.50	1669.3995	1681.9719	0.74
5	1038.8641	1042.4311	0.34	1823.6153	1851.3797	1.49	1952.7087	1982.9746	1.52
6	1066.1754	1071.4455	0.49	1667.7389	1658.9915	-0.52	1883.8507	1823.3644	-3.31
7	1080.1584	1080.4712	0.02	1512.9429	1521.6498	0.57	1769.0166	1810.3306	2.28
8	1090.8478	1096.73	0.53	1418.1741	1419.2578	0.07	1460.9109	1463.7977	0.19
9	1440.2397	1442.3915	0.14	1549.0449	1546.8758	-0.14	1832.9351	1818.1486	-0.81
10	1439.8274	1436.7995	-0.21	1818.0987	1817.5471	-0.03	1827.8668	1835.3138	0.40

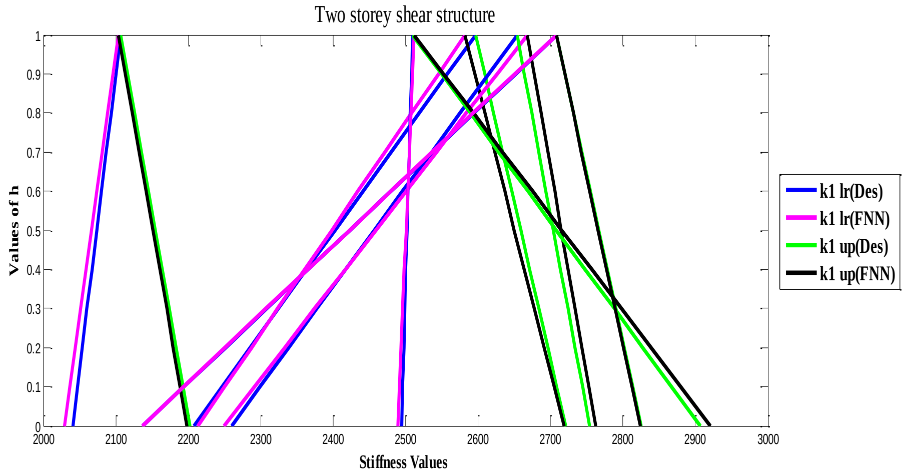

Figure 4.13(a): TFN for forced vibration with crisp forcing function for $[\underline{k}_1]_h, [\overline{k}_1]_h$

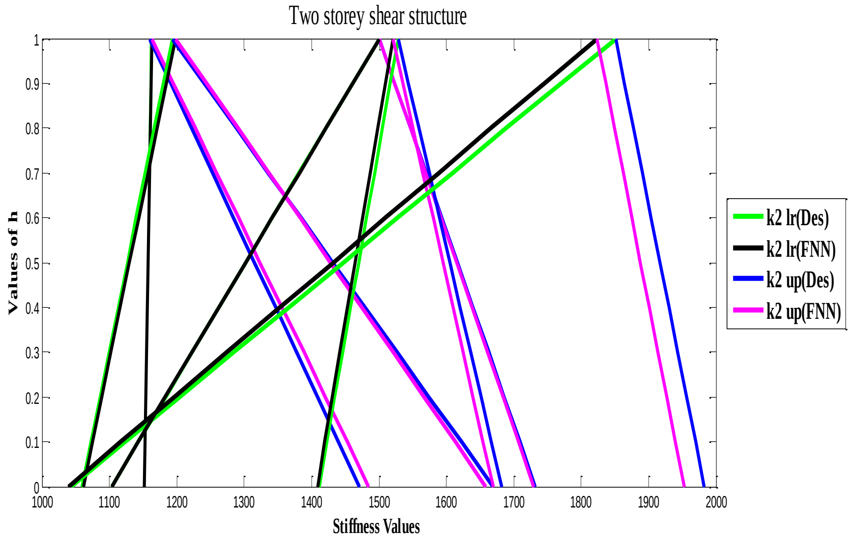

Figure 4.13(b): TFN for forced vibration with crisp forcing function for $[\underline{k}_2]_h, [\overline{k}_2]_h$

Table 4.10(a): Comparison of desired and FNN value for forced vibration with crisp forcing function for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2614.794	2600.9378	-0.53	2657.854	2651.6512	-0.23	2725.9658	2718.5949	-0.27
2	2237.7572	2235.4871	-0.10	2996.1695	3041.3797	1.48	3167.844	3172.9746	0.16
3	2809.4457	2848.9915	1.38	2982.6746	3013.3644	1.01	3158.9261	3204.0637	1.40
4	2278.5684	2270.4712	-0.35	2398.4785	2387.1684	-0.47	3118.8233	3000.3306	-3.94
5	2318.154	2326.2163	0.34	2597.9402	2609.2578	0.43	2647.5255	2653.7977	0.23
6	2609.0385	2592.4097	-0.64	2646.3188	2632.3915	-0.52	2751.5342	2736.8758	-0.53
7	2403.9125	2418.1184	0.58	2623.576	2626.7995	0.12	2998.4058	3025.3138	0.88
8	2281.515	2283.4698	0.08	2710.6778	2709.6876	-0.03	2865.1965	2866.8599	0.05
9	2336.7559	2333.171	-0.15	2564.2437	2559.4934	-0.18	2845.6772	2837.9734	-0.27
10	2370.5161	2373.3886	0.12	2495.5118	2501.9841	0.25	3116.6444	3171.6305	1.73

Table 4.10(b): Comparison of desired and FNN value for forced vibration with crisp forcing function for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1661.3052	1681.1998	1.18	1812.5259	1811.0988	-0.07	2026.268	2020.332	-0.29
2	1152.4887	1152.677	0.01	1778.5541	1776.3389	-0.12	2002.2421	2001.8665	-0.01
3	1676.4244	1649.1265	-1.65	1789.8393	1789.1753	-0.03	1836.5096	1837.8581	0.07
4	1302.1777	1310.4333	0.62	1367.3487	1369.1194	0.12	1807.2617	1808.1055	0.04
5	1496.1721	1488.9165	-0.48	1524.9724	1522.8356	-0.14	1773.5072	1776.5279	0.17
6	1281.5658	1288.1325	0.50	1583.2043	1580.7259	-0.15	1647.7504	1647.8709	0.00
7	1242.0437	1238.0144	-0.32	2016.0422	2042.737	1.30	2088.4795	2101.638	0.62
8	1277.2321	1276.405	-0.06	1518.6201	1517.7441	-0.05	2100.8337	2109.0804	0.39
9	1278.9354	1281.1211	0.17	2042.633	1975.5228	-3.39	2109.0231	2083.0525	-1.24
10	1146.158	1142.6008	-0.31	1403.0615	1401.4549	-0.11	1765.6606	1764.7645	-0.05

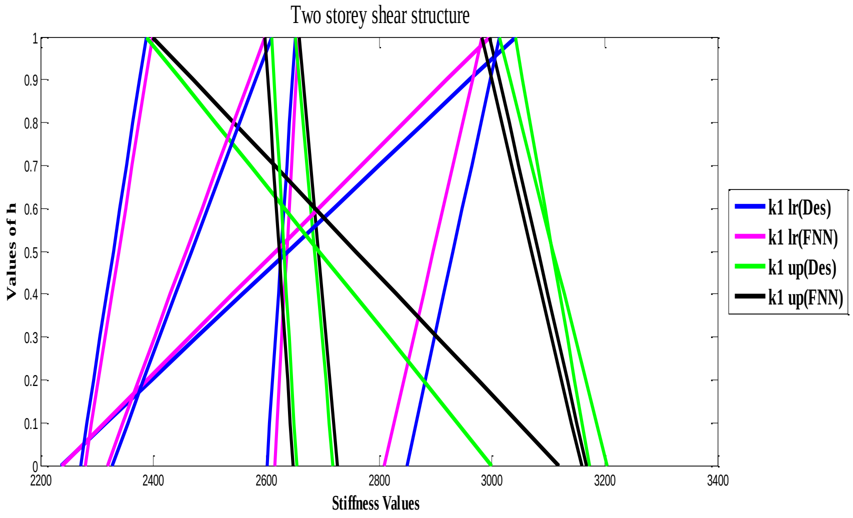

Figure 4.14(a): TFN for forced vibration with crisp forcing function for $[\underline{k}_1]_h, [\overline{k}_1]_h$

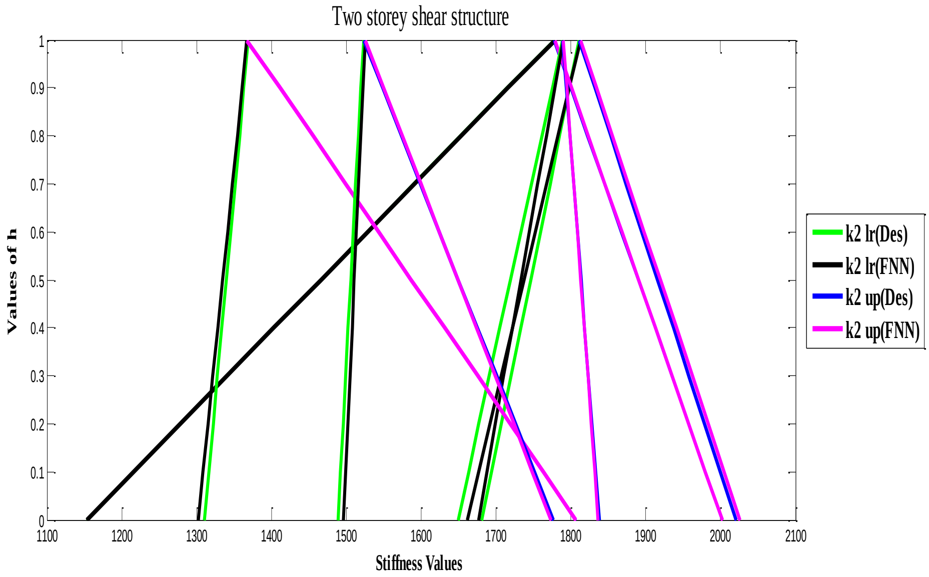

Figure 4.14(b): TFN for forced vibration with crisp forcing function for $[\underline{k}_2]_h, [\overline{k}_2]_h$

Examples for case (iv):

Next, in case (iv), the forcing function vector in fuzzy form with zero initial condition is defined as.

$$\hat{F}(t) = \begin{Bmatrix} (90\sin(1.6\pi t+\pi), 100\sin(1.6\pi t+\pi), 110\sin(1.6\pi t+\pi)) \\ (90\sin(1.6\pi t), 100\sin(1.6\pi t), 110\sin(1.6\pi t)) \end{Bmatrix}^T$$

Again two problems have been solved considering the masses and stiffnesses as in the previous cases. Here 100 data sets are trained with 15 hidden nodes in a single hidden layer so as to get an accuracy of 0.001. Comparison between the desired and FNN values for 10 data among 100 sets of data have been incorporated in Tables 4.11(a) and 4.11(b) for first problem. These tables are plotted as TFN in Figures 4.15(a) and 4.15(b) for 5 data. Similarly, results of second problem have been tabulated in Tables 4.12(a) and 4.12(b) and are plotted as TFN in Figures 4.16(a) and 4.16(b) again for 5 data.

Table 4.11(a): Comparison of desired and FNN value for forced vibration with fuzzy forcing function for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2115.7122	2117.4929	0.08	2772.6421	2791.9339	0.69	2899.2965	2876.9303	-0.77
2	2089.2616	2096.2345	0.33	2226.3707	2225.6745	-0.03	2630.1728	2640.7179	0.39
3	2325.1569	2328.8142	0.15	2380.6635	2376.4366	-0.17	2401.1587	2408.2716	0.29
4	2393.9273	2379.1988	-0.61	2572.4426	2571.7785	-0.02	2659.2466	2653.812	-0.20
5	2252.9072	2248.9533	-0.17	2699.8276	2695.0285	-0.17	2758.2341	2749.1315	-0.33
6	2577.9237	2583.1857	0.20	2612.4951	2607.9416	-0.17	2659.2828	2661.9406	0.09
7	2485.4321	2504.4804	0.76	2714.0388	2740.0323	0.94	2800.4727	2799.3639	-0.03
8	2170.7521	2171.8455	0.05	2230.0568	2234.8269	0.21	2392.2718	2377.6529	-0.61
9	2216.1262	2216.0279	-0.00	2752.5605	2734.9575	-0.64	2801.4019	2801.932	0.01
10	2097.3132	2096.6665	-0.03	2119.1431	2120.6063	0.06	2952.4424	2970.5985	0.61

Table 4.11(b): Comparison of desired and FNN value for forced vibration with fuzzy forcing function for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1295.308	1294.0663	-0.09	1798.6636	1817.8303	1.05	1934.1169	1974.4227	2.04
2	1239.5249	1237.373	-0.17	1428.6221	1442.485	0.96	1511.3849	1507.6038	-0.25
3	1502.7485	1530.8723	1.83	1759.7085	1704.7522	-3.22	1782.7926	1788.9583	0.34
4	1088.5777	1091.4987	0.26	1435.6669	1416.0067	-1.38	1744.2078	1768.0993	1.35
5	1293.5141	1292.9388	-0.04	1404.6348	1405.3154	0.04	1443.3676	1442.6423	-0.05
6	1058.1286	1057.2346	-0.08	1101.3682	1104.8462	0.31	1693.6749	1665.498	-1.69
7	1112.7571	1112.284	-0.04	1134.9847	1119.7551	-1.36	1648.9866	1693.2949	2.61
8	1460.6378	1449.5645	-0.76	1761.8366	1784.4279	1.26	1906.7613	1943.7598	1.90
9	1195.3196	1197.4608	0.17	1277.2241	1291.5703	1.11	1471.6221	1471.7392	0.00
10	1275.8824	1276.1788	0.02	1598.1384	1603.5334	0.33	1638.0726	1629.8572	-0.50

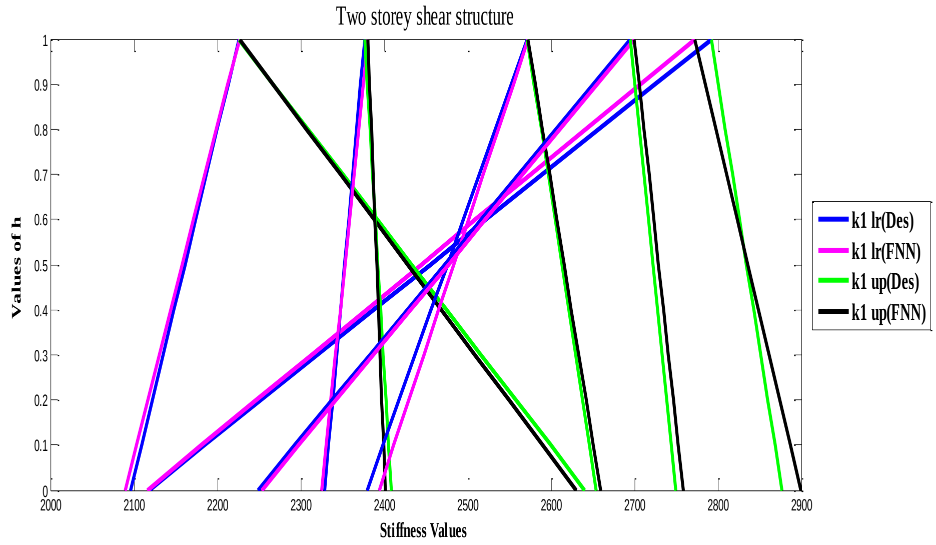

Figure 4.15(a): TFN for forced vibration with fuzzy forcing function for $[\underline{k}_1]_h, [\overline{k}_1]_h$

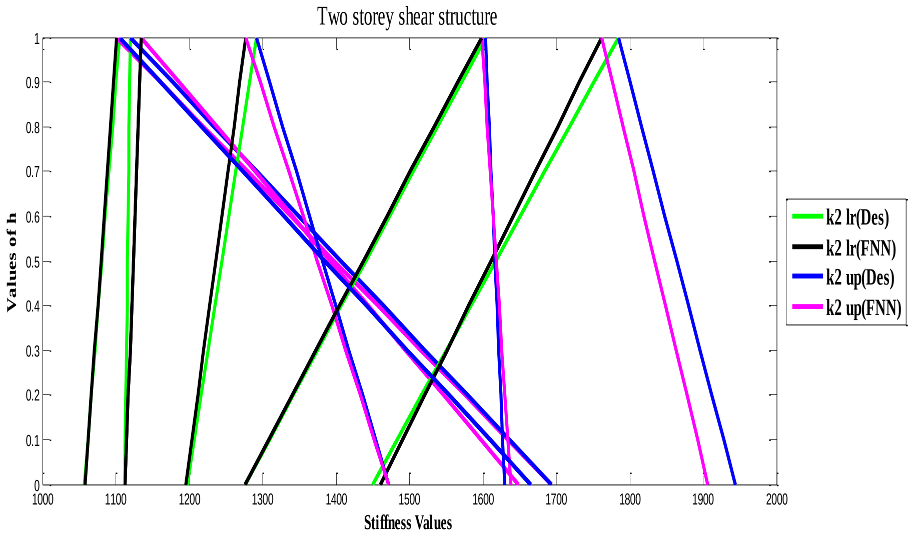

Figure 4.15(b): TFN for forced vibration with fuzzy forcing function for $[\underline{k}_2]_h, [\overline{k}_2]_h$

Table 4.12(a): Comparison of desired and FNN value for forced vibration with fuzzy forcing function for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2343.7788	2336.5531	-0.30	2925.0914	2908.7458	-0.56	3020.7684	3025.4894	0.15
2	2465.6244	2407.8098	-2.40	2784.1906	2796.7215	0.44	2934.6894	2921.2275	-0.46
3	2235.6757	2240.5409	0.21	2320.6483	2306.7619	-0.60	2415.9846	2402.9225	-0.54
4	2480.8046	2459.932	-0.84	2838.5157	2853.7573	0.53	2951.9008	2954.0743	0.07
5	2704.8557	2694.1739	-0.39	2715.0265	2710.0224	-0.18	3096.5705	3106.5119	0.32
6	2297.2588	2248.6742	-2.16	2678.9873	2689.9221	0.40	2985.5553	2979.0517	-0.21
7	2622.3973	2709.9014	3.22	2924.5746	2915.0371	-0.32	3098.2496	3114.7222	0.52
8	2386.5461	2387.9271	0.05	2832.6454	2819.8666	-0.45	3077.5387	3103.7206	0.84
9	2805.8481	2827.6664	0.77	3070.2154	3090.9225	0.66	3188.7561	3198.6806	0.31
10	2522.7177	2534.1631	0.45	2943.0726	2932.6945	-0.35	3115.8346	3069.4423	-1.51

Table 4.12(b): Comparison of desired and FNN value for forced vibration with fuzzy forcing function for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1215.656	1203.4698	-1.01	1590.7598	1600.4716	0.60	1814.1133	1832.4396	1.00
2	1267.9885	1253.171	-1.18	1265.0985	1259.8654	-0.41	1570.3198	1571.0884	0.04
3	1185.62	1159.6189	-2.24	1283.0525	1293.3886	0.79	1760.2019	1769.6053	0.53
4	1467.1961	1510.9378	2.89	1644.8809	1628.5949	-1.00	1803.5189	1781.9719	-1.20
5	1146.6204	1142.4311	-0.36	1932.7711	1951.3797	0.95	2052.817	2082.9746	1.44
6	1178.2138	1171.4455	-0.57	1778.4556	1758.9915	-1.10	1936.4008	1923.3644	-0.67
7	1189.0004	1180.4712	-0.72	1617.8357	1621.6498	0.23	1920.1467	1910.3306	-0.51
8	1198.9945	1196.73	-0.18	1518.4555	1519.2578	0.05	1565.9976	1563.7977	-0.14
9	1532.9409	1542.3915	0.61	1645.0711	1646.8758	0.10	1916.8517	1918.1486	0.06
10	1537.5921	1536.7995	-0.05	1897.5134	1917.5471	1.04	1929.6035	1935.3138	0.29

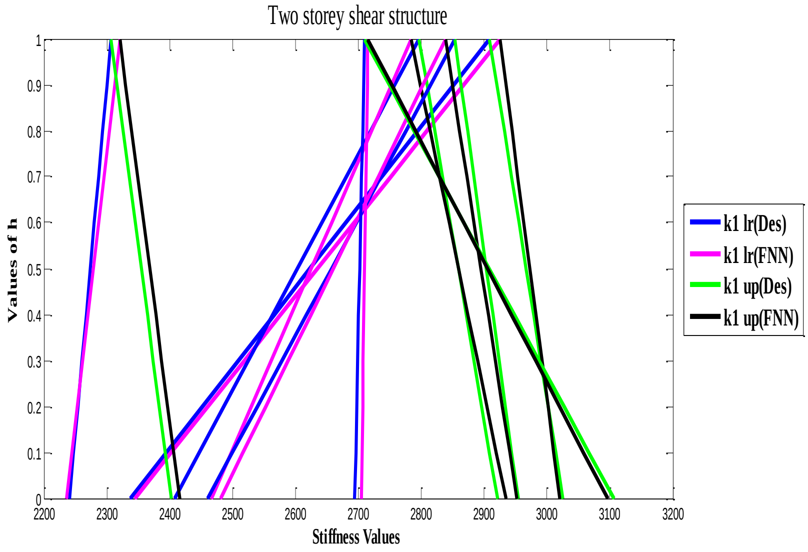

Figure 4.16(a): TFN for forced vibration with fuzzy forcing function for $[\underline{k}_1]_h, [\overline{k}_1]_h$

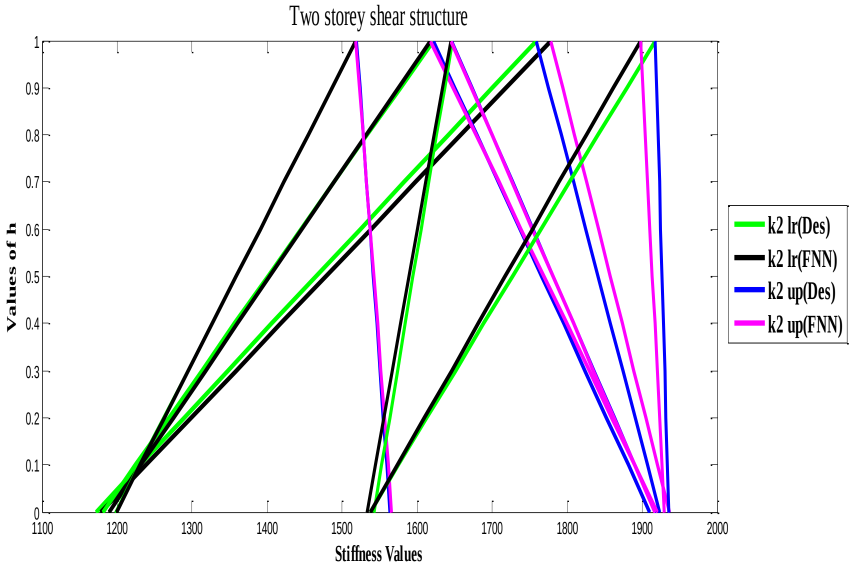

Figure 4.16(b): TFN for forced vibration with fuzzy forcing function for $[\underline{k}_2]_h, [\overline{k}_2]_h$

Examples for case (v) (Testing case):

Finally, in case (v), two examples of a two storey shear structure for comparing the testing values (which are not used in the training) of desired and FNN values in fuzzified form are considered for both ambient vibration and Forced vibration. These test values are fed into the neural network along with the stored (converged) weights to generate corresponding stiffness parameters. The neural network architecture as well as the training pattern used for the first problem were that of case (ii) and for the second problem the training pattern are similar to that of case (iv). Comparison between the test values of desired and FNN for ambient vibration with the initial condition in fuzzified form for 10 data are incorporated in Tables 4.13(a) and 4.13(b). Again comparison between the test values of desired and FNN for forced vibration with the forcing function in fuzzified form have been given in Tables 4.14(a) and 4.14(b).

It may be seen that the neural results are comparable with the desired ones and the deviations in percentage between them have also been shown in all the tables.

Table 4.13(a): Comparison of desired and FNN value of testing data for ambient vibration with fuzzy initial condition for $\underline{k}_1, k_1c, \overline{k}_1$.

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2174.7997	2167.6131	-0.33	2680.7162	2675.7407	-0.18	2859.3327	2814.7237	-1.58
2	2050.3589	2055.7117	0.26	2923.9713	2905.7919	-0.62	2972.4337	2980.5928	0.27
3	2142.4909	2126.9868	-0.72	2864.1349	2869.1293	0.17	2931.6114	2967.1669	1.19
4	2468.1327	2495.3756	1.09	2898.7087	2913.3759	0.50	2928.584	2953.9932	0.86
5	2651.3273	2632.3592	-0.72	2700.3598	2698.7352	-0.06	2810.0473	2810.2805	0.00
6	2089.8352	2097.5404	0.36	2148.3035	2151.8863	0.16	2754.2708	2777.7401	0.84
7	2278.3204	2278.4982	0.00	2434.9387	2431.7613	-0.13	2793.5458	2763.1325	-1.10
8	2422.5316	2412.227	-0.42	2553.7105	2546.8815	-0.26	2909.9751	2925.7355	0.53
9	2639.485	2675.4779	1.34	2799.262	2802.2073	0.10	2924.7594	2957.5068	1.10
10	2207.0929	2191.1867	-0.72	2942.6106	2964.8885	0.75	2972.4436	2969.4924	-0.09

Table 4.13(a): Comparison of desired and FNN value of testing data for ambient vibration with fuzzy initial condition for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k_2}$ (FNN)	$\underline{k_2}$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k_2}$ (FNN)	$\overline{k_2}$ (Des)	Deviation %
1	1312.4152	1296.0251	-1.26	1442.3685	1448.7444	0.44	1701.8707	1706.0461	0.24
2	1040.7478	1031.8328	-0.86	1409.5532	1391.5585	-1.29	1687.0277	1699.7027	0.74
3	1276.1957	1276.923	0.05	1654.1886	1675.098	1.24	1729.4283	1775.5168	2.59
4	1050.3947	1046.1714	-0.40	1200.8051	1182.6117	-1.53	1790.4334	1805.1999	0.81
5	1100.0156	1097.1318	-0.26	1149.9579	1138.9977	-0.96	1205.7938	1196.8726	-0.74
6	1473.3246	1499.7644	1.76	1538.594	1518.3641	-1.33	1811.8321	1823.4578	0.63
7	1482.1069	1455.5862	-1.82	1673.2063	1694.8286	1.27	1925.0262	1979.744	2.76
8	1317.9178	1317.0995	-0.06	1363.3098	1360.3857	-0.21	1669.3333	1656.313	-0.78
9	1578.2463	1605.2678	1.68	1715.7618	1719.3648	0.20	1923.608	1950.222	1.36
10	1034.8191	1034.4461	-0.03	1239.7345	1243.8119	0.32	1774.8388	1764.6867	-0.57

Table 4.14(a): Comparison of desired and FNN value of testing data for forced vibration with fuzzy forcing function for $\underline{k}_1, k_1c, \overline{k}_1$

Data No.	$\underline{k}_1$ (FNN)	$\underline{k}_1$ (Des)	Deviation %	k_1c (FNN)	k_1c (Des)	Deviation %	$\overline{k}_1$ (FNN)	$\overline{k}_1$ (Des)	Deviation %
1	2606.5245	2610.6291	0.15	2903.914	2890.1784	-0.47	2930.4785	2939.5701	0.30
2	2775.2165	2737.8468	-1.36	3151.659	3184.3494	1.02	3184.1227	3216.1561	0.99
3	2599.8702	2574.5343	-0.98	2636.5963	2621.5935	-0.57	3111.6694	3145.5792	1.07
4	2726.6371	2712.6382	-0.51	2887.6096	2876.6447	-0.38	3175.5783	3191.2588	0.49
5	2581.7431	2566.4488	-0.59	2944.0919	2960.5201	0.55	3211.6556	3188.3023	-0.73
6	2729.6716	2793.5332	2.28	2984.8221	2966.8314	-0.60	3148.7835	3106.5439	-1.35
7	2571.8335	2536.6993	-1.38	2689.7628	2674.6949	-0.56	2745.6539	2761.7925	0.58
8	2651.0194	2633.4273	-0.66	2767.9257	2793.5706	0.91	2888.2044	2862.3819	-0.90
9	2437.3052	2437.7321	0.01	2437.8596	2444.1653	0.25	2727.487	2721.8199	-0.20
10	2283.5792	2292.5927	0.39	2347.4621	2345.6546	-0.07	2497.0013	2495.5073	-0.05

Table 4.14(b): Comparison of desired and FNN value of testing data for forced vibration with fuzzy forcing function for $\underline{k}_2, k_2c, \overline{k}_2$

Data No.	$\underline{k}_2$ (FNN)	$\underline{k}_2$ (Des)	Deviation %	k_2c (FNN)	k_2c (Des)	Deviation %	$\overline{k}_2$ (FNN)	$\overline{k}_2$ (Des)	Deviation %
1	1407.5404	1408.9146	0.09	1820.2395	1814.3396	-0.32	1889.6371	1886.9217	-0.14
2	1810.2488	1826.1044	0.86	1822.0224	1839.513	0.95	2023.1157	2054.4783	1.52
3	1224.4351	1227.8889	0.28	1333.8205	1334.2771	0.03	1868.3794	1882.8721	0.76
4	1305.7017	1302.2275	-0.26	1384.7555	1379.0547	-0.41	1800.5751	1793.7876	-0.37
5	1106.6943	1109.8023	0.28	1219.0247	1219.0953	0.00	1779.6545	1783.0312	0.18
6	1593.7799	1587.4922	-0.39	1623.4222	1609.7638	-0.84	1969.8186	1943.2133	-1.36
7	1307.7088	1313.2453	0.42	1751.4163	1733.7164	-1.02	2010.2226	2022.332	0.59
8	1346.9579	1346.4449	-0.03	1799.8936	1800.9542	0.05	2014.6625	2015.8916	0.06
9	1144.6688	1142.6599	-0.17	1213.3021	1219.0896	0.47	1280.5553	1287.1238	0.51
10	1165.6446	1164.1656	-0.12	1475.931	1478.1861	0.15	1964.6192	1939.6434	-1.28

4.6 Conclusion

Protection of various structures against the effect of earthquake is an interdisciplinary research where the knowledge, skills and experience of earthquake along with structural engineers assisted by architects, art historians, material scientists and applied mathematicians are required. Health monitoring, system identification, theoretical and experimental assessment of structural performance, design, testing and implementation of retrofit are some of the main steps of any modern earthquake protection methodology for conservation of structures. As such after a long span of time, structures deteriorate due to application of various manmade and natural hazards such as earthquake etc. So, it is a challenging task to know the present health of the above structures to avoid failure. Hence the present study demonstrates application of fuzzy neural network (FNN) with solution of forward vibration problem for the identification of uncertain structural parameters of multi-storey shear buildings utilizing uncertain design parameters of the system by a proposed FNN methodology. Here forward problem for each time step is solved for a given input to the system, rather than solving the inverse vibration problem. Thus the solution vector is generated. The initial uncertain design parameters (viz. stiffness, mass) are known and so the initial responses of the said problem are also known. The initial values of the physical parameters (in uncertain viz. fuzzy form) of the system are used to obtain the responses in fuzzified form. Responses and the corresponding parameters are used as the input/output in the neural net. It is assumed that only the stiffness is changed and the mass remains the same. The present values of the responses may be obtained by available experiments and using these one may get the present parameter values by FNN. Although to train the new FNN model, set of data are generated numerically beforehand. As such converged FNN model gives the present stiffness parameter values in fuzzy form for each storey. Thus one may predict the health of the structure from the knowledge of the identified stiffness parameters in fuzzified form. Corresponding example problems have been solved and related results are reported to show the reliability and powerfulness of the model.

Chapter 5

System Identification through Seismic Data Using Interval and Fuzzy Neural Network

Earthquakes are one of the most destructive natural phenomena which consist of rapid vibrations of rock near the earth's surface. Usually the earthquake acceleration is noted from the equipment in crisp or exact form. But in actual practice those data may not be obtained exactly at each time step, rather those may be with some error. So those records at each time step are assumed here as interval and fuzzy. Using those interval and fuzzy acceleration data, the structural responses are found. The primary aim of the present study is to model Interval and Fuzzy Neural Network and to estimate structural response of single and multi-storey shear buildings by training the model for Indian earthquakes at Chamoli and Uttarkashi using interval and fuzzified ground motion data. The interval neural network is first trained for a real earthquake data viz. the ground acceleration as input and the numerically generated responses of different floors of multi-storey buildings as output. Till date no model exists to handle positive and negative data in the INN and FNN. As such here the bipolar data in [-1, 1] are converted first to unipolar form that is to [0, 1] by means of a novel transformation for the first time to handle the above training patterns in normalized form. Once the training is done, again the unipolar data are converted back to its bipolar form by using the inverse transformation. The trained INN and FNN architecture is then used to simulate and test the structural response of different floors for various intensity earthquake data and it is found that the predicted responses given by INN and FNN model are good for practical purposes.

5.1 Response Analysis for Single Degree of Freedom System Subject to Ground Motion for Interval Case

Basic concept behind the proposed methodology is to predict the structural response of uncertain (intervals) shear structural system subjected to uncertain earthquake forces. Two scenarios viz without damping and with damping have been considered for the analysis.

5.1.1 Without Damping

Here we will discuss the procedure for an example problem of Single Degree of Freedom (SDOF) systems. Let $\tilde{M}$ be the mass of the generalized one storey structure, $\tilde{K}$ the stiffness of the structure and $\tilde{X}$ be the displacement relative to the ground all in interval form. Then the equation of motion may be written as

$$\tilde{M}\tilde{\ddot{X}} + \tilde{K}\tilde{X} = -\tilde{M}\tilde{\ddot{a}} \qquad (5.1)$$

$\tilde{\ddot{X}}$ = Acceleration in interval,

$\tilde{X}$ = Displacement in interval,

$\tilde{\ddot{a}}$ = Ground acceleration in interval.

Equation (5.1) may be written as,

$$\tilde{\ddot{X}} + \tilde{\omega}^2\tilde{X} = -\tilde{\ddot{a}} \qquad (5.2)$$

where $\tilde{\omega}^2$ is the interval natural frequency parameter of the undamped structure. It may be noted that the above equation can be solved by Interval Duhamel integral. Here to obtain the solution for equation (5.2), the Duhamel integral are considered for different sets of lower and upper form. This is done to avoid complicacy raised while getting the above solution. And that is why now we will drop '~' from all notations and will consider the case for lower form first and similarly for upper form. Hence Equation (5.2) in lower form is written as,

$$\underline{\ddot{X}} + \underline{\omega}^2\underline{X} = -\underline{\ddot{a}} \qquad (5.3)$$

where $\underline{\omega}^2 = \frac{\underline{K}}{\underline{M}}$ is the natural frequency parameter of the undamped structure in lower form. $\underline{K}$ is the stiffness parameter and $\underline{M}$ is the mass of the storey in lower form.

$\underline{\ddot{X}}$ = Acceleration in lower form,

$\underline{X}$ = Displacement in lower form,

$\underline{\ddot{a}}$ = Ground acceleration in lower form.

Hence the solution of equation (5.3), Newmark and Rosenblueth [201] in lower form is written as

$$\underline{X(t)} = -\frac{1}{\underline{\omega}}\int_0^t \underline{\ddot{a}}(\tau)\sin[\underline{\omega}(t-\tau)]d\tau \qquad (5.4)$$

From this solution the response of the structure namely acceleration in lower form is obtained for no damping. In a similar fashion we can compute for upper form.

5.1.2 With Damping

Let $\tilde{M}$ be the mass of the generalized one storey structure in interval form, $\tilde{K}$ the stiffness of the structure in interval form, $\tilde{C}$ the damping and $\tilde{X}$ be the displacement relative to the ground all are in interval form. Then the equation of motion may be written as

$$\tilde{M}\ddot{\tilde{X}} + \tilde{C}\dot{\tilde{X}} + \tilde{K}\tilde{X} = -\tilde{M}\tilde{\ddot{a}} \tag{5.5}$$

Equation (5.5) may be written in lower form as,

$$\underline{\ddot{X}} + 2\underline{\xi\omega}\,\underline{\dot{X}} + \underline{\omega}^2\,\underline{X} = -\underline{\ddot{a}} \tag{5.6}$$

where $\underline{\xi\omega} = \frac{C}{\underline{M}}$ and $\underline{\omega}^2 = \frac{K}{\underline{M}}$ are the natural frequency parameter of the damped and undamped structure in lower form. $\underline{C}$ is the damping and $\underline{\dot{X}}$ is the velocity in lower form.

Here also solution is obtained for different sets in lower and upper form. Hence the solution of equation (5.6) in lower form is given as

$$\underline{X(t)} = -\frac{1}{\underline{\omega}}\int_0^t \underline{\ddot{a}}(\tau)\exp\left[-\underline{\xi}\,\underline{\omega}(t-\tau)\right]\sin\left[\underline{\omega}(t-\tau)\right]d\tau \tag{5.7}$$

From this solution the response of the structure namely the acceleration in lower form with damping is obtained. In a similar manner we can compute for the upper form. Hence, the neural network architecture is constructed by taking the interval form of ground acceleration as input and the responses obtained from the above solution as output for each time step which is also in interval form.

5.2 Interval Neural Network Model with Single Input and Output Node

The interval neural network model is considered with single input and single output node. The interval neural network model with single input and single output node is given in Figure 5.1. In Figure 5.1, $\tilde{O}_i$, $\tilde{O}_j$ and $\tilde{O}_k$ are taken as input, hidden and output layers. The inputs $\tilde{O}_i = \tilde{\ddot{a}}_i = \left[\underline{\ddot{a}}_i, \overline{\ddot{a}}_i\right]$ are the ground acceleration in interval and the outputs $\tilde{O}_k = \tilde{\ddot{x}}_k = \left[\underline{\ddot{x}}_k, \overline{\ddot{x}}_k\right]$ are responses of the structure in interval form. The training and learning algorithm have already been discussed in Sec. 2.7 and 3.2. The weights between input and hidden layers are denoted by $\tilde{v}_{ji} = \left[\underline{v}_{ji}, \overline{v}_{ji}\right]$ and the weights between hidden and output layers are denoted by $\tilde{w}_{kj} = \left[\underline{w}_{kj}, \overline{w}_{kj}\right]$ which are all in intervals. $\left[\underline{\theta}_j, \overline{\theta}_j\right]$ are the bias weights of the hidden layer and $\left[\underline{\theta}_k, \overline{\theta}_k\right]$ are the bias weights of the output layer.

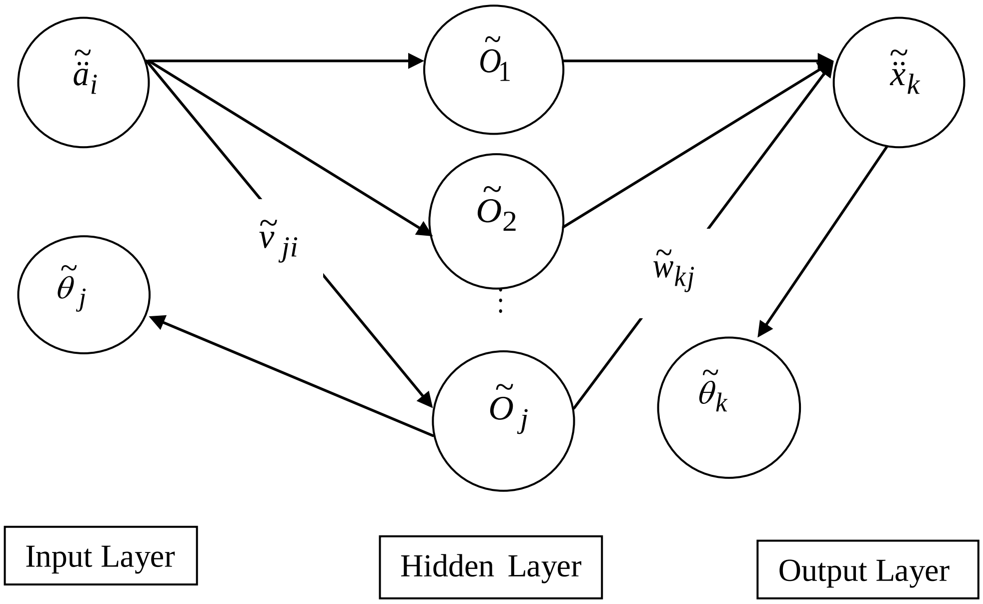

Figure 5.1: Interval Neural Network Model with Single Input and Output Node

Next we discuss the fuzzified form of the above neural network.

5.3 Response Analysis for Single Degree of Freedom System Subject to Ground Motion for Fuzzy Case

The basic concept behind the proposed methodology is to predict the structural response (in fuzzified form) of an uncertain shear structural system subject to earthquake forces which are also in fuzzified form. Two scenarios namely, without damping and with damping have been considered for the analysis.

5.3.1 Without Damping

Here we discuss the procedure for an example problem of a Single Degree of Freedom (SDOF) system. Let $\hat{\mathrm{M}}$ be the mass of the generalized one storey structure in fuzzified form, $\hat{\mathrm{K}}$ the stiffness of the structure in fuzzified form and $\hat{\mathrm{X}}$ be the displacement relative to the ground also in fuzzified form. Then the equation of motion may be written as

$$\hat{M}\hat{\ddot{X}} + \hat{K}\hat{X} = -\hat{M}\hat{\ddot{a}} \qquad (5.8)$$

$\hat{\ddot{X}}$ = Acceleration in fuzzified form,

$\hat{X}$ = Displacement in fuzzified form,

$\hat{\ddot{a}}$ = Ground acceleration in fuzzified form.

Equation (5.8) may be written as,

$$\hat{\ddot{X}} + \hat{\omega}^2\hat{X} = -\hat{\ddot{a}} \qquad (5.9)$$

where $\hat{\omega}^2$ is the fuzzified natural frequency parameter of the undamped structure. It may be noted that the above equation can be solved by Fuzzy Duhamel integral. Here to obtain the solution for equation (5.9), the Duhamel integral are considered for different sets of lower, centre and upper form. This is done to avoid complicacy raised while getting the above solution. And that is why now we will drop '^' from all notations and will consider the case for lower form first and similarly for centre and upper form.

Hence Equation (5.9) in lower form is written as,

$$\underline{\ddot{X}} + \underline{\omega}^2\underline{X} = -\underline{\ddot{a}} \qquad (5.10)$$

where $\underline{\omega}^2 = \frac{\underline{K}}{\underline{M}}$ is the natural frequency parameter of the undamped structure in lower form. $\underline{K}$ is the stiffness parameter and $\underline{M}$ is the mass of the storey in lower form.

$\underline{\ddot{X}}$ = Acceleration in lower form,

$\underline{X}$ = Displacement in lower form,

$\underline{\ddot{a}}$ = Ground acceleration in lower form.

Hence the solution of equation (5.10), Newmark and Rosenblueth [201] in lower form is written as

$$\underline{X(t)} = -\frac{1}{\underline{\omega}}\int_0^t \underline{\ddot{a}}(\tau)\sin[\underline{\omega}(t-\tau)]d\tau \qquad (5.11)$$

From this solution the response of the structure namely acceleration in lower form is obtained for no damping. In a similar fashion we can compute for centre and upper form.

5.3.2 With Damping

Let $\hat{M}$ be the mass of the generalized one storey structure in fuzzified form, $\hat{K}$ the stiffness of the structure in fuzzified form, $\hat{C}$ the damping and $\hat{X}$ be the displacement relative to the ground all are in fuzzified form. Then the equation of motion may be written as

$$\hat{\mathrm{M}}\hat{\ddot{\mathrm{X}}} + \hat{\mathrm{C}}\hat{\dot{\mathrm{X}}} + \hat{\mathrm{K}}\hat{\mathrm{X}} = -\hat{\mathrm{M}}\hat{\ddot{\mathrm{a}}} \tag{5.12}$$

Equation (5.12) may be written in lower form as,

$$\underline{\ddot{\mathrm{X}}} + 2\underline{\xi\omega}\underline{\dot{\mathrm{X}}} + \underline{\omega}^2\,\underline{\mathrm{X}} = -\underline{\ddot{\mathrm{a}}} \tag{5.13}$$

where $\underline{\xi\omega} = \frac{\underline{C}}{\underline{M}}$ and $\underline{\omega}^2 = \frac{\underline{K}}{\underline{M}}$ are the natural frequency parameter of the damped and undamped structure in lower form. $\underline{C}$ is the damping and $\underline{\dot{X}}$ is the velocity in lower form. Here also solution is obtained for different sets in lower, centre and upper form. Hence the solution of equation (5.13) in lower form is given as

$$\underline{\mathrm{X(t)}} = -\frac{1}{\underline{\omega}}\int_0^{\mathrm{t}} \underline{\ddot{\mathrm{a}}}(\tau)\exp\left[-\underline{\xi}\,\underline{\omega}(\mathrm{t}-\tau)\right]\sin\left[\underline{\omega}(\mathrm{t}-\tau)\right]\mathrm{d}\tau \tag{5.14}$$

From this solution the response of the structure namely the acceleration in lower form with damping is obtained. In a similar manner we can compute for the centre and upper form. Hence, the neural network architecture is constructed by taking the fuzzified form of ground acceleration as input and the responses obtained from the above solution as output for each time step which is also in fuzzified form.

5.4 Transformation of Data from Bipolar to Unipolar for Fuzzy Case

The data are in fuzzified form, hence they are first converted to h-level form and then the transformation is applied. The data are transferred from bipolar to unipolar i.e., from [-1, 1] to [0, 1] by the following transformation. Here the input is considered as the ground acceleration and output as the response of the structure

$$\left[\hat{S}_k\right]_h = \frac{1+\left[\hat{d}_k\right]_h}{2} \tag{5.15}$$

where $\left[\hat{d}_k\right]_h$ is the desired output in h-level form.

5.5 Fuzzy Neural Network Model with Single Input and Output Node

The fuzzy neural network model is considered with single input and single output node. The fuzzy neural network model is given in Figure 5.2. In Figure 5.2, $\hat{O}_i$, $\hat{O}_j$ and $\hat{O}_k$ are taken as input, hidden and output layers. The inputs $\hat{O}_i = \hat{\ddot{a}}_i = \left[\underline{\ddot{a}}_i, \ddot{a}_i c, \overline{\ddot{a}}_i\right]$ are the ground acceleration in interval and the outputs $\hat{O}_k = \hat{\ddot{x}}_k = \left[\underline{\ddot{x}}_k, \ddot{x}_k c, \overline{\ddot{x}}_k\right]$ are responses of the structure in interval form. The training and learning algorithm have already been discussed in Sec. 2.8 and 3.4. The weights between input and hidden layers are denoted by $\hat{v}_{ji} = \left[\underline{v}_{ji}, v_{ji} c, \overline{v}_{ji}\right]$ and the weights between hidden and output layers are denoted by $\hat{w}_{kj} = \left[\underline{w}_{kj}, w_{kj} c, \overline{w}_{kj}\right]$ which are all in intervals. $\hat{\theta}_j = \left[\underline{\theta}_j, \theta_j c, \overline{\theta}_j\right]$ are the bias weights of the hidden layer and $\hat{\theta}_k = \left[\underline{\theta}_k, \theta_k c, \overline{\theta}_k\right]$ are the bias weights of the output layer.

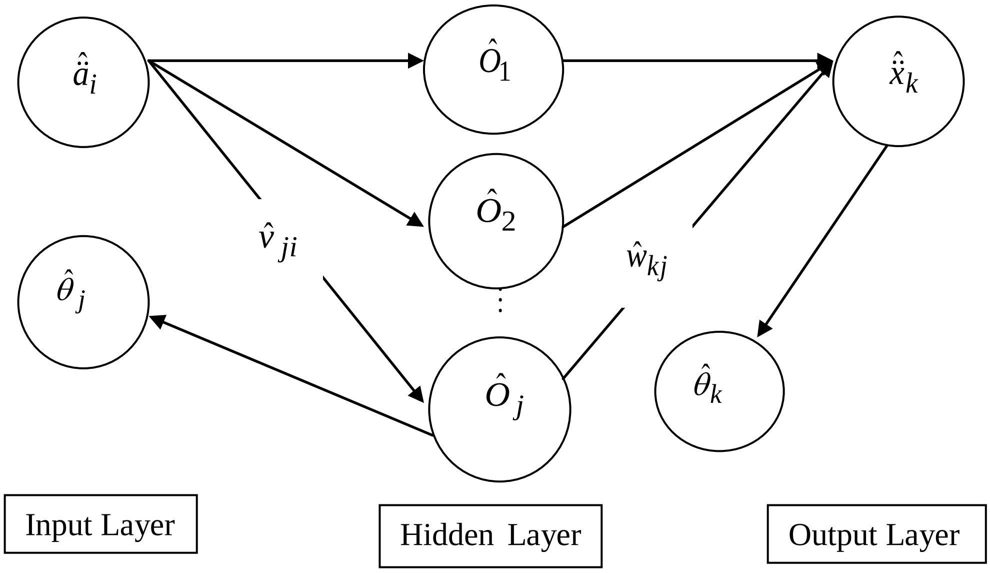

Figure 5.2: Fuzzy Neural Network Model with Single Input and Single Output Node

5.6 Response Analysis for Multi Degree of Freedom System Subject to Ground Motion for Interval Case

The basic concept behind the proposed methodology is to predict the interval structural response of multi degree of freedom system viz. two, six and ten storey shear buildings subject to various earthquake accelerations by training the same for one particular earthquake data. When a shear building with n storey is subjected to base excitation, then the governing equation of motion in interval form may be written as

$$[\tilde{M}]\{\ddot{\tilde{x}}(t)\} + [\tilde{C}]\{\dot{\tilde{x}}(t)\} + [\tilde{K}]\{\tilde{x}(t)\} = -[\tilde{M}]\{\psi\}\ddot{\tilde{x}}_g(t) \tag{5.16}$$

where $\left[\tilde{M}\right] = \left[\underline{M}\,, \overline{M}\right]$ is $n \times n$ interval mass matrix, $\left[\tilde{K}\right] = \left[\underline{K}\,, \overline{K}\right]$ is $n \times n$ interval stiffness matrix of the structure in interval form and $\left[\tilde{C}\right] = \left[\underline{C}\,, \overline{C}\right]$ represents $n \times n$ interval damping matrix, $\{\psi\}$ is the influence co-efficient vector. Here $\{\tilde{x}(t)\}$ is the displacement relative to the ground, $\{\ddot{\tilde{x}}(t)\}$ is the response acceleration, $\{\dot{\tilde{x}}(t)\}$ is the response velocity and $\ddot{\tilde{x}}_g(t)$ is the earthquake ground acceleration all are in interval form. The global mass, stiffness and damping matrices in interval form can be written as $\left[\tilde{M}\right]$, $\left[\tilde{K}\right]$ and $\left[\tilde{C}\right]$ as below:

$$\left[\tilde{M}\right] = \begin{bmatrix} \left[\underline{m}_1, \overline{m}_1\right] & 0 & \cdots & \cdots & 0 \\ 0 & \left[\underline{m}_2\,, \overline{m}_2\right] & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & 0 & \left[\underline{m}_{n-1}\,, \overline{m}_{n-1}\right] & 0 \\ 0 & \cdots & \cdots & 0 & \left[\underline{m}_n\,, \overline{m}_n\right] \end{bmatrix},$$

$$\left[\tilde{K}\right] = \begin{bmatrix} \left[\underline{k}_1, \overline{k}_1\right] + \left[\underline{k}_2\,, \overline{k}_2\right] & -\left[\underline{k}_2\,, \overline{k}_2\right] & 0 & \cdots & 0 \\ -\left[\underline{k}_2\,, \overline{k}_2\right] & \left[\underline{k}_2\,, \overline{k}_2\right] + \left[\underline{k}_3\,, \overline{k}_3\right] & -\left[\underline{k}_3\,, \overline{k}_3\right] & \cdots & 0 \\ \cdots & \cdots & & & \cdots \\ 0 & \cdots & -\left[\underline{k}_{n-1}\,, \overline{k}_{n-1}\right] & \left[\underline{k}_{n-1}\,, \overline{k}_{n-1}\right] + \left[\underline{k}_n\,, \overline{k}_n\right] & -\left[\underline{k}_n\,, \overline{k}_n\right] \\ 0 & & \cdots & -\left[\underline{k}_n\,, \overline{k}_n\right] & \left[\underline{k}_n\,, \overline{k}_n\right] \end{bmatrix}.$$

and

$$\left[\tilde{C}\right] = \begin{bmatrix} \left[\underline{c}_1, \overline{c}_1\right] + \left[\underline{c}_2\,, \overline{c}_2\right] & -\left[\underline{c}_2\,, \overline{c}_2\right] & 0 & \cdots & 0 \\ -\left[\underline{c}_2\,, \overline{c}_2\right] & \left[\underline{c}_2\,, \overline{c}_2\right] + \left[\underline{c}_3\,, \overline{c}_3\right] & -\left[\underline{c}_3\,, \overline{c}_3\right] & \cdots & 0 \\ \cdots & \cdots & & & \cdots \\ 0 & \cdots & -\left[\underline{c}_{n-1}\,, \overline{c}_{n-1}\right] & \left[\underline{c}_{n-1}\,, \overline{c}_{n-1}\right] + \left[\underline{c}_n\,, \overline{c}_n\right] & -\left[\underline{c}_n\,, \overline{c}_n\right] \\ 0 & & \cdots & -\left[\underline{c}_n, \overline{c}_n\right] & \left[\underline{c}_n\,, \overline{c}_n\right] \end{bmatrix}$$

Equation (5.16) is generally a set of n-coupled equations. The classical method of solving these equations in the absence of damping is to find the normal modes of oscillations of the homogeneous equation and to determine the normal co-ordinates.

$$[\tilde{M}]\{\ddot{\tilde{x}}(t)\} + [\tilde{K}]\{\tilde{x}(t)\} = 0 \tag{5.17}$$

Let us take normal co-ordinates in lower form as $\left(\{\underline{x}(t)\} = [\underline{A}]\{\underline{y}(t)\}\right)$. We have considered the lower form to avoid complicacy arised due to the interval computation. Hence we will remove '~' from all notations and denote the lower and upper form separately.

Substituting $\left(\{\underline{x}(t)\} = [\underline{A}]\{\underline{y}(t)\}\right)$ and premultiplying by $[\underline{A}]^T$ in Equation (5.16) we get

$$[\underline{A}]^T [\underline{M}] [\underline{A}] \{\underline{\ddot{y}}(t)\} + [\underline{A}]^T [\underline{C}][\underline{A}] \{\underline{\dot{y}}(t)\} + [\underline{A}]^T [\underline{K}] [\underline{A}] \{\underline{y}(t)\} = -[\underline{A}]^T [\underline{M}][\psi]\underline{\ddot{x}}_g(t) \quad (5.18)$$

The above equation reduces to

$$[\underline{M}_m] \{\underline{\ddot{y}}(t)\} + [\underline{C}_d] \{\underline{\dot{y}}(t)\} + [\underline{K}_d] \{\underline{y}(t)\} = -[\underline{A}]^T [\underline{M}][\psi]\underline{\ddot{x}}_g(t) \quad (5.19)$$

where $[\underline{A}]^T [\underline{M}] [\underline{A}] = [\underline{M}_m]$ is the generalized mass matrix which is diagonal

$[\underline{A}]^T [\underline{K}] [\underline{A}] = [\underline{K}_d]$ is the generalized stiffness matrix which is diagonal

$[\underline{A}]^T [\underline{C}][\underline{A}] = [\underline{C}_d]$ is the damping matrix which is in general not diagonal.

If $[\underline{C}]$ is proportional to $[\underline{M}]$ or $[\underline{K}]$ then $[\underline{A}]^T [\underline{C}][\underline{A}]$ becomes diagonal. Then we can say that the system has proportional damping and Equation (5.19) becomes completely uncoupled. The above differential equations can be solved as single degree of freedom and the solution can be got easily. After getting $\{\underline{y}(t)\}$ from Equation (5.19), the displacement $\underline{x}_i(t)$ of any floor can be obtained by using the equation, $\left(\{\underline{x}(t)\} = [\underline{A}]\{\underline{y}(t)\}\right)$.

For a linear system with proportional damping, Equation (5.16) can be solved by modal analysis technique. But for earthquake response analysis, the modal analysis technique becomes more efficient. Thus, Equation (5.16) can be reduced to *n*-modal equations of the form (Chakraverty, [86] and Chakraverty et al. [89])

$$\underline{\ddot{x}}_r(t) + 2\underline{\xi}_r \tilde{\omega}_r \underline{\dot{x}}_r(t) + \underline{\omega}_r^2 \underline{x}_r(t) = -\underline{\Gamma}_r \underline{\ddot{x}}_g(t) \quad r=1,2,\ldots,n \qquad (5.20)$$

where *n* ($\leq N$) is the number of significant modes, $\underline{\xi}_r$ is the damping ratio and the modal coordinate $\underline{x}_r(t)$ is related to the displacement of the i-th mass as follows:

$$\underline{v}_i = \sum_{r=1}^{n} \underline{\Phi}_{ir} \underline{x}_r \qquad (5.21)$$

in which $\underline{\Phi}_{ir}$ is the i-th component of the r-th mode-shape vector and $\underline{\Gamma}_r$ is the modal participation factor. Equation (5.20) represents the equation of motion of *n* SDOF system and the response is obtained from Duhamel integral. The Duhamel integral in lower form is written as

$$\underline{x}_r(t) = -\frac{\underline{\Gamma}_r}{\underline{\omega}_{Dr}} \int_{\underline{0}}^{t} \ddot{x}_g(\tau) \exp\left[-\underline{\xi}_r \, \underline{\omega}_r (t-\tau)\right] \sin\left[\underline{\omega}_r (t-\tau)\right] d\tau \tag{5.22}$$

Here $\underline{\omega}_{Dr} = \underline{\omega}_r \sqrt{1-\underline{\xi}_r^2}$, where $\underline{\omega}_{Dr}$,$\underline{\omega}_r$ and $\underline{\xi}_r$ are damped frequency, free vibration frequency and damping ratio respectively. The time history response of the *i*-th mass is then determined from Equation (5.20) as

$$\underline{v}_i(t) = \underline{\phi}_{i1}\underline{x}_1(t) + \underline{\phi}_{i2}\underline{x}_2(t) + \cdots$$

From this the floor responses of the multi-storey structures viz. displacement is obtained in lower form. In similar manner we can compute for upper form as well. The interval neural network architecture is constructed, taking ground acceleration in interval form as input and the storey response as output in interval form, obtained from the above solution for each time step. Therefore, the whole network consists of one input layer, one hidden layer with varying nodes and one output layer as shown in Figure 5.3.

5.7 Transformation of Data from [-1, 1] to [0, 1]

Here input is ground acceleration and output is the structural response which may be positive and/or negative. As said above first the normalization is done for both input and output data. Then these normalized data are transformed from [-1, 1] to [0, 1] by means of the following transformation

$$\tilde{S}_k = \frac{1+\tilde{d}'_k}{2} \tag{5.23}$$

where $\tilde{d}'_k$ is the normalized desired output in interval form.

5.8 Learning Algorithm of Interval Neural Network Using Unipolar Activation Function

The architecture of multi-layer interval neural network model is shown in Figure 5.3. The training and learning algorithm have already been discussed in Sec. 2.7. $\tilde{O}_i$, $\tilde{O}_j$ and $\tilde{O}_k$ are taken as input, hidden and output layers. In Figure 5.3, the inputs $\tilde{O}_i = \tilde{\ddot{a}}_i = \left[\underline{\ddot{a}}_i, \overline{\ddot{a}_i}\right]$ are the ground acceleration and the outputs $\tilde{O}_k = \tilde{x}_k = \left[\underline{x}_k, \bar{x}_k\right]$ are responses of the structure both in interval form. $\tilde{d}_k$ is the desired output. The weights between input and hidden layers are denoted by $\tilde{v}_{ji} = \left[\underline{v}_{ji}, \bar{v}_{ji}\right]$ and the weights between hidden and output layers are denoted by $\tilde{w}_{kj} = \left[\underline{w}_{kj}, \overline{w}_{kj}\right]$ which are all in intervals. $\left[\underline{\theta}_j, \overline{\theta}_j\right]$ are the bias weights of the hidden layer and $\left[\underline{\theta}_k, \overline{\theta}_k\right]$ are the bias weights of the output layer.

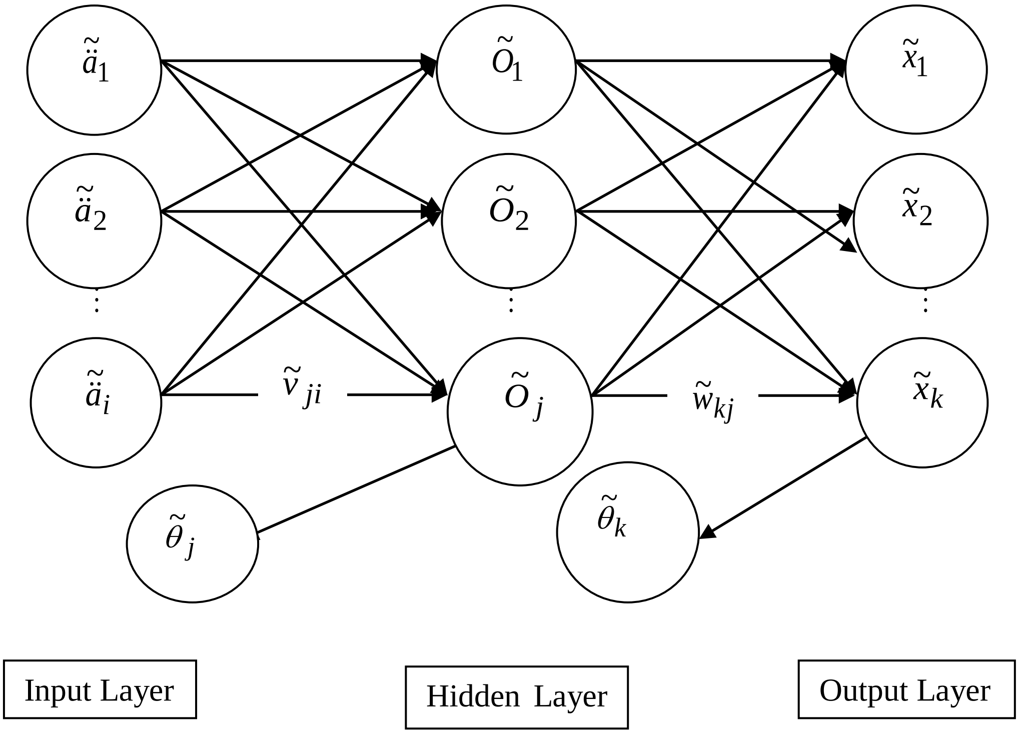

Figure 5.3: Multi-Layer Interval Neural Network Model

5.9 Transformation of Data from [0, 1] to [-1, 1] by Inverse Transformation

Once Training is done the data are again transferred back to its bipolar form i.e., [-1, 1] by using inverse transformation. The product of inverse transformation and the normalization factor gives the INN results in its bipolar form. After back propagation training is completed the final INN output is calculated as

$$\tilde{O}_k = \tilde{R}_k \cdot \tilde{l} \tag{5.24}$$

where $\tilde{R}_k = 2 \cdot \tilde{S}_k - 1$

$\tilde{O}_k$ are the final INN outputs, $\tilde{R}_k$ is the inverse transformation and $\tilde{l}$ is the normalizing factor in interval form.

5.10 Numerical Results and Discussion

5.10.1 Single Degree of Freedom System for Interval Case

For present study two Indian earthquakes viz. the Chamoli earthquake at Barkot in NE (north east) direction and the Uttarkashi earthquake at Barkot in NE (north-east) direction have been considered for training and testing for different cases. Different cases are discussed below:

Case (i) Without damping, Ground acceleration as well as response data in interval form with crisp frequency.

Case (ii) Without damping, Ground acceleration as well as response data in interval form with interval frequency.

Case (iii) With damping, Ground acceleration as well as response data in interval form with crisp frequency and interval damping.

Case (iv) With damping, Ground acceleration as well as response data in interval form with interval frequency and interval damping.

Case (v) Testing for different earthquake data with / without damping.

It is worth mentioning that the earthquake acceleration data are actually both positive and negative. But the present INN cannot handle the data with negative sign due to the complexity in interval computation and also in the INN model. As such we have taken all the earthquake data as absolute (positive) value and then those have been trained. Accordingly, all the plots are presented in the positive y-axis. This is also due to the fact we may concentrate on the amplitude of the acceleration at any instant of time.

Example for case (i):

For case (i), initially the system without damping is studied and for that the system is subjected to Chamoli earthquake with maximum ground acceleration in interval form as [19.088, 20.088] cm/sec/sec at Barkot in NE (north-east) direction was used to compute the response for single storey structure using Eq. (5.4). The obtained response of the structure and the ground acceleration in interval form are trained first for the assumed frequency parameters in crisp form as $\omega=0.5$ with time range 0 to 14.96 secs. (749 data points) for the mentioned earthquake and are shown in Figures 5.4(a) and 5.4(b). Figure 5.4(a) shows the plot for lower values and Figure 5.4(b) depicts the upper values. Similar plots with $\omega=0.02$ are shown in Figures 5.5(a) and 5.5(b). Simulations have been done for different hidden layer nodes and it was seen that the response result is almost same and good for 15 to 18 nodes in the hidden layer. However, 18 hidden layer nodes are used here to generate the results for 749 data points.

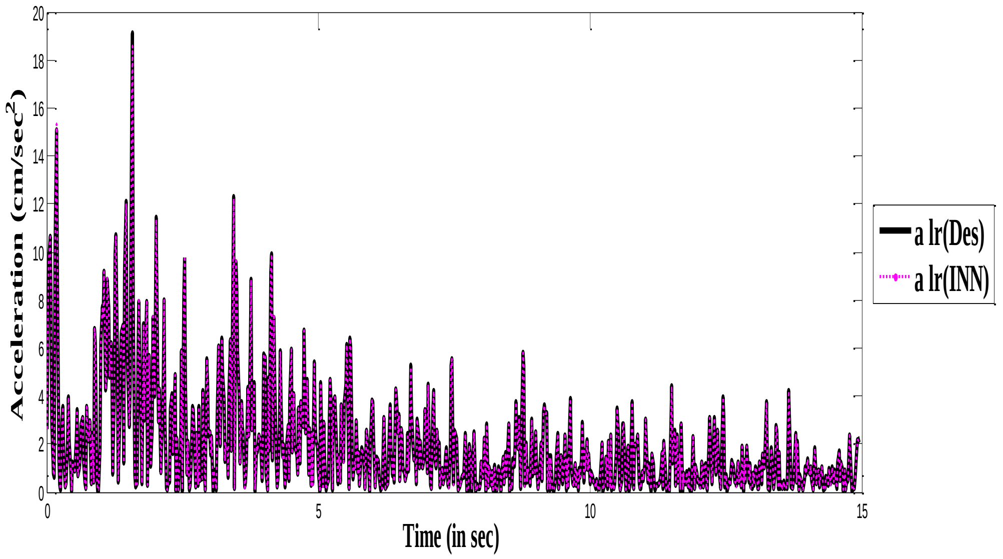

Figure 5.4(a): Comparison between the desired and INN seismic response for lower values (without damping) for Chamoli earthquake at Barkot in NE direction with ω =0.5

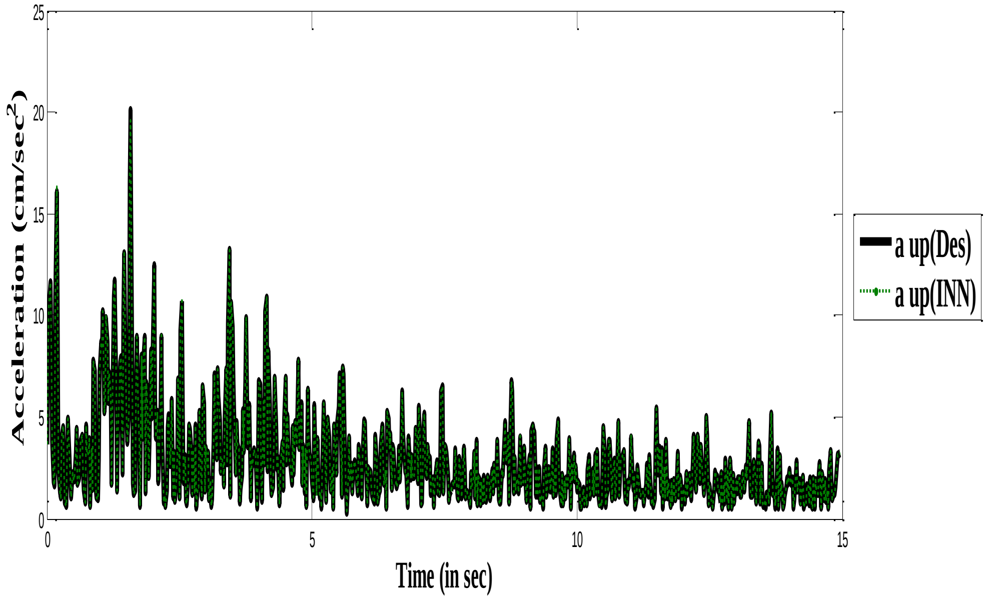

Figure 5.4(b): Comparison between the desired and INN seismic sesponse for upper values (without damping) for Chamoli earthquake at Barkot in NE direction with ω =0.5

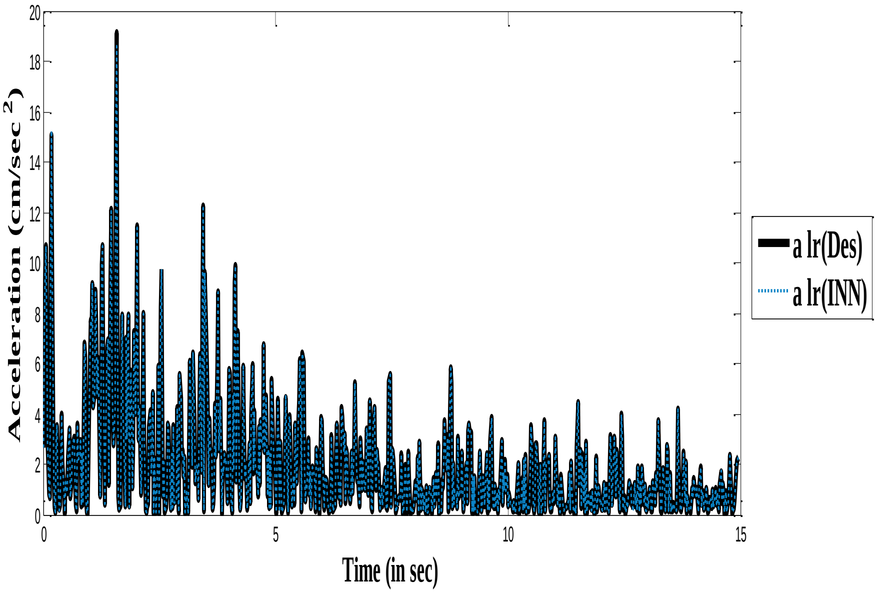

Figure 5.5(a): Comparison between the desired and INN seismic response for lower values (without damping) for Chamoli earthquake at Barkot in NE direction with ω =0.02

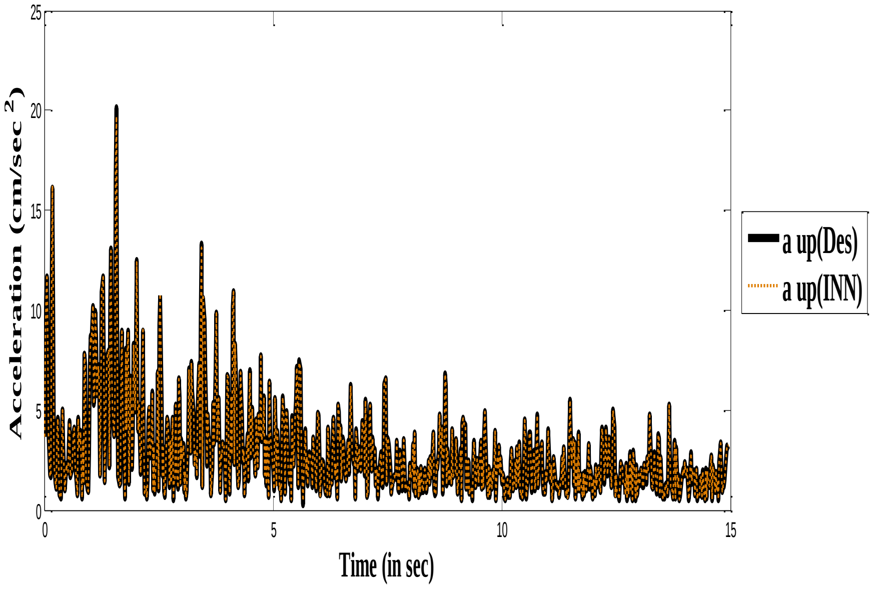

Figure 5.5(b): Comparison between the desired and INN seismic response for upper values (without damping) for Chamoli Earthquake at Barkot in NE direction with ω =0.02

Example for case (ii):

In case (ii), the system is considered without damping with the same earthquake. The earthquake has maximum ground acceleration in interval form as [19.088, 20.088] cm/sec/sec and the response obtained are also in interval form. The neural network architecture is trained within the time range 0 to 14.96 secs. (749 data points) but for two sets of frequency parameters taken in interval form as $\omega = [0.4, 0.6]$ and $\omega = [0.01, 0.03]$. Training has been done for different hidden layer nodes. Again 18 hidden layer nodes are used to generate the results for 749 data points. The results are plotted in Figures 5.6(a), 5.6(b) and 5.7(a), 5.7(b).

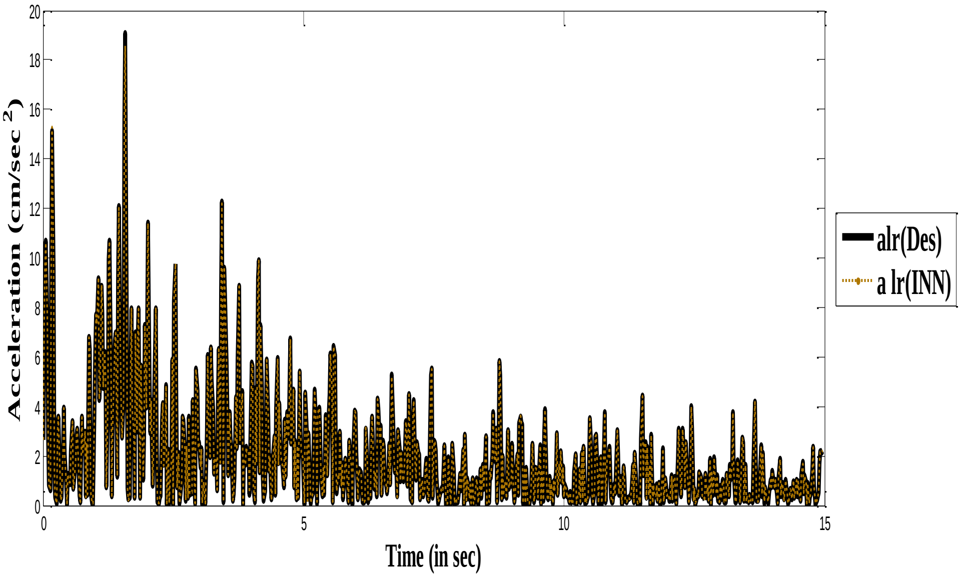

Figure 5.6(a): Comparison between the desired and INN seismic response for lower values (without damping) for Chamoli earthquake at Barkot in NE direction with $\omega = [0.4, 0.6]$

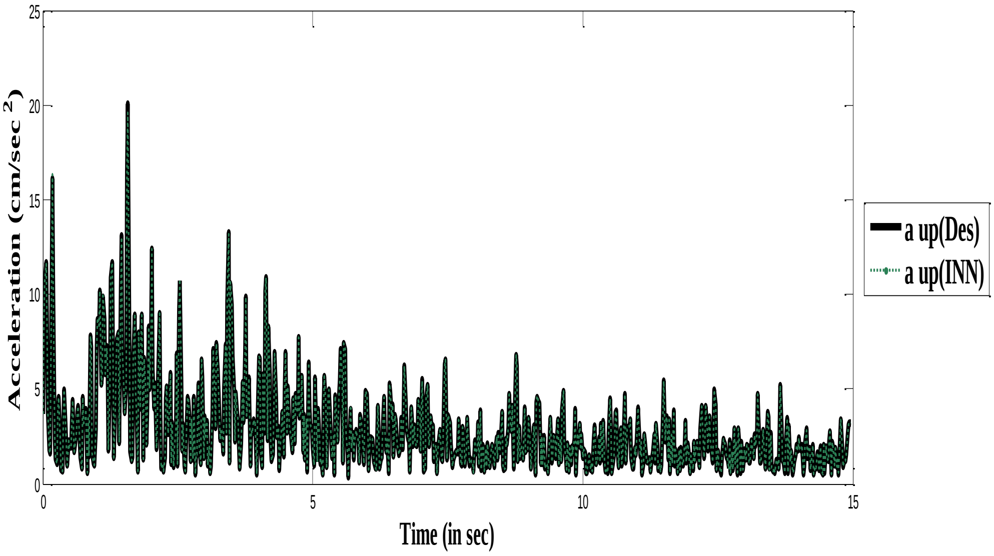

Figure 5.6(b): Comparison between the desired and INN seismic response for upper values (without damping) for Chamoli earthquake at Barkot in NE directions with $\omega = [0.4, 0.6]$

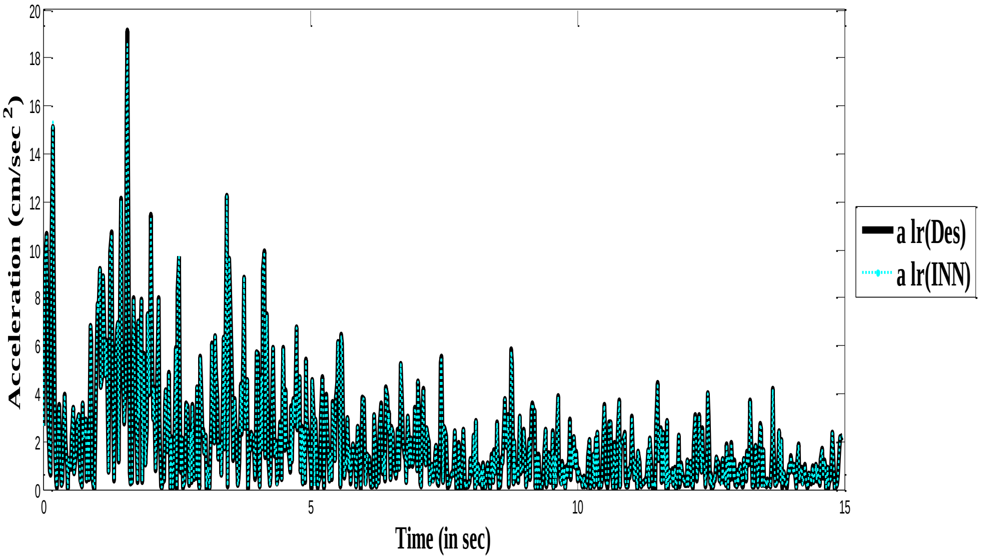

Figure 5.7(a): Comparison between the desired and INN seismic response for lower values (without damping) for Chamoli earthquake at Barkot in NE direction with ω = [0.01, 0.03]

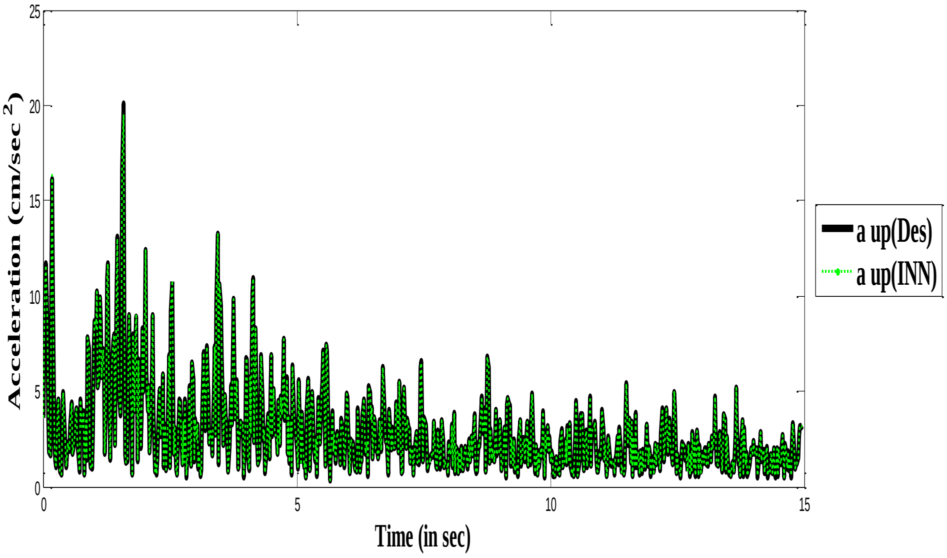

Figure 5.7(b): Comparison between the desired and INN seismic response for upper values (without damping) for Chamoli earthquake at Barkot in NE directions with ω = [0.01, 0.03]

Example for case (iii):

The system with damping is considered in case (iii). Considering the ground acceleration of Uttarkashi earthquake at Barkot (NE) in interval form, the response is computed using Eq. (5.7). Obtained responses and the ground acceleration in interval form are trained by the said IANN model for an example structural system with frequency parameter in crisp form as $\omega = 0.68981$ and damping as [1.48033, 1.68033]. Training was done for the total time range 0 to 14.96 secs. (749 data points). After training, ground acceleration and response data for Uttarkashi earthquake for various nodes in hidden layer it was confirmed that 18 nodes are again sufficient for the prediction. So, the weights corresponding to 18 hidden nodes are stored and they are used to predict responses for various intensity earthquakes. Figures 5.8(a) and 5.8(b) show the response comparison between the desired and INN data.

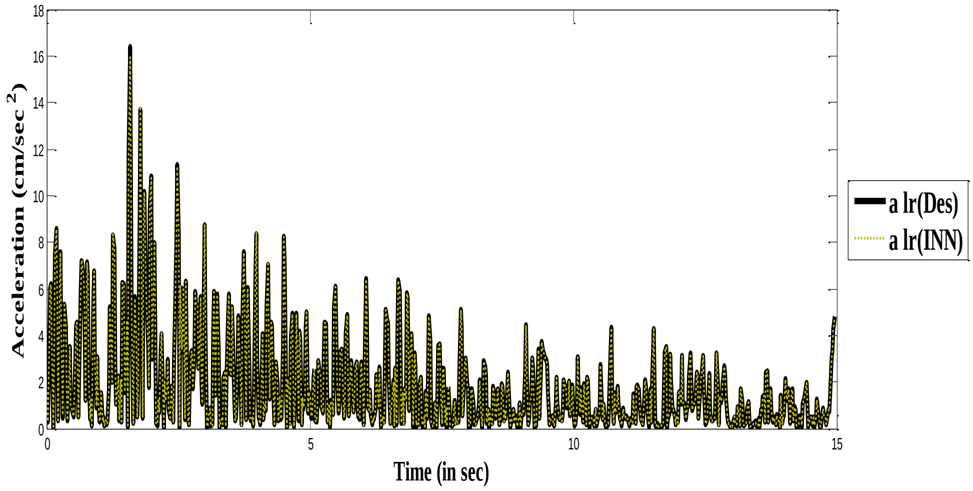

Figure 5.8(a): Comparison between the desired and INN seismic response for lower values for Uttarkashi earthquake at Barkot in NE direction with ω =0.68981 and damping [1.48033, 1.68033]

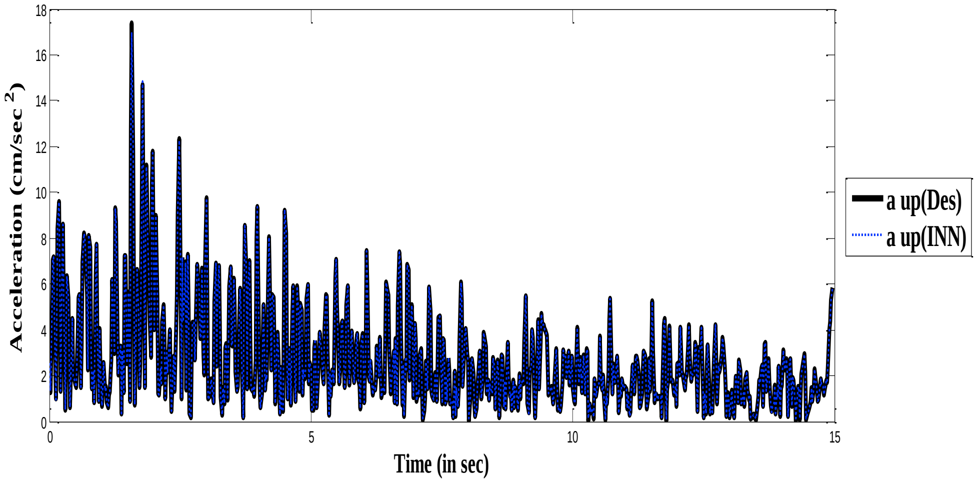

Figure 5.8(b): Comparison between the desired and INN seismic response for upper values for Uttarkashi earthquake at Barkot in NE direction with ω =0.68981 and damping [1.48033, 1.68033]

Example for case (iv):

For case (iv) the system is again considered with damping. Ground acceleration in interval form is used to calculate the required response of the structure in interval form with frequency parameter in interval form as $\omega = [0.58981, 0.78981]$and damping as [1.48033, 1.68033]. The data are trained with different hidden nodes in the hidden layer and it was found that 16 hidden nodes are sufficient to get an accuracy of 0.001. Comparison between the desired and INN are shown in Figures 5.9(a) and 5.9(b). After training the ground acceleration and response data in interval form for Uttarkashi earthquake at Barkot (NE) for different hidden nodes in hidden layer, the weights are stored and they are used to predict responses for various intensity earthquakes.

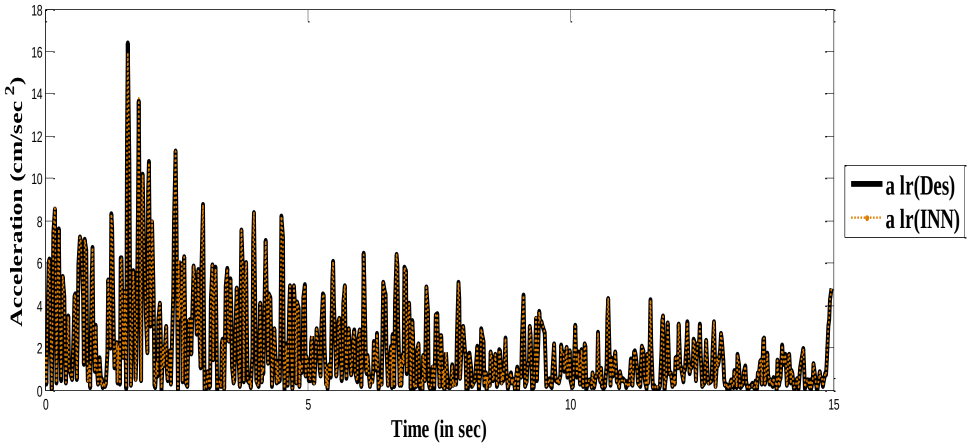

Figure 5.9(a): Comparison between the desired and INN seismic response for lower values for Uttarkashi earthquake at Barkot in NE direction with $\omega =$ [0.58981, 0.78981] and damping [1.48033, 1.68033]

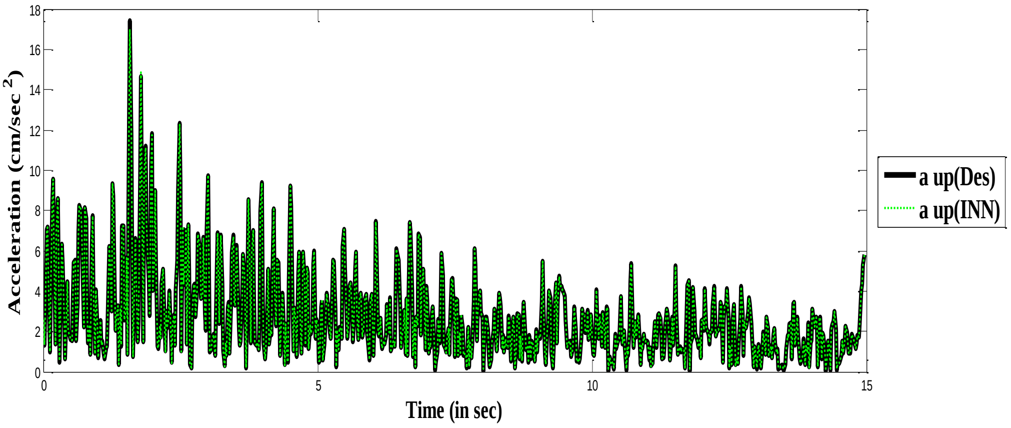

Figure 5.9(b): Comparison between the desired and INN seismic response for upper values for Uttarkashi earthquake at Barkot in NE directions with $\omega =$ [0.58981, 0.78981] and damping [1.48033, 1.68033]

Example for case (v) (Testing case with damping):

Finally in case (v) the training is extended with various intensities for time range 0 to 9.98 secs. (500 data points) and tested with different hidden layer nodes. It was found that the response result is almost same and good for 15 to 20 nodes in the hidden layer. But here 16 hidden layer nodes are used to generate the results for 500 data points without damping with frequency parameter in interval form. Figures 5.10(a) and 5.10(b) show response comparison between INN and desired for the 80% of Uttarkashi earthquake at Barkot (NE) for $\omega = [0.4, 0.6]$ using the stored converged weights of Chamoli earthquake directly. Similarly, the response comparison for 120% Uttarkashi earthquake at Barkot (NE) for $\omega = [0.01, 0.03]$ using the converged weights of Chamoli earthquake are shown in Figures 5.11(a) and 5.11(b).

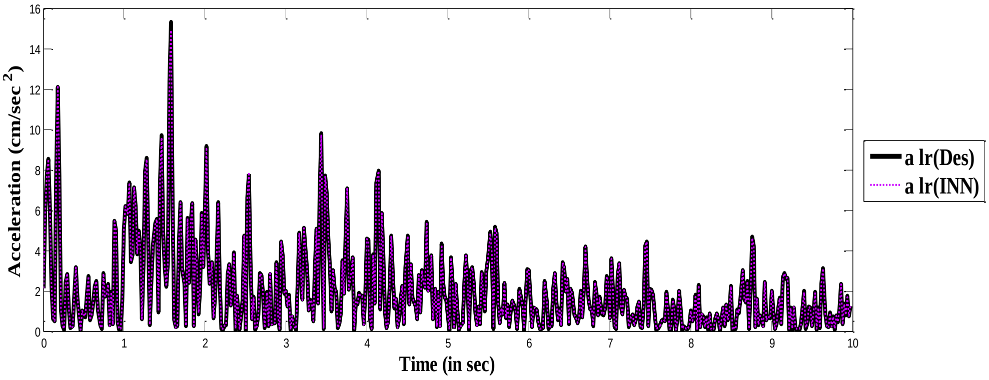

Figure 5.10(a): Comparison between the desired and INN of 80% seismic response for lower values (without damping) for Uttarkashi earthquake at Barkot (NE) with ω = [0.4, 0.6]

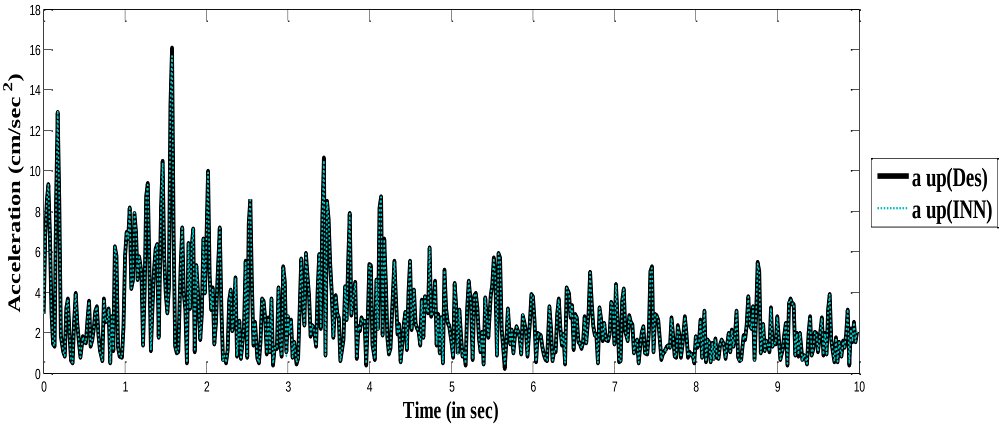

Figure 5.10(b): Comparison between the desired and INN of 80% seismic response for upper values (without damping) for Uttarkashi earthquake at Barkot (NE) with ω = [0.4, 0.6]

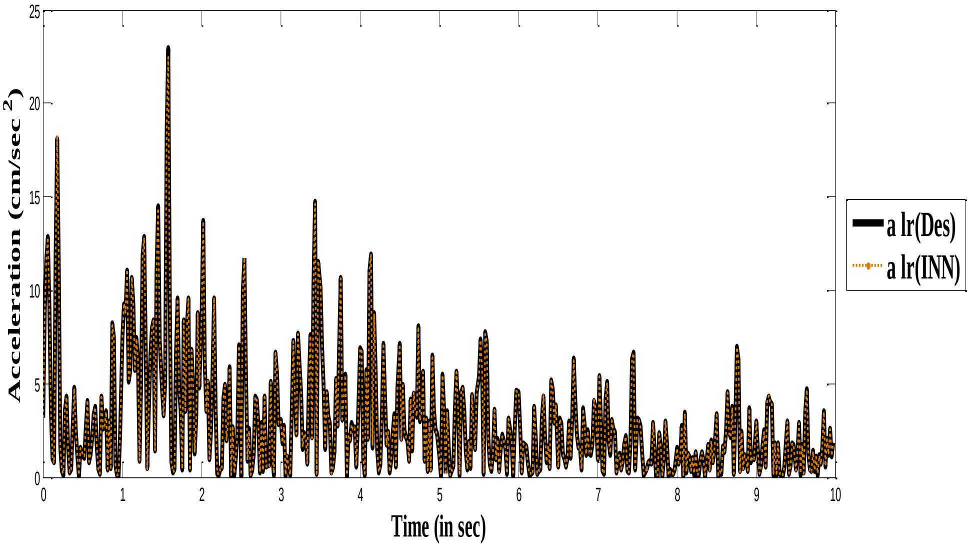

Figure 5.11(a): Comparison between the desired and INN of 120% seismic response for lower values (without damping) for Uttarkashi earthquake at Barkot (NE) with $\omega = [0.01, 0.03]$

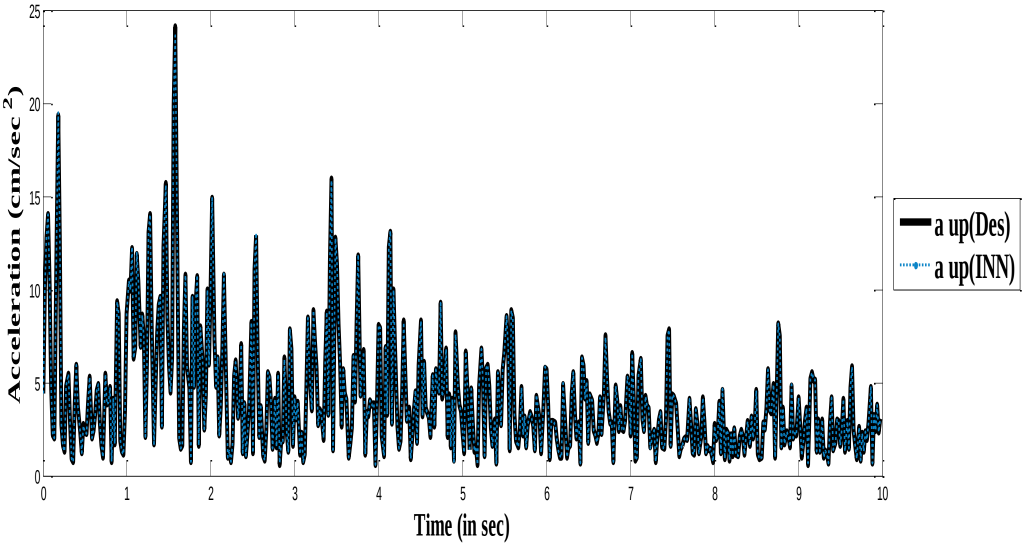

Figure 5.11(b): Comparison between the desired and INN of 120% seismic response for upper values (without damping) for Uttarkashi earthquake at Barkot (NE) with $\omega = [0.01, 0.03]$

Example for case (v) (Testing case without damping):

The training is also extended with damping for various intensities within the time range 0 to 9.98 secs. (500 data points) and tested with different hidden layer nodes. It was found that the response result is almost same and good for 16 hidden layer nodes. The comparison between the desired and ANN response data for 80% of Chamoli earthquake acceleration at Barkot (NE) with $\omega = [0.58981, 0.78981]$ and damping = [1.48033, 1.68033] using the converged weights of Uttarkashi earthquake are shown in Figures 5.12(a) and 5.12(b). Similarly, for 120% of Chamoli earthquake acceleration at Barkot (NE) with $\omega = [0.58981, 0.78981]$and damping = [1.48033, 1.68033] and the response comparison between neural and desired are plotted in Figures 5.13(a) and 5.13(b).

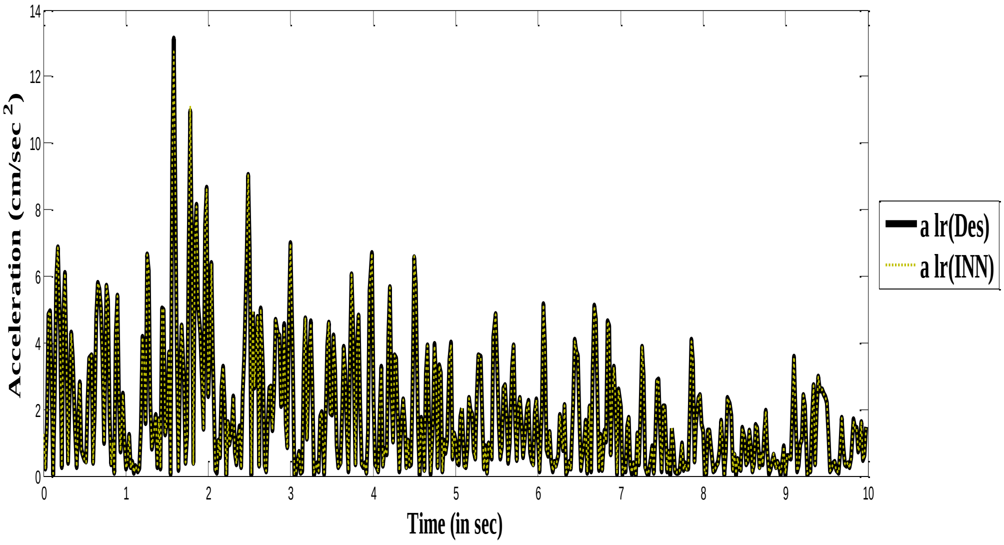

Figure 5.12(a): Comparison between the desired and INN of 80% seismic response for lower values for Chamoli earthquake at Barkot (NE) with ω = [0.58981, 0.78981] and damping [1.48033, 1.68033]

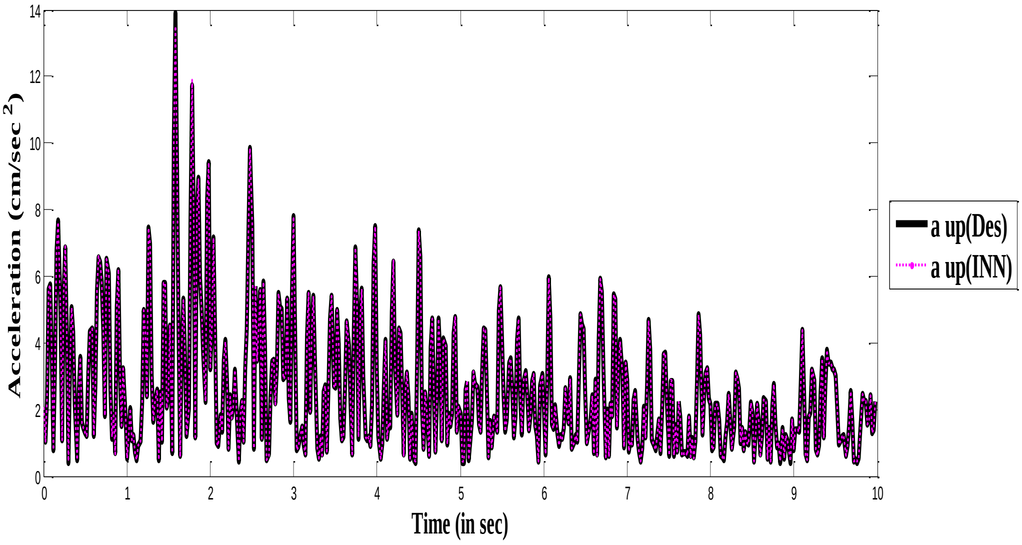

Figure 5.12(b): Comparison between the desired and INN of 80% seismic response for upper values for Chamoli earthquake at Barkot (NE) with ω = [0.58981, 0.78981] and damping [1.48033, 1.68033]

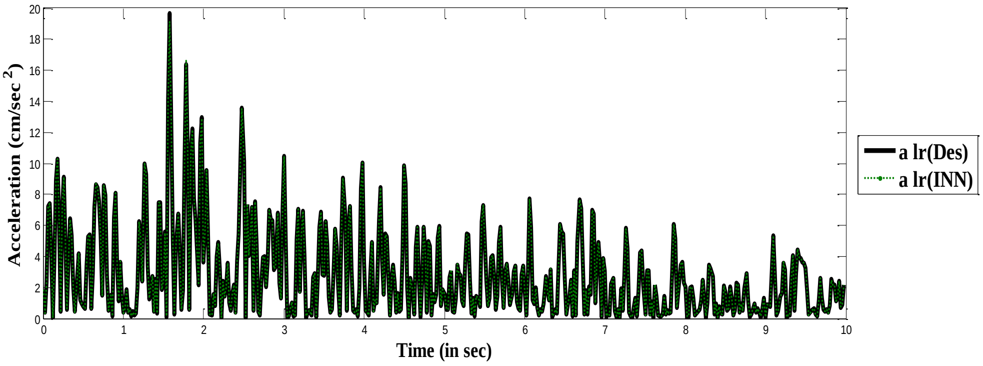

Figure 5.13(a): Comparison between the desired and INN of 120% seismic response for lower values for Chamoli earthquake at Barkot (NE) with ω = [0.58981, 0.78981] and damping [1.48033, 1.68033]

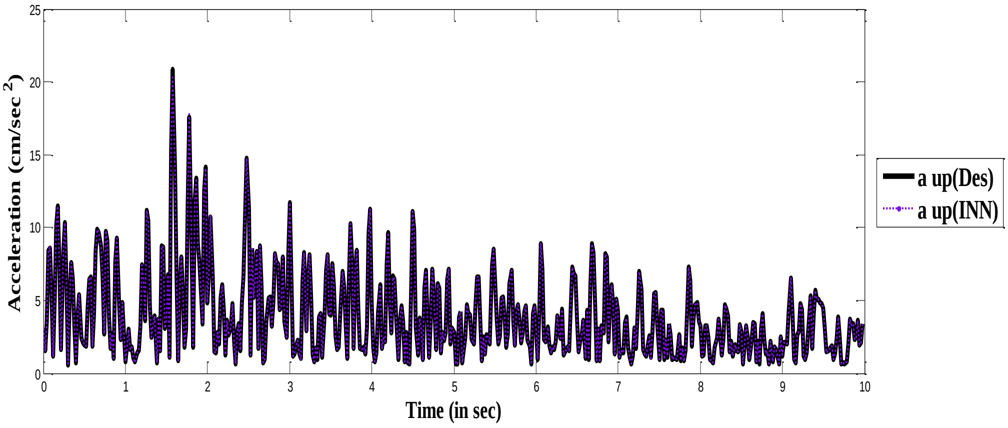

Figure 5.13(b): Comparison between the desired and INN of 120% seismic response for upper values for Chamoli earthquake at Barkot (NE) with ω = [0.58981, 0.78981] and damping [1.48033, 1.68033]

Comparison between desired and INN peak acceleration values (testing) with various intensities of Uttarkashi (without damping) and Chamoli earthquake acceleration at Barkot (NE) (with damping) has been presented in Table 5.1.

Table 5.1: Comparison between the desired and INN Peak Acceleration values (Testing Values)

Case	Intensities	Desired		INN	
		Lower	Upper	Lower	Upper
Without Damping (Uttarkashi at Barkot NE)	80%	15.2704	16.0704	14.7954	15.6091
	120%	22.9056	24.1056	22.3373	23.4528
With Damping (Chamoli at Barkot NE)	80%	13.108	13.908	12.7279	13.4997
	120%	19.662	20.862	19.0955	20.2757

5.10.2 Single Degree of Freedom System for Fuzzy Case

For present scenario two Indian earthquakes, the Chamoli Earthquake at Barkot in NE (north east) direction and the Uttarkashi earthquake at Barkot in NE (north-east) direction have been considered for training and testing for different cases. The various cases considered are shown below:

Case (i): Without damping: Ground acceleration as well as response data in fuzzified form with crisp frequency.
Case (ii): Same as (i), but with fuzzy frequency
Case (iii): With damping: Ground acceleration as well as response data in fuzzified form with crisp frequency and fuzzy damping.
Case (iv): Same as (iii), but with fuzzy frequency
Case (v): Testing for different earthquake data with / without damping.

The earthquake acceleration data are actually both positive and negative. As mentioned in the Introduction that the transformation Eq. (5.15) first converts the data from bipolar to unipolar i.e., from range [-1, 1] to range [0, 1], then the data are transferred back to its bipolar form. The data in fuzzified form are converted to h-level form first and then the transformation is applied.

Example for case (i):

For case (i), initially the system without damping is studied and for that the system is subjected to Chamoli Earthquake with maximum ground acceleration in fuzzified form as [19.088, 19.588, 20.088] cm/sec^{2}at Barkot in NE (north-east) direction. The response of the structure is obtained by solving Eq. (5.11). The ground acceleration in fuzzified form is converted to h-level form and is used for training. The mass in fuzzified is taken as $\hat{M} = [1,1,1]$ for all cases. The stiffness parameter is not used here directly as we are considering the frequency parameter in the model. Frequency parameter in crisp form is taken as ω=0.02 with time range 0 to 14.96 sec. (749 data points) for the mentioned earthquake. Comparisons between desired and FNN are shown in Figures 5.14(a)-5.14(c). Figure 5.14(a) shows the plot for lower values, Figure 5.14(b) shows the plot for centre values and Figure 5.14(c) depicts the upper values. Simulations have been done for different hidden layer nodes and it was seen that the response result is almost same and good for 15 to 18 nodes in the hidden layer. However, 18 hidden layer nodes are used here to generate the results for 749 data points.

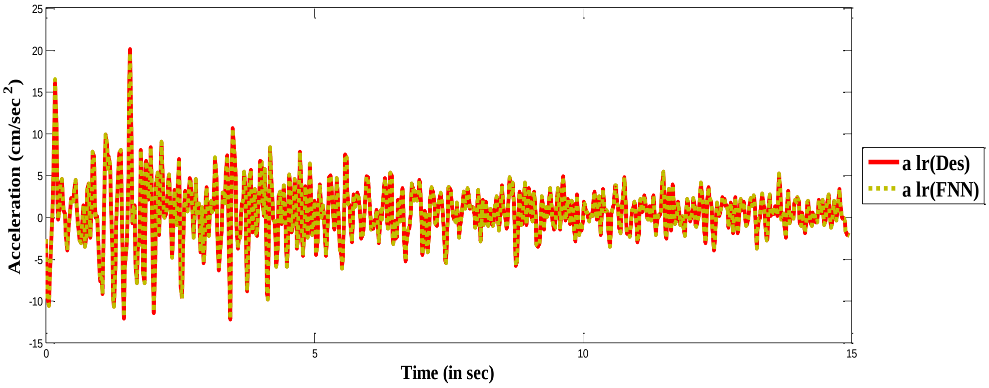

Figure 5.14(a): Comparison between the desired and FNN acceleration for lower values (without damping)

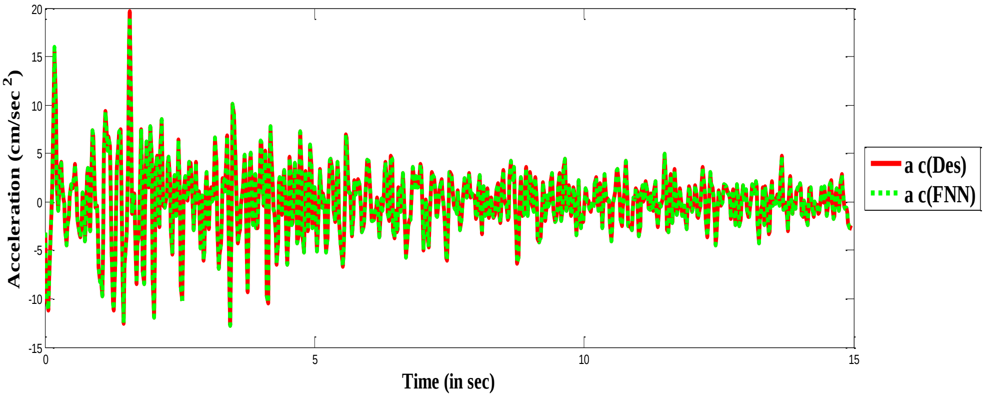

Figure 5.14(b): Comparison between the desired and FNN acceleration for centre values (without damping)

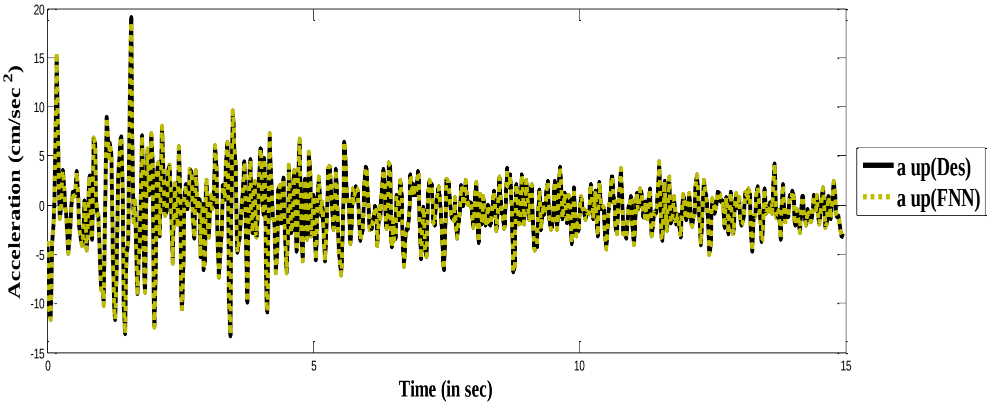

Figure 5.14(c): Comparison between the desired and FNN acceleration for upper values (without damping)

Example for case (ii):

In case (ii), the system is considered without damping with the same earthquake. Here ground acceleration is taken as input and structural displacement as output. The neural network architecture is trained within the time range 0 to 14.96 sec. (749 data points) taking frequency parameter in fuzzified form as $\hat{\omega} = [0.4, 0.5, 0.6]$. Training has been done for different hidden layer nodes. Again 18 hidden layer nodes are used to generate the results for 749 data points. The displacement time plots are plotted in Figures 5.15(a)-5.15(c).

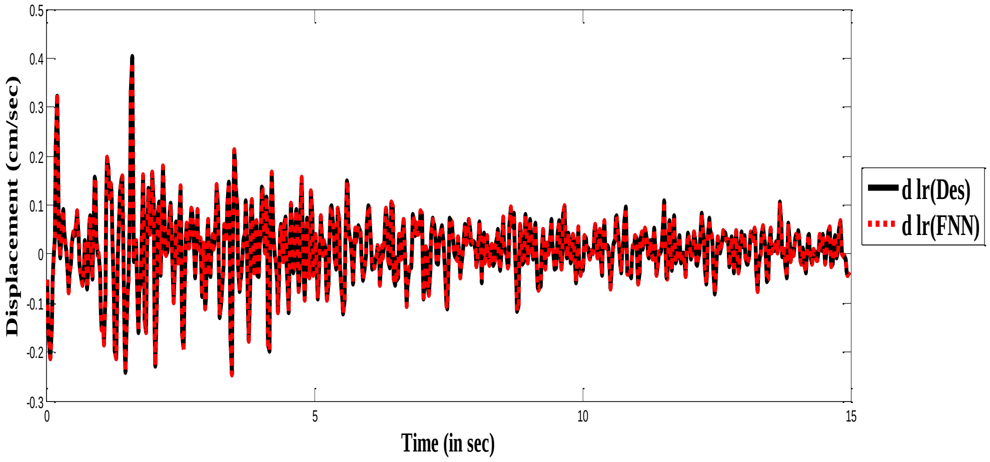

Figure 5.15(a): Comparison between the desired and FNN displacement for lower values (without damping)

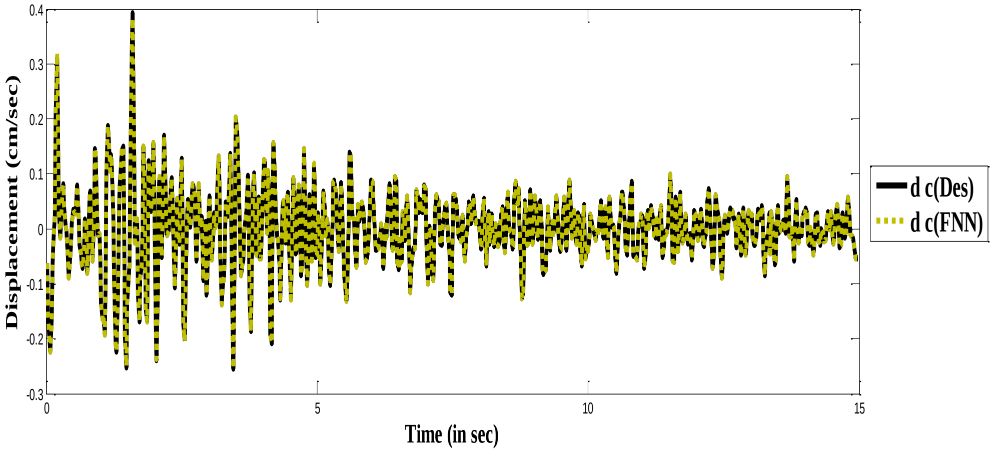

Figure 5.15(b): Comparison between the desired and FNN displacement for centre values (without damping)

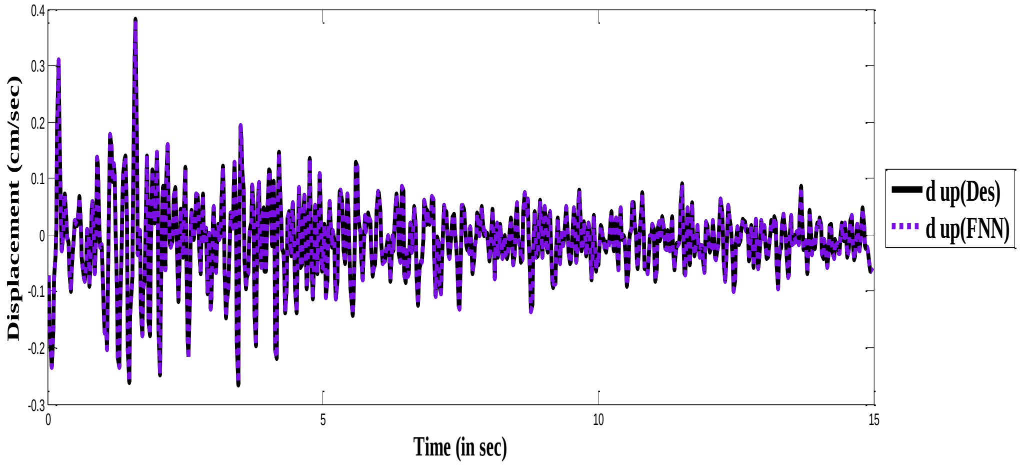

Figure 5.15(c): Comparison between the desired and FNN displacement for upper values (without damping)

Example for case (iii):

The system with damping is taken in case (iii). The ground acceleration of Uttarkashi earthquake at Barkot (NE) and the displacement in h-level form are used as training patterns. The training patterns are trained by the FNN model with frequency parameter in crisp form as $\omega = 0.68981$ and damping $\hat{C} = [1.48033, 1.58033, 1.68033]$. Training was done for a total time range of 0 to 14.96 sec (749 data points). After training ground acceleration and response data for Uttarkashi earthquake for various nodes in hidden layer it was confirmed that 18 nodes are again sufficient for the prediction. So, the weights corresponding to 18 hidden nodes are stored and they are used to predict responses for various intensity earthquakes. Figures 5.17(a)-5.17(c) show the displacement time plots between the desired and FNN.

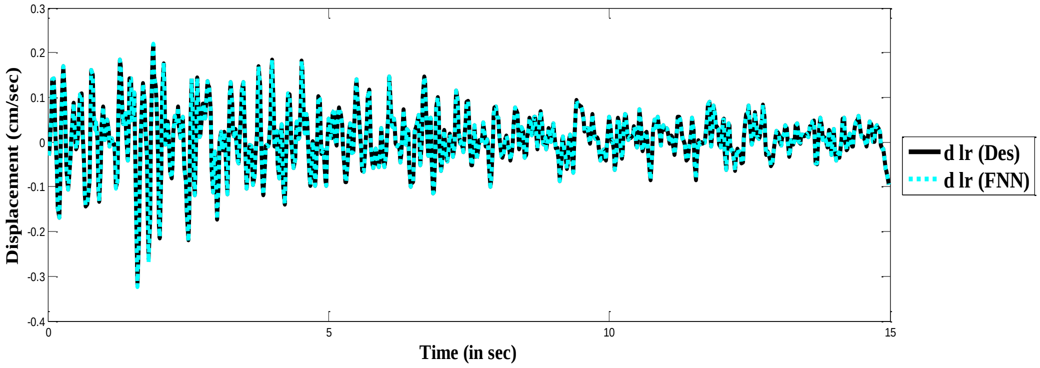

Figure 5.16(a): Comparison between the desired and FNN displacement for lower values (with damping)

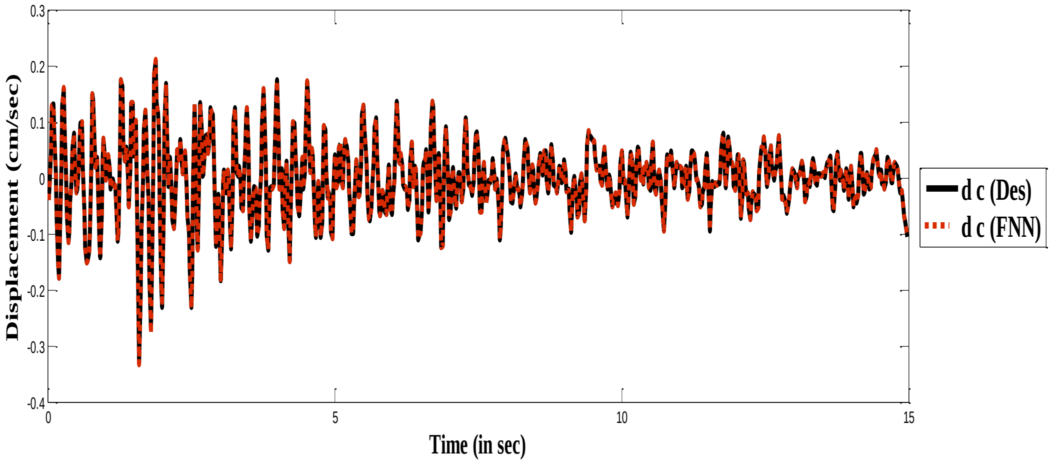

Figure 5.16(b): Comparison between the desired and FNN displacement for centre values (with damping)

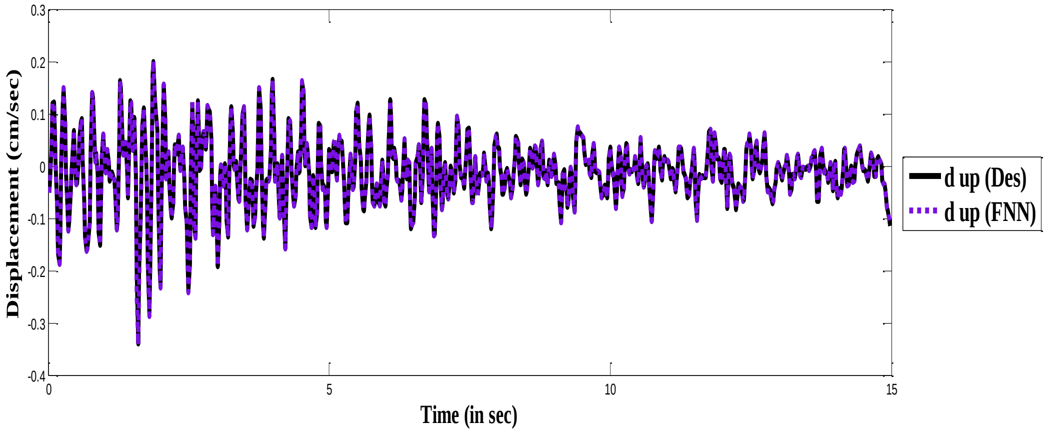

Figure 5.16(c): Comparison between the desired and FNN displacement for upper values (with damping)

Example for case (iv):

For case (iv), the system is again considered with damping. The response of the structure in fuzzified form is computed using Eq. (5.14). Then the ground acceleration and structural response in h-level form is used for training. Here frequency parameter in fuzzified form is taken as $\hat{\omega} = [0.58981, 0.68981, 0.78981]$ and damping $\hat{C} = [1.48033,$ 1.58033, 1.68033]. The data are trained with different hidden nodes in the hidden layer and it was found that 16 hidden nodes are sufficient to get an accuracy of 0.001. Comparison between the desired and FNN is shown in Figures 5.17(a)-5.17(c). After training the ground acceleration and response data for Uttarkashi earthquake at Barkot

(NE) for different hidden nodes in hidden layer, the weights are stored and they are used to predict responses for various intensity earthquakes.

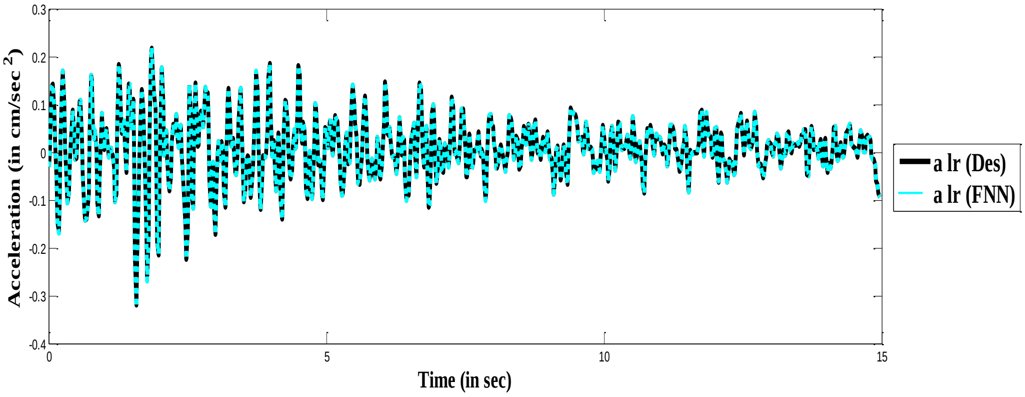

Figure 5.17(a): Comparison between the desired and FNN acceleration for lower values (with damping)

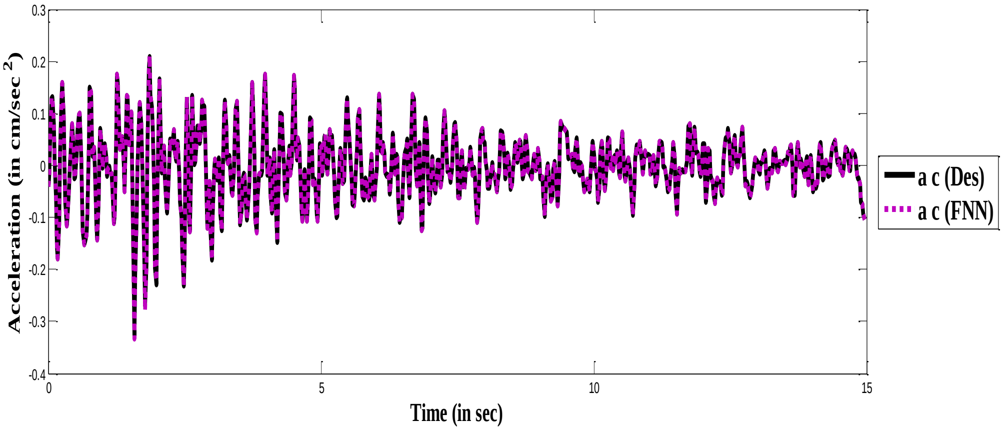

Figure 5.17(b): Comparison between the desired and FNN acceleration for centre values (with damping)

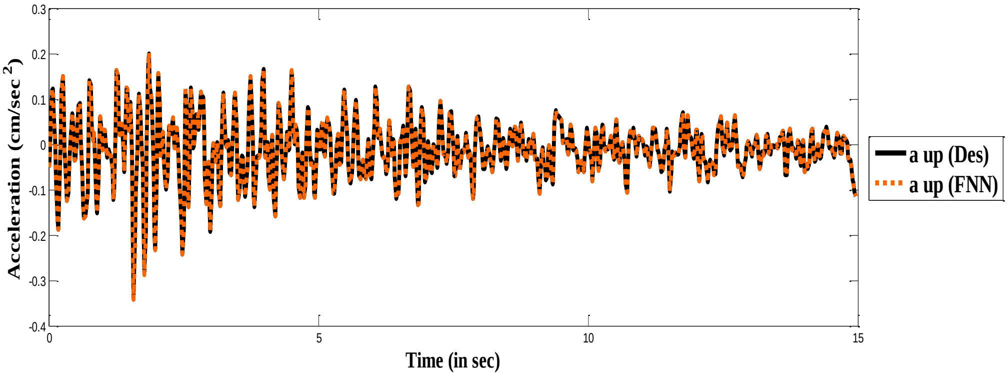

Figure 5.17(c): Comparison between the desired and FNN acceleration for upper values (with damping)

Example for case (v) (Testing case without damping):

Finally in case (v) the training is extended with damping and without damping for various intensities with time range 0 to 9.98 sec. (500 data points) and are tested with different hidden nodes in hidden layer. It was found that the response result is almost the same and good for 15 to 20 nodes in the hidden layer. But here 16 hidden layer nodes are used to generate the results for 500 data points without damping with frequency parameter in fuzzified form. Figures 5.18(a)-5.18(c) show displacement time plot comparison between FNN and desired for 80% of Uttarkashi earthquake at Barkot (NE) for $\hat{\omega} = [0.01, 0.02, 0.03]$ using the stored converged weights of Chamoli earthquake directly.

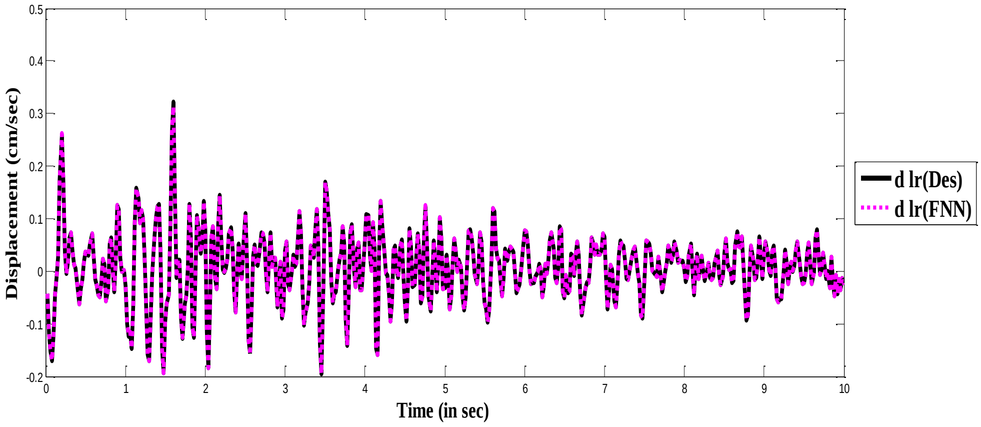

Figure 5.18(a): Comparison between the desired and FNN (Testing) displacement of 80% for lower values (without damping)

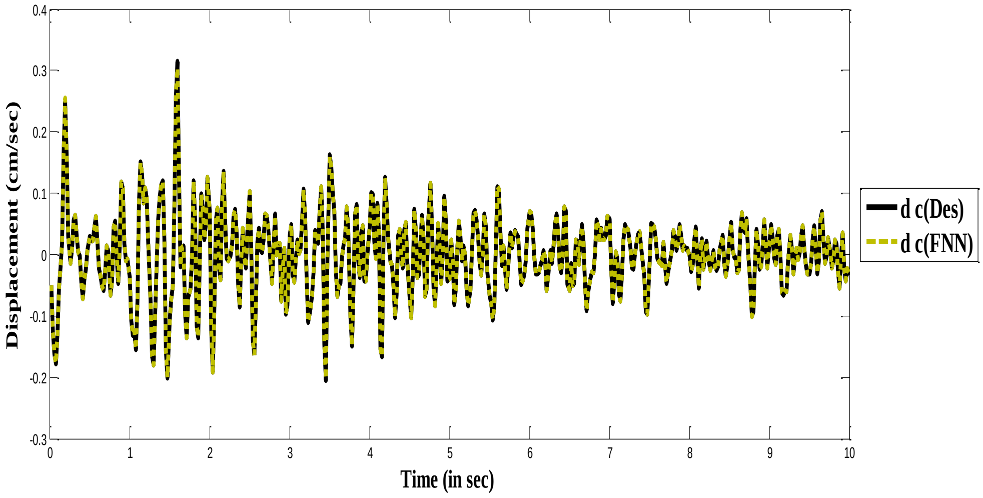

Figure 5.18(b): Comparison between the desired and FNN (Testing) displacement of 80% for centre values (without damping)

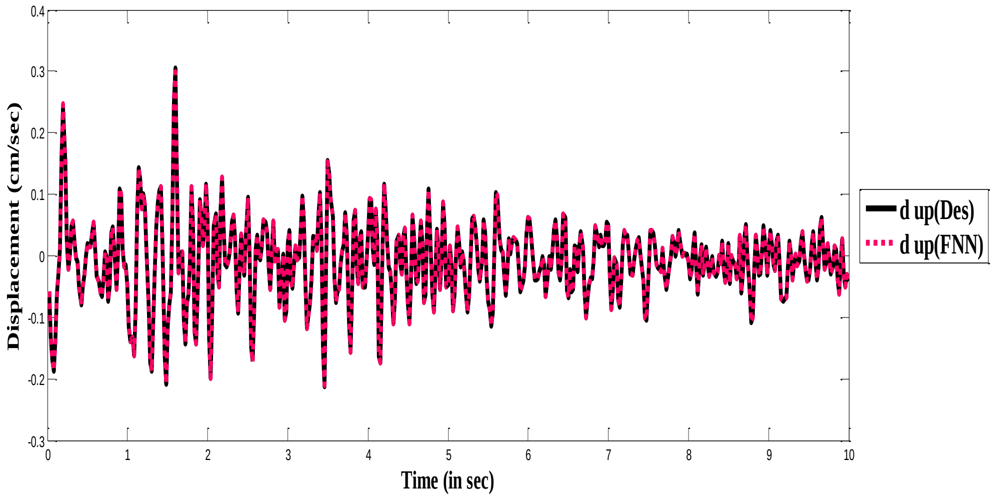

Figure 5.18(c): Comparison between the Desired and FNN (Testing) displacement of 80% for upper values (without damping)

Example for case (v) (Testing case with damping):

Similarly the training with damping for time range 0 to 9.98 sec (500 data points) is done and is tested with different hidden nodes. It was found that the response result is almost the same and good for 16 hidden layer nodes. The comparison between the desired and FNN response data for 120% of Chamoli earthquake acceleration at Barkot (NE) with $\hat{\omega} = [0.58981, 0.68981, 0.78981]$ and damping $\hat{C} = [1.48033, 1.58033, 1.68033]$ using the converged weights of Uttarkashi earthquake are shown in Figures 5.19(a)-5.19(c).

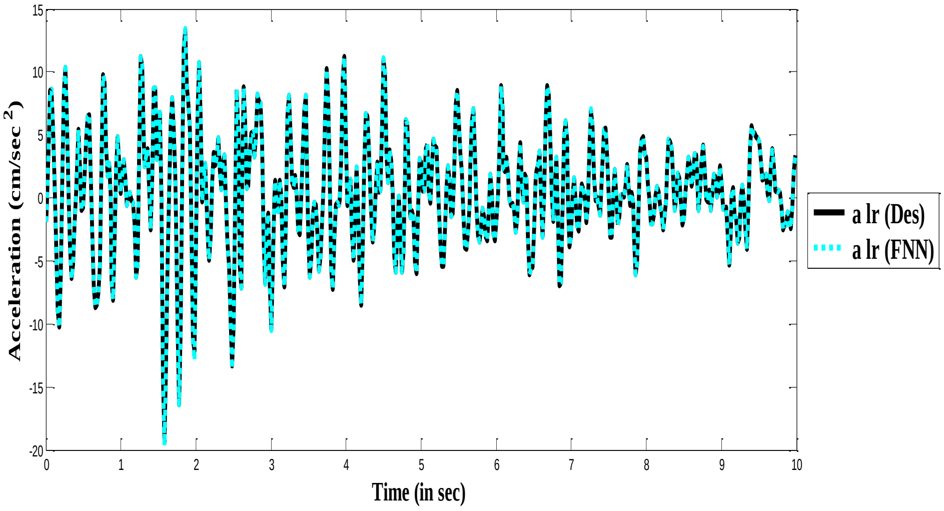

Figure 5.19(a): Comparison between the desired and FNN (Testing) acceleration of 120% for lower values (with damping)

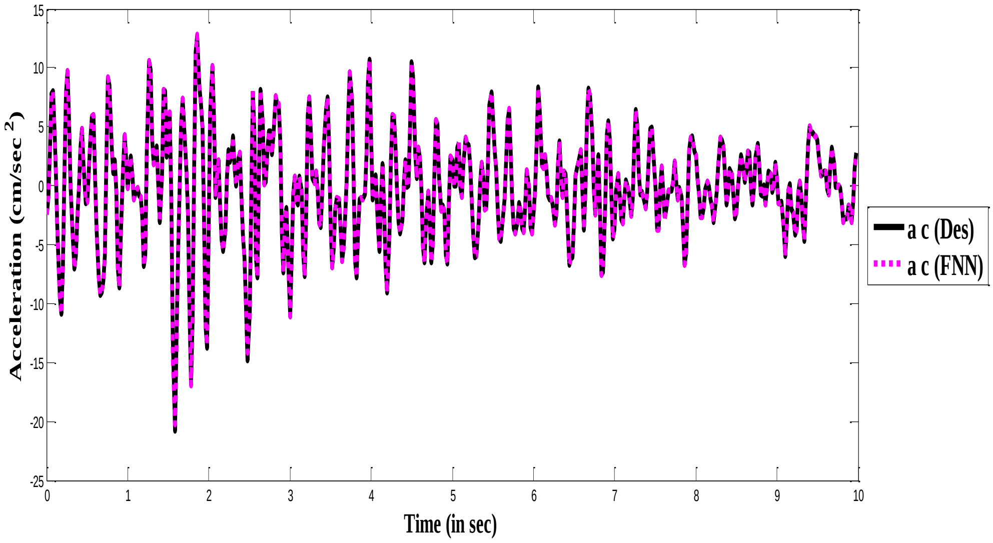

Figure 5.19(b): Comparison between the desired and FNN (Testing) acceleration of 120% for centre values (with damping)

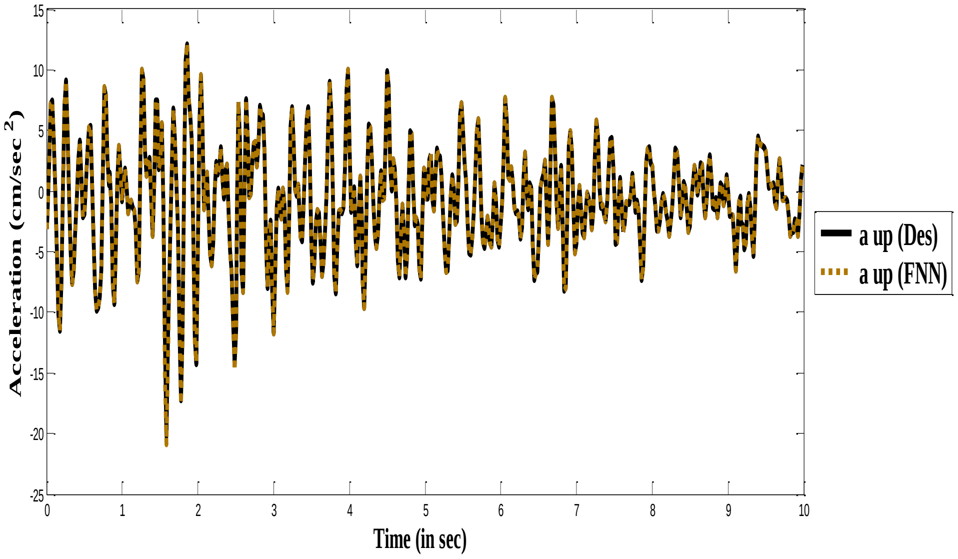

Figure 5.19(c): Comparison between the desired and FNN (Testing) acceleration of 120% for upper values (with damping)

A comparison between desired and FNN peak acceleration values (testing) with various intensities of Uttarkashi (without damping) and Chamoli earthquake acceleration at Barkot (NE) (with damping) is presented in Table 5.2. The error % between desired and FNN Peak values of predicted data is given in Table 5.3.

Table 5.2: Comparison between the Desired and FNN Peak Acceleration values (Testing Values)

Case	Intensities	Desired			FNN		
		Lower	Centre	Upper	Lower	Centre	Upper
Without Damping (Uttarkashi at Barkot NE)	80%	15.27	15.67	16.07	14.95	15.34	15.71
	120%	22.90	23.50	24.10	22.46	23.08	23.64
With Damping (Chamoli at Barkot NE)	80%	13.10	13.50	13.90	12.72	13.12	13.51
	120%	19.66	20.26	20.86	19.42	20.81	20.21

Table 5.3: Error % between Desired and FNN (Peak Values)

Case	Response	Error %		
		Lower	Centre	Upper
Without Damping (Chamoli at Barkot NE)	Acceleration ω =0.02	2	2	2
	Displacement $\hat{\omega} = [0.4, 0.5, 0.6]$	2	2	2
With Damping (Uttarkashi at Barkot NE)	Displacement ω =0.68981	0.08	0. 06	0. 04
	Acceleration $\hat{\omega} = [0.589, 0.689, 0.6789]$	0.8	0.00	0.00

5.10.3 Multi Degree of Freedom System for Interval Case

In the present investigation, examples of two, six and ten storey structures have been considered. The Chamoli earthquake data at Barkot in NE (north east) direction has been considered for training and different intensities of Uttarkashi earthquakes data at Barkot in NE direction are used for testing different storeys. The original Chamoli earthquake is plotted in Figure 5.20. The peak acceleration value of this Chamoli earthquake is 19.58 cm/sec^2. As discussed earlier that the earthquake acceleration data have values both in positive and negative. A transformation has been used here which converts all the bipolar data to unipolar form. After the data are converted within the range [0, 1] the INN algorithm is used and once the training is done again the data are transferred back to its bipolar form by means of the inverse transformation. Interval neural network training is done till a desired accuracy is achieved. The methodology has been discussed by giving the results for following three examples.

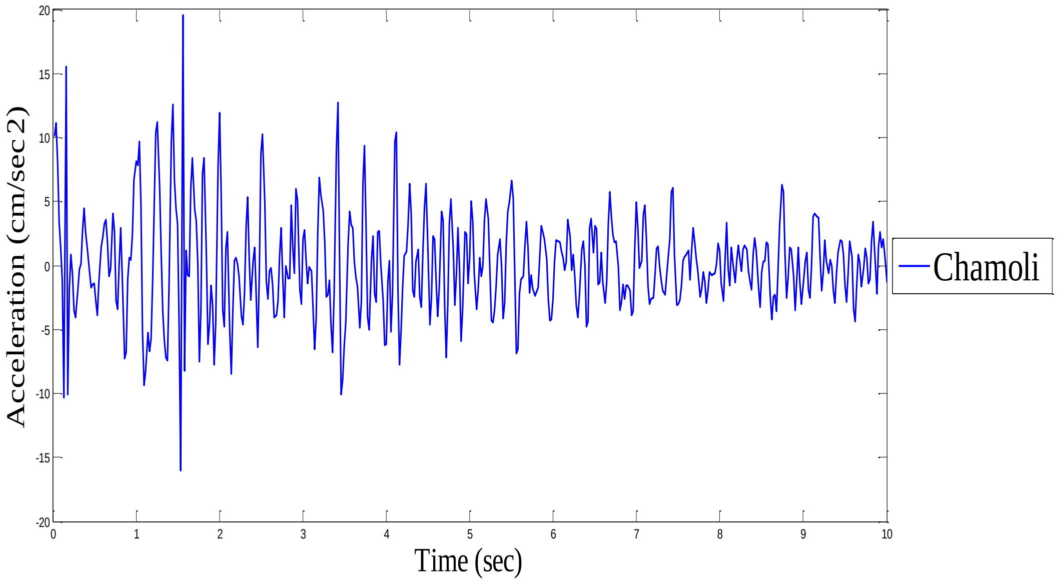

Figure 5.20: Chamoli earthquake, Barkot, March 29, 1999, Peak acceleration: 19.58 cm/sec^2

Example 1. Two storey shear buildings:

In the present investigation, an example of two storey structure (2DOF system) with natural frequency parameters [23.08, 39.58] and [60.22, 103.30] in interval form has been considered. The damping ratio in interval form is assumed as [8%, 10%] critical. Here, the Chamoli earthquake at Barkot (NE direction) has been considered for training with peak values as [19.08, 20.08]. The responses of the first and second storey are obtained numerically by solving the Duhamel integral, considering the ground acceleration of Chamoli earthquake at Barkot (NE).Training is done by INN using the ground acceleration and responses of the two storeys in interval form. This training was done for a total time range of 0-10 secs (500 data points, earthquake period) till desired accuracy reached as 0.001. After training the converged weights are stored for testing. The desired and INN results of first storey are shown in Figures 5.21(a)-5.21(b). Figures 5.22(a)-5.22(b) show the comparison between the neural and desired of second storey. The training is also done for time range of 0-4 secs (200 data) and for time range of 0-14 secs (700 data). The plot for 200 data points for first storey is shown in Figures 5.23(a)-5.23(b) and for second storey in Figures 5.24(a)-5.24(b). Similarly the plots with 700 data points are shown in Figures 5.25(a)-5.25(b) for first storey and in Figures 5.26(a)-5.26(b) for second storey.

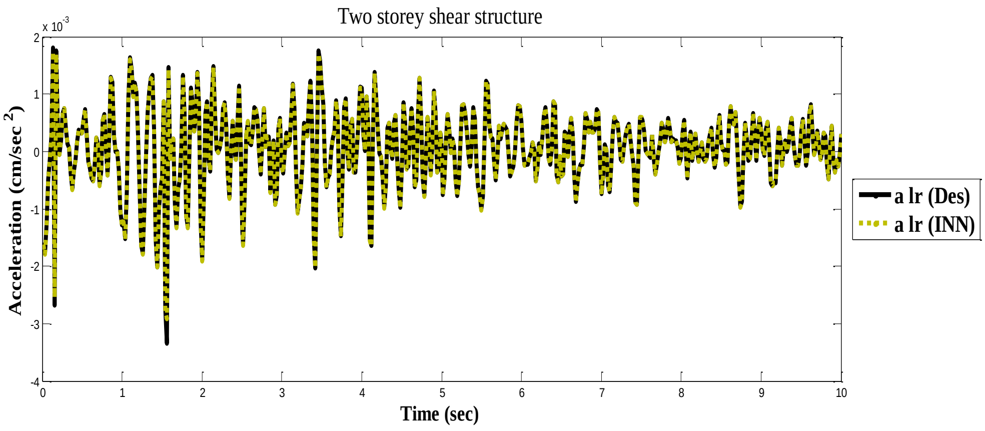

Figure 5.21(a): Comparison between the desired and INN response for lower values of Chamoli earthquake at Barkot (NE) for first storey (500data)

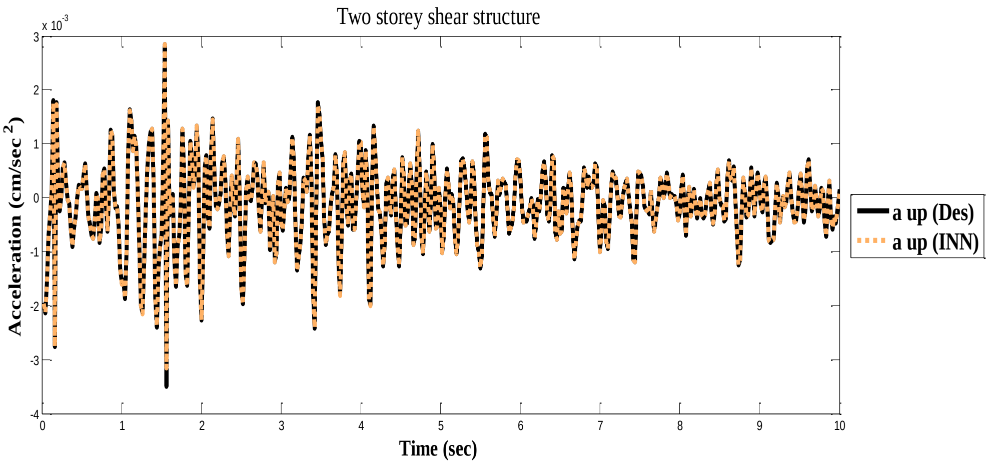

Figure 5.21(b): Comparison between the desired and INN response for upper values of Chamoli earthquake at Barkot (NE) for first storey (500 data)

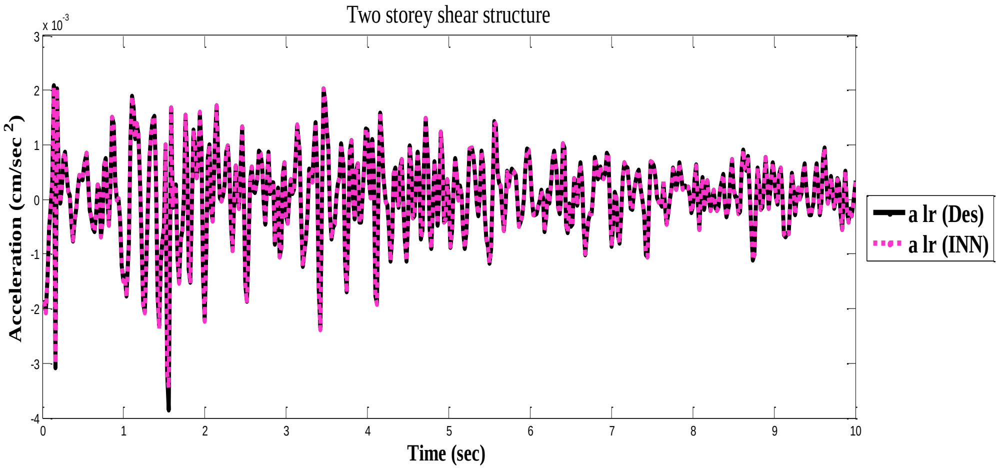

Figure 5.22(a): Comparison between the desired and INN response for lower values of Chamoli earthquake at Barkot (NE) for second storey (500 data)

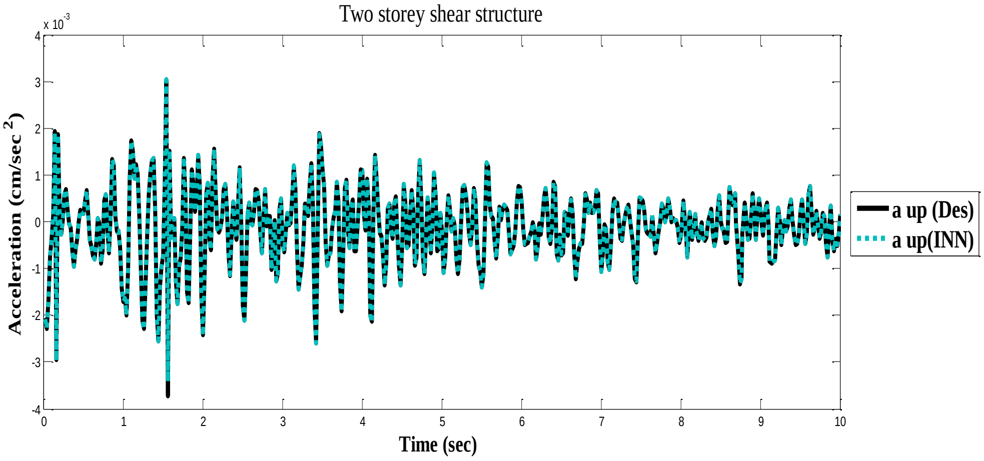

Figure 5.22(b): Comparison between the desired and INN response for upper values of Chamoli earthquake at Barkot (NE) for second storey (500 data)

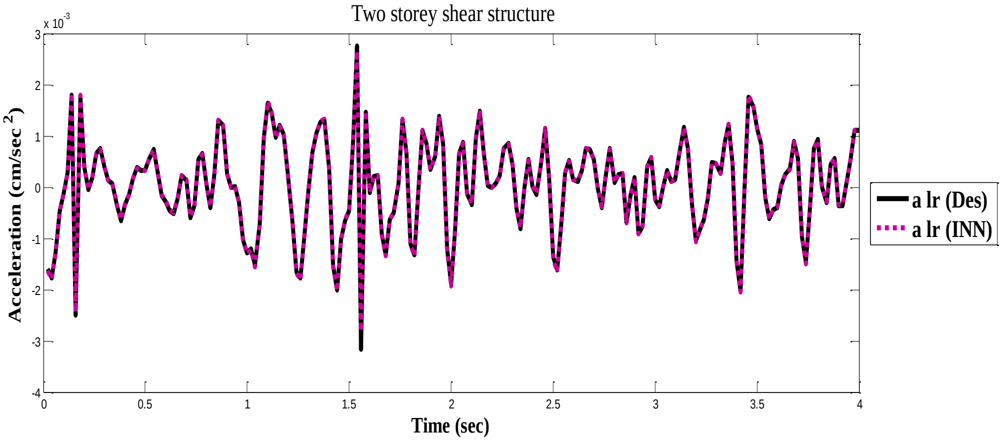

Figure 5.23(a): Comparison between the desired and INN response for lower values of Chamoli earthquake at Barkot (NE) for first storey (200 data)

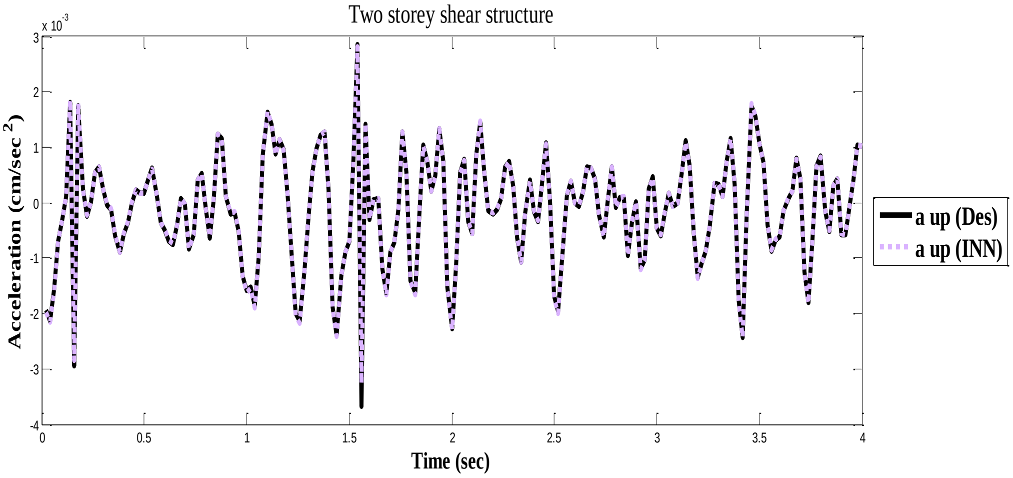

Figure 5.23(b): Comparison between the desired and INN response for upper values of Chamoli earthquake at Barkot (NE) for first storey (200 data)

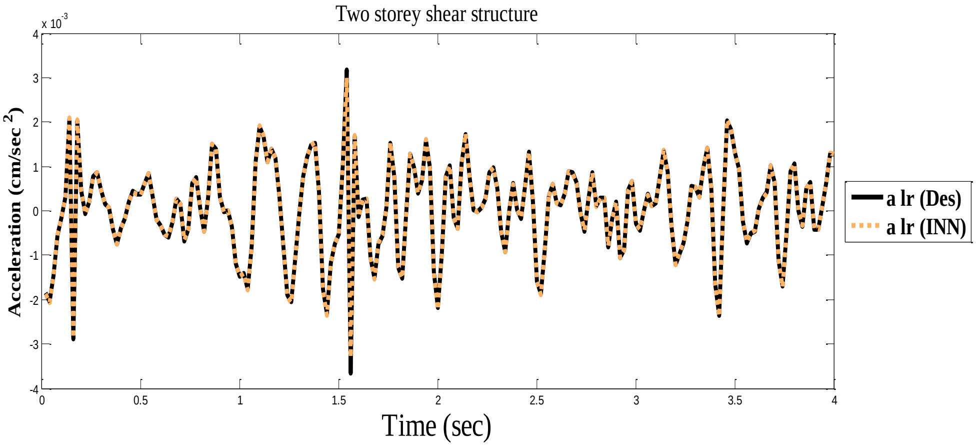

Figure 5.24(a): Comparison between the desired and INN response for lower values of Chamoli earthquake at Barkot (NE) for second storey (200 data)

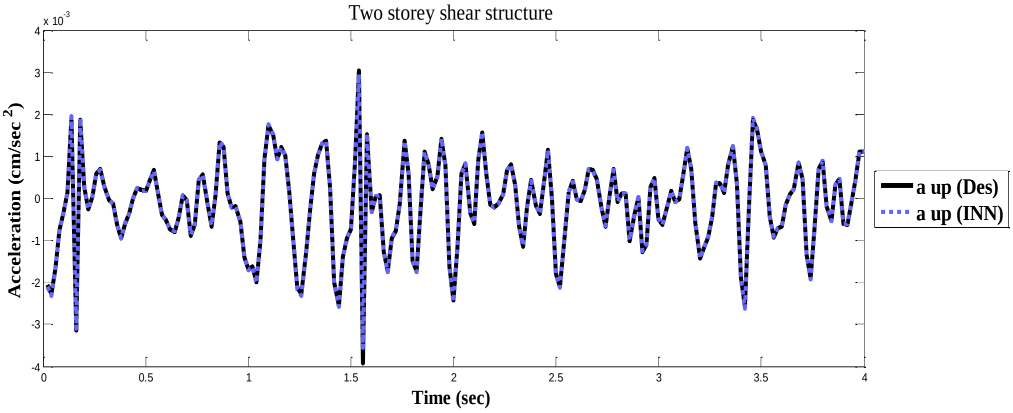

Figure 5.24(b): Comparison between the desired and INN response for upper values of Chamoli earthquake at Barkot (NE) for second storey (200 data)

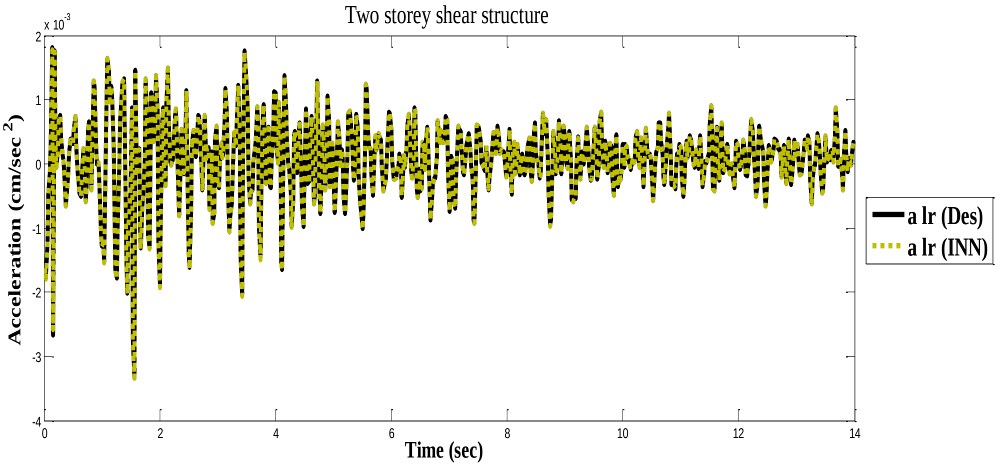

Figure 5.25(a): Comparison between the desired and INN response for lower values of Chamoli earthquake at Barkot (NE) for first storey (700 data)

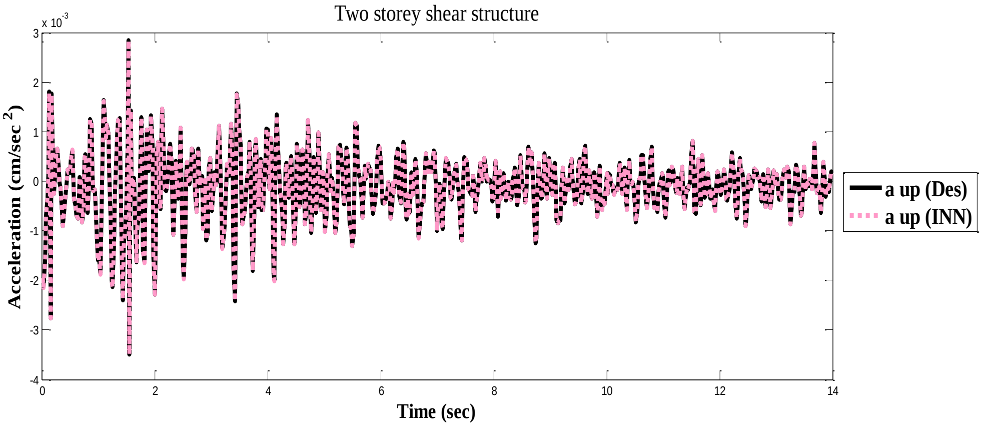

Figure 5.25(b): Comparison between the desired and INN response for upper values of Chamoli earthquake at Barkot (NE) for first storey (700 data)

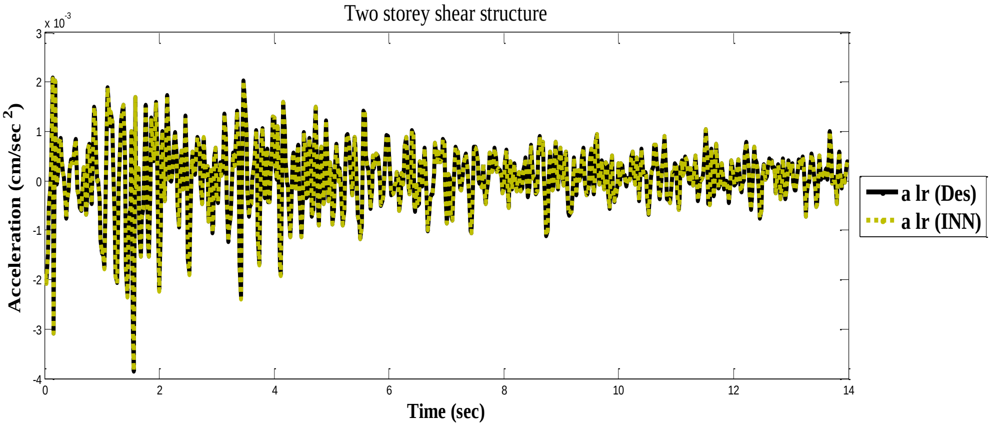

Figure 5.26(a): Comparison between the desired and INN response for lower values of Chamoli earthquake at Barkot (NE) for second storey (700 data)

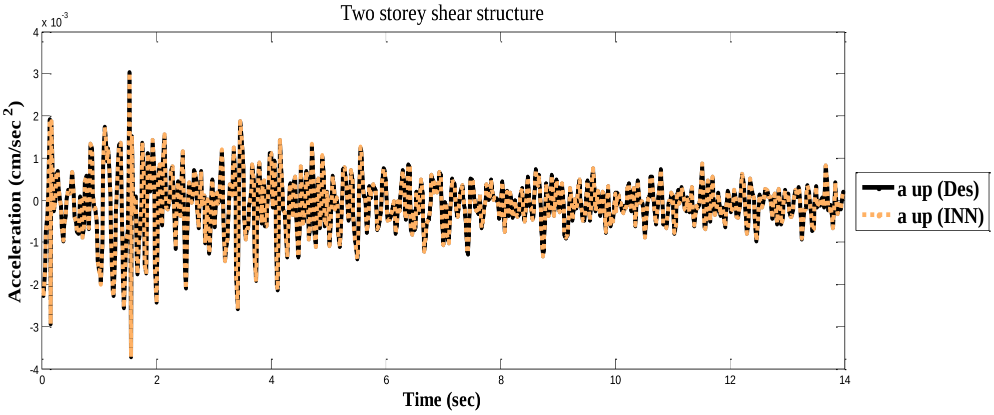

Figure 5.26(b): Comparison between the desired and INN response for upper values of Chamoli earthquake at Barkot (NE) for second storey (700 data)

Example 2. Six storey shear buildings:

Here, the damping ratio in interval form is assumed as [4%, 6%] critical. The Chamoli earthquake at Barkot (NE) direction has been considered here for training. The training is done in a similar way as done in example 1. This training was done for a total time range of 0-10 sec (500 points, earthquake period) with a desired accuracy of 0.001. After training the converged weights are stored. The neural and desired results obtained after training for sixth storey of the building are shown in Figures 5.27(a)-5.27(h). Figure 5.27(a) shows the data of lower case in [-1, 1] range, 5.27(b) shows conversion of the same data from [-1, 1] to [0, 1] using the transformation and 5.27(c) shows the conversion of data from [0, 1] to [-1, 1] using the inverse transformation. Finally the result comparison between the desired and INN is shown in Figure 5.27(d). For upper case, the results are plotted in Figures 5.27(e)-5.27(h).

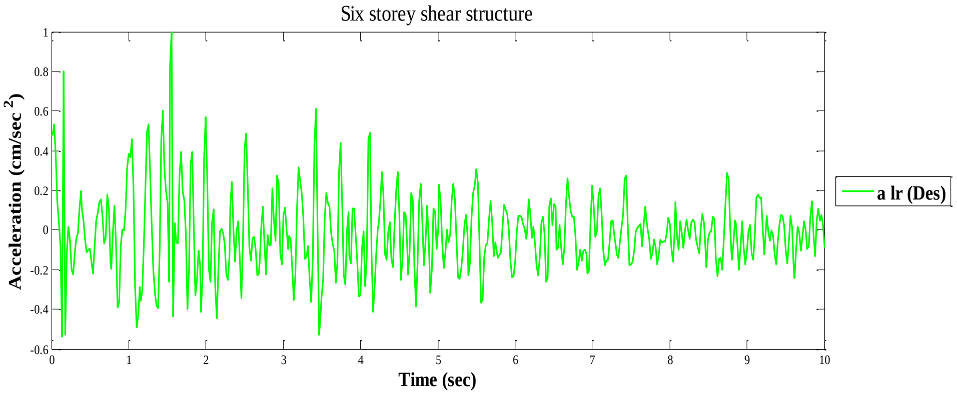

Figure 5.27(a): Lower values response of Chamoli earthquake at Barkot (NE) within [-1, 1] range

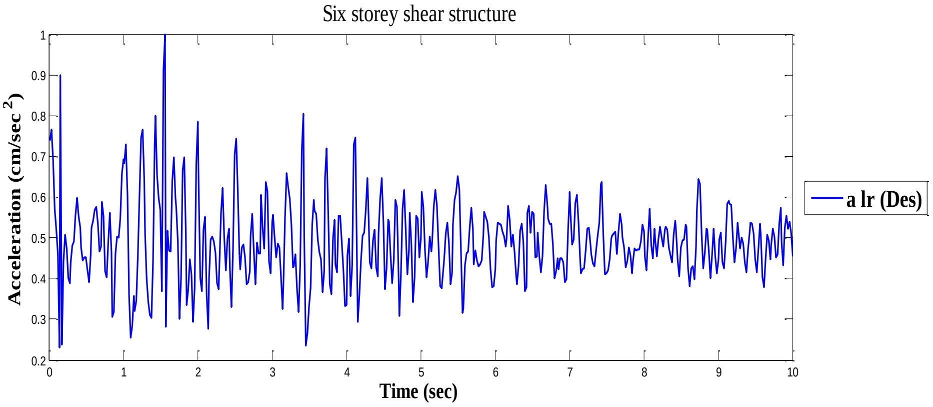

Figure 5.27(b): Conversion of lower values response of Chamoli earthquake at Barkot (NE) within [0, 1] range by transformation

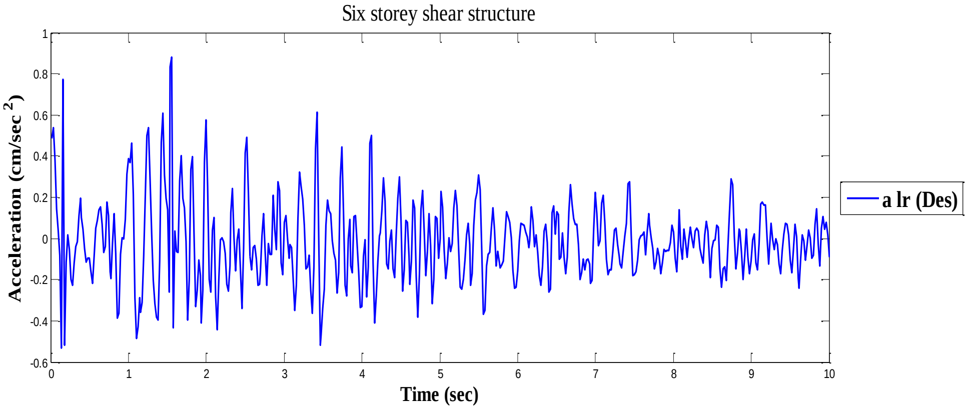

Figure 5.27(c): Conversion of lower values response of Chamoli earthquake at Barkot (NE) within [-1, 1] range by inverse transformation

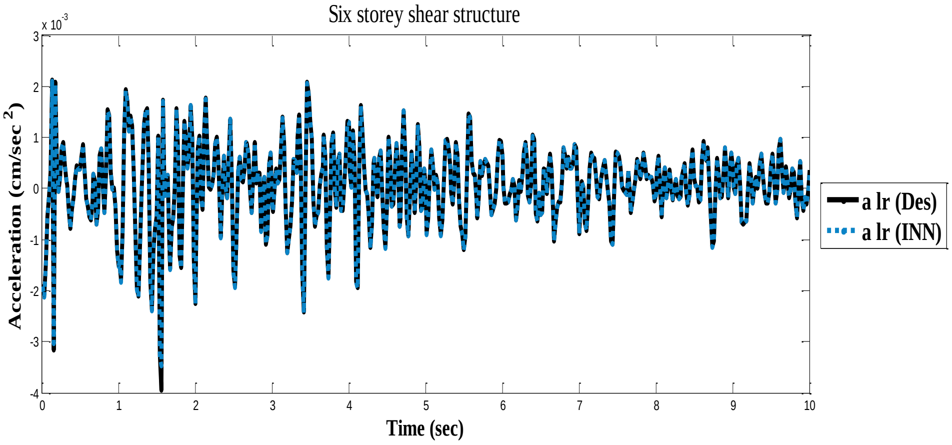

Figure 5.27(d): Comparison between the desired and INN response for lower values of Chamoli earthquake at Barkot (NE) for sixth storey

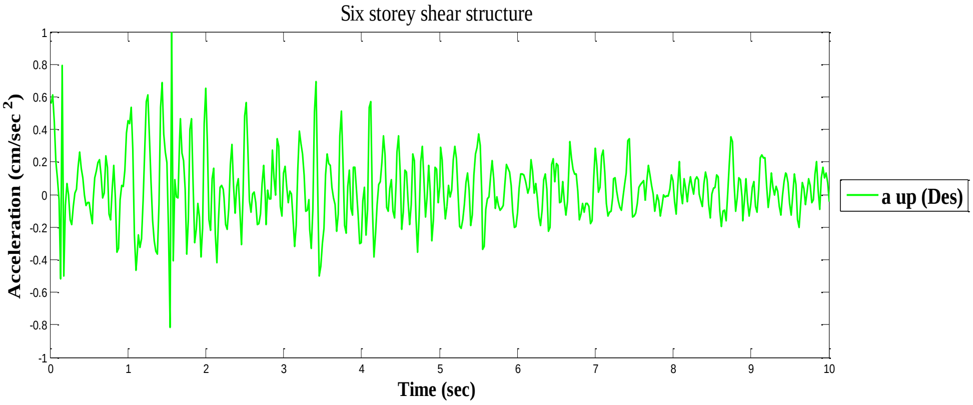

Figure 5.27(e): Upper values response of Chamoli earthquake at Barkot (NE) within [-1, 1] range

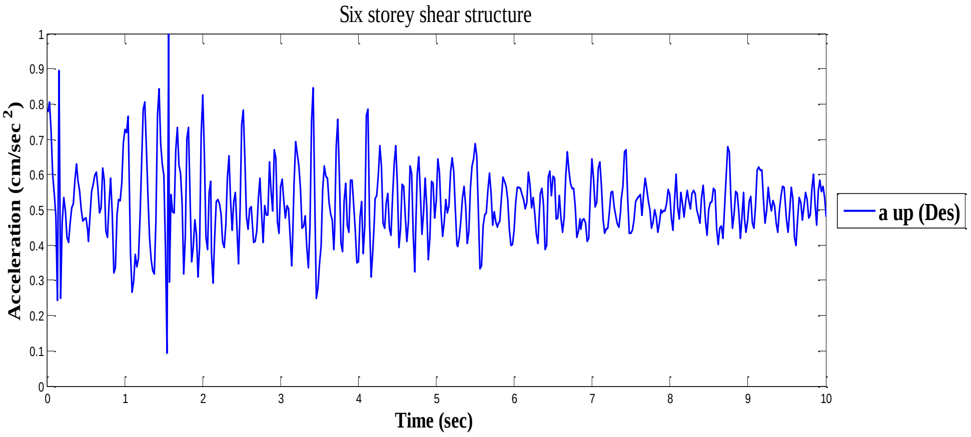

Figure 5.27(f): Conversion of upper values response of Chamoli earthquake at Barkot (NE) within [0, 1] range by transformation

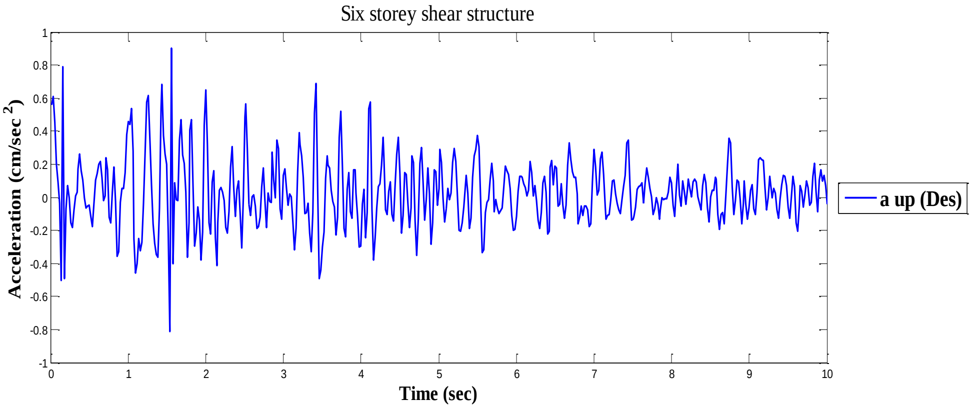

Figure 5.27(g): Conversion of upper values response of Chamoli earthquake at Barkot (NE) within [-1, 1] range by inverse transformation

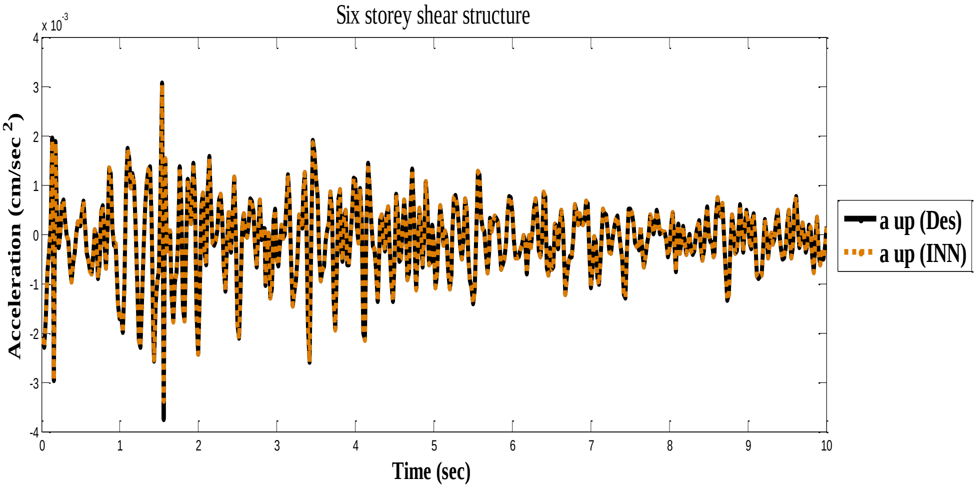

Figure 5.27(h): Comparison between the desired and INN response for upper values of Chamoli earthquake at Barkot (NE) for sixth storey

Example 3. Ten storey shear buildings:

The damping ratio in interval form is assumed here as [8%, 10%] critical for all natural modes for the entire storey. In the similar manner using same earthquake data as in examples 1 and 2, the training has been done for ten storey shear building and the converged weights are stored for testing. The neural and desired results obtained after training for tenth-storey of the structure are shown in Figures 5.28(a)-5.28(h). Figure 5.28(a) shows the data of lower case in [-1, 1] range, 5.28(b) shows conversion of the same data from [-1, 1] to [0, 1] using the transformation, 5.28(c) shows the conversion of data from [0, 1] to [-1, 1] using the inverse transformation. Finally the result comparison between the desired and INN is shown in Figure 5.28(d). For upper case, the figures are plotted in Figures 5.28(e)-5.28(h).

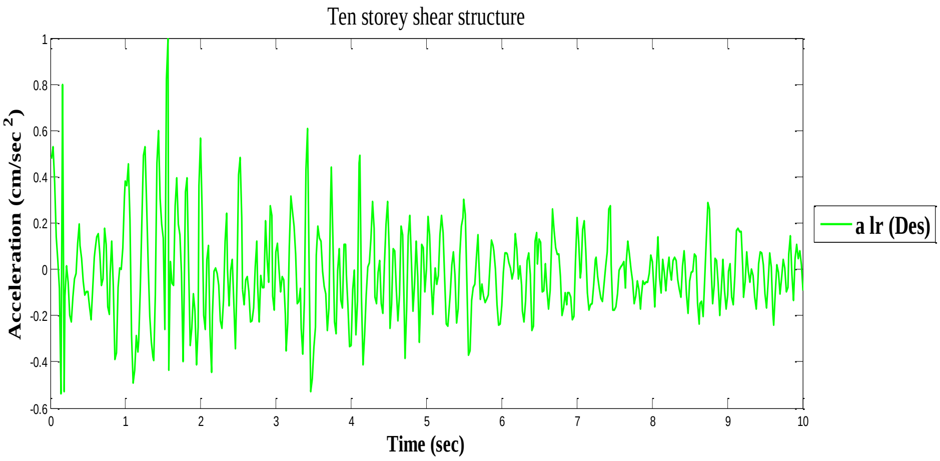

Figure 5.28(a): Lower values response of Chamoli earthquake at Barkot (NE) within [-1, 1] range

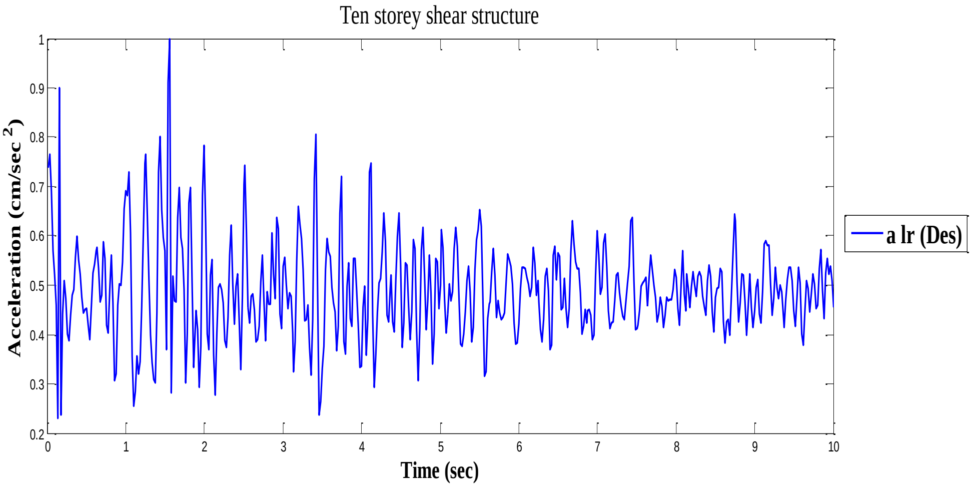

Figure 5.28(b): Conversion of lower values response of Chamoli earthquake at Barkot (NE) within [0, 1] range by transformation

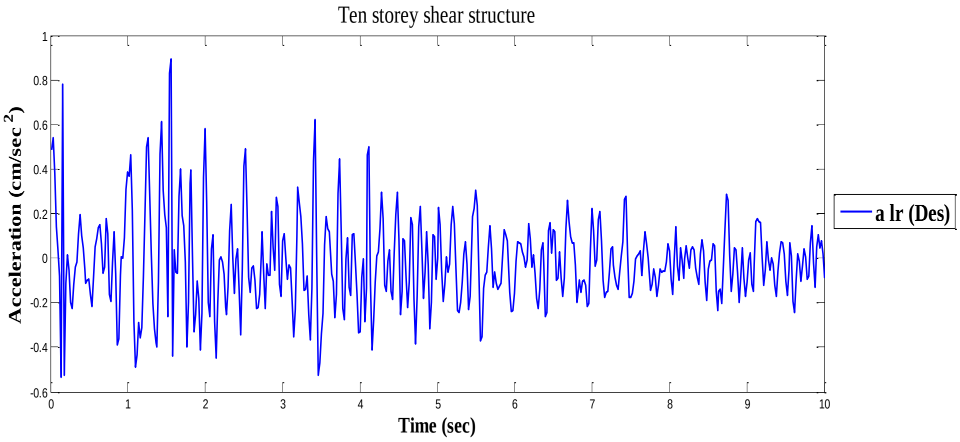

Figure 5.28(c): Conversion of lower values response of Chamoli earthquake at Barkot (NE) within [-1, 1] range by inverse transformation

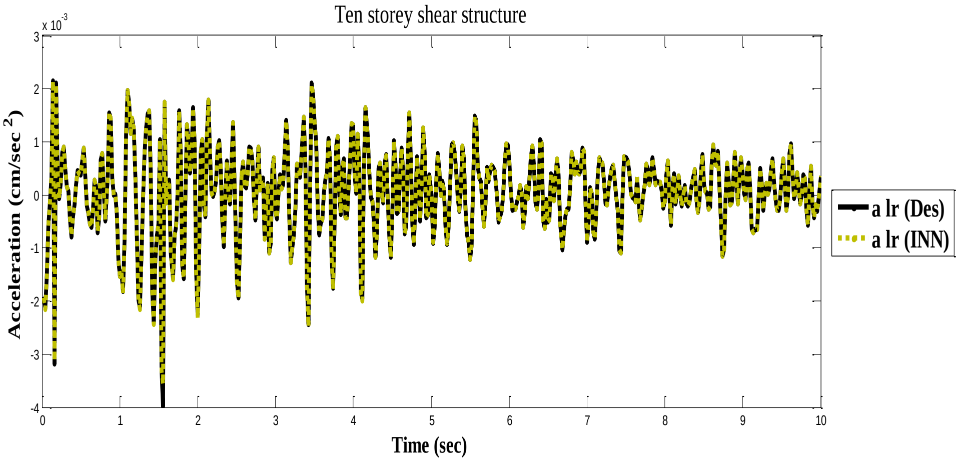

Figure 5.28(d): Comparison between the desired and INN response for lower values of Chamoli earthquake at Barkot (NE) for tenth storey

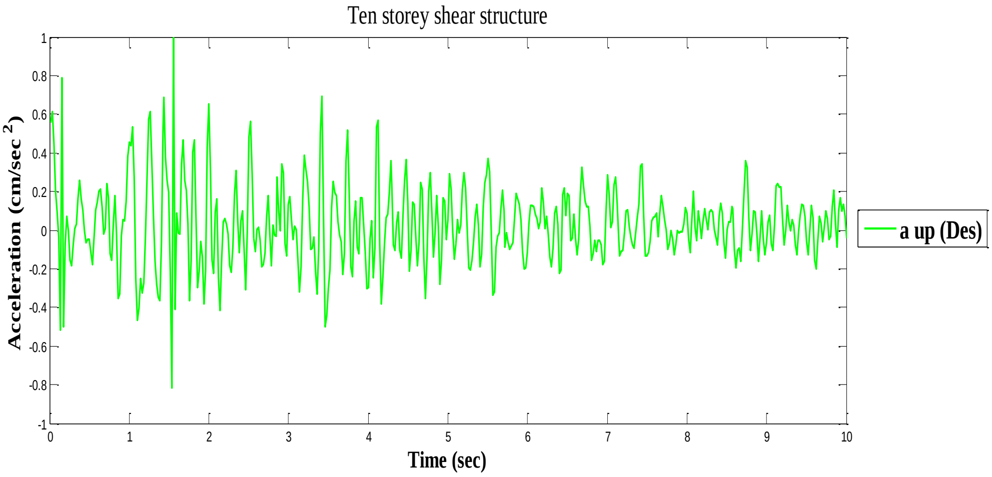

Figure 5.28(e): Upper values response of Chamoli earthquake at Barkot (NE) within [-1, 1] range

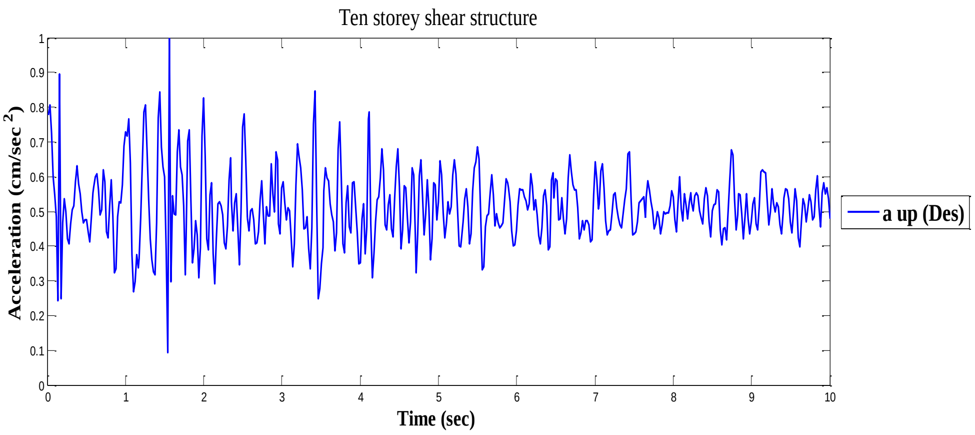

Figure 5.28(f): Conversion of upper values response of Chamoli earthquake at Barkot (NE) within [0, 1] range by transformation

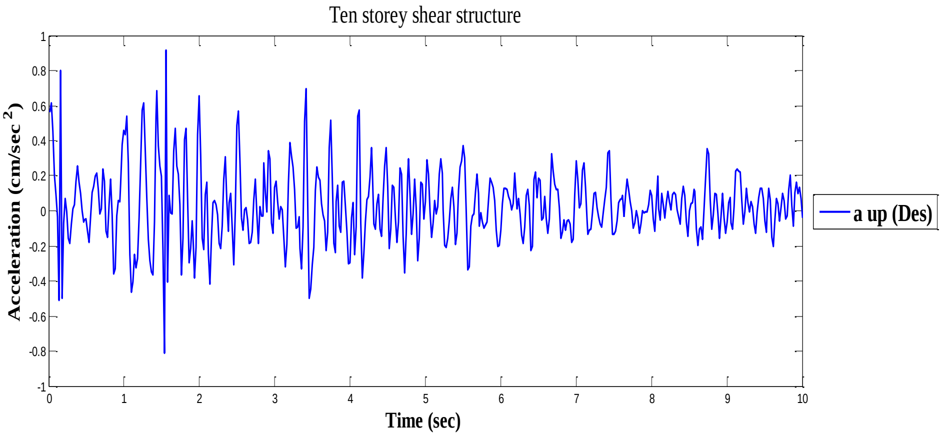

Figure 5.28(g): Conversion of upper values response of Chamoli earthquake at Barkot (NE) within [-1, 1] range by inverse transformation

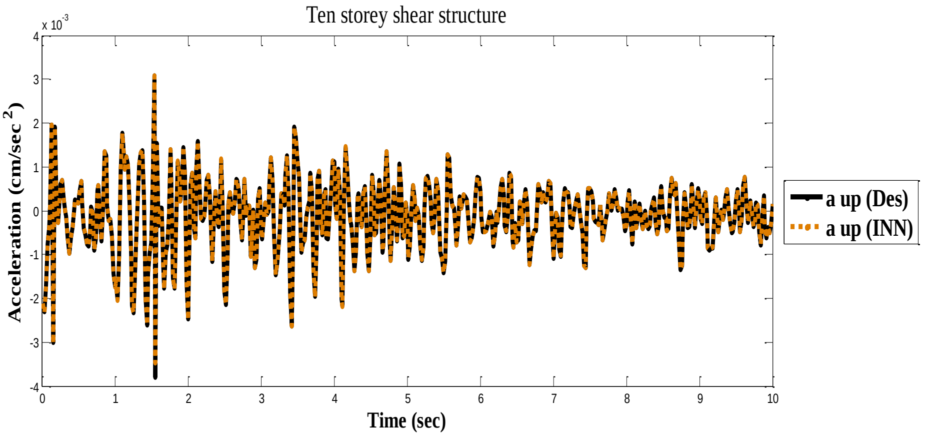

Figure 5.28(h): Comparison between the desired and INN response for upper values of Chamoli earthquake at Barkot (NE) for tenth storey

Example for testing case:

Next, the stored converged weights are used to predict storey responses with different intensity of earthquakes. The stored weights of problem 2 are used to predict fourth storey responses of the sixth storey for 80% and 120% of Uttarkashi earthquake at Barkot (NE). The comparison of the neural and desired responses for 80% is shown in Figures 5.29(a)-5.29(b) and for 120% is plotted in Figures 5.30(a)-5.30(b). Similarly, the ninth-storey responses of tenth storey for 80% and 120% of Uttarkashi earthquake at Barkot (NE) are found using the converged weights of problem 3. The neural and desired result for 80% is shown in Figures 5.31(a)-5.31(b) and for 120% is plotted in Figures 5.32(a)-5.32(b). The error % between desired and INN peak acceleration values (testing) with various intensities of Uttarkashi earthquake acceleration at Barkot (NE) have been presented in Table 5.4.

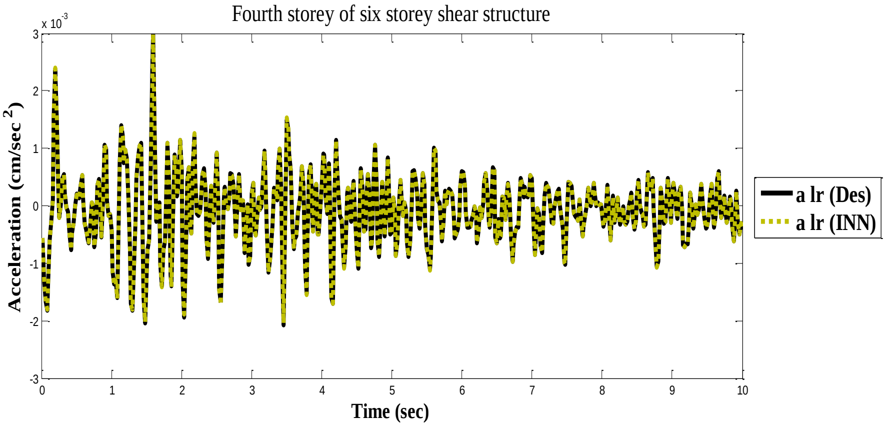

Figure 5.29(a): Lower values response comparison between the desired and INN for 80% of Uttarkashi earthquake at Barkot (NE) of fourth storey

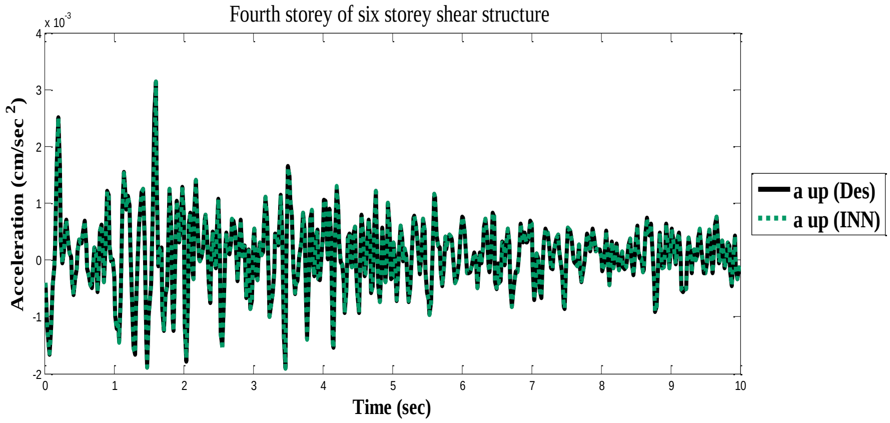

Figure 5.29(b): Upper values response comparison between the desired and INN for 80% of Uttarkashi earthquake at Barkot (NE) of fourth storey

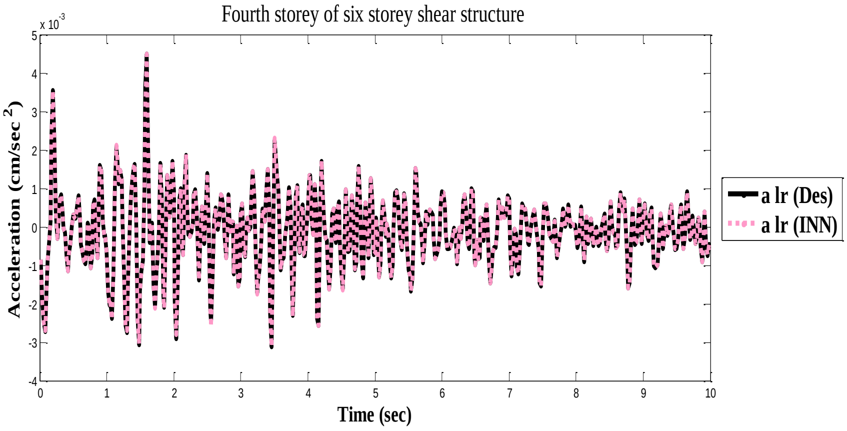

Figure 5.30(a): Lower values response comparison between the desired and INN for 120% of Uttarkashi earthquake at Barkot (NE) of fourth storey

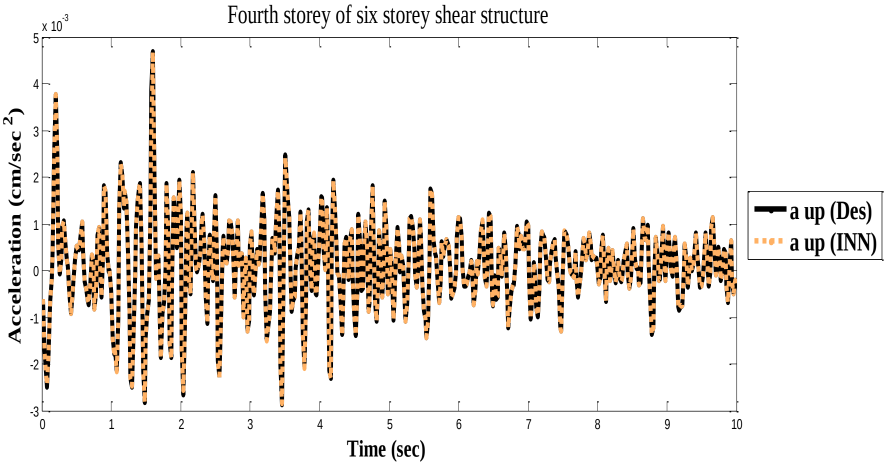

Figure 5.30(b): Upper values response comparison between the desired and INN for 120% of Uttarkashi earthquake at Barkot (NE) of fourth storey

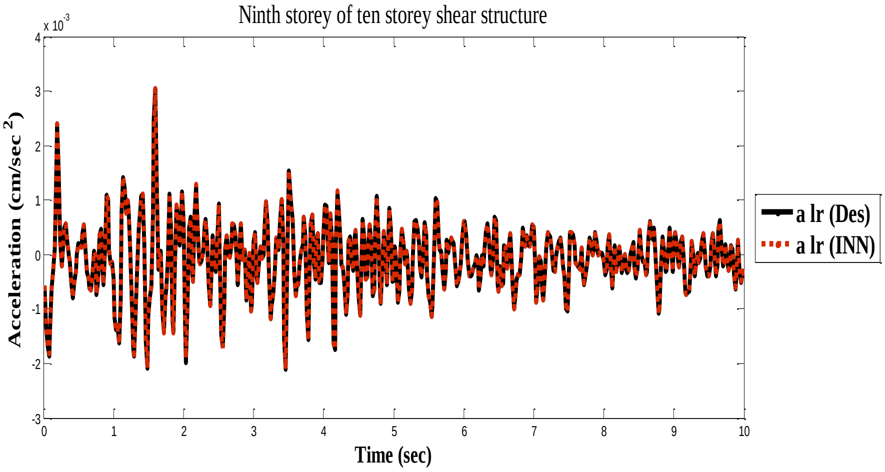

Figure 5.31(a): Lower values response comparison between the desired and INN for 80% of Uttarkashi earthquake at Barkot (NE) of ninth storey

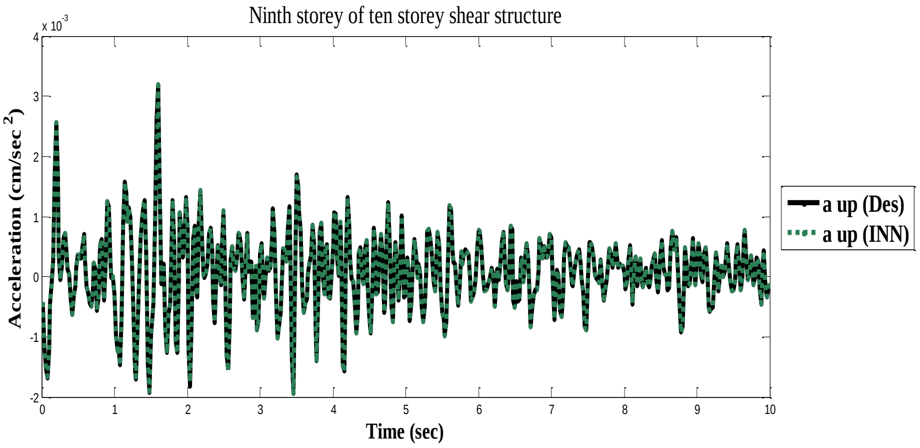

Figure 5.31(b): Upper values response comparison between the desired and INN for 80% of Uttarkashi earthquake at Barkot (NE) of ninth storey

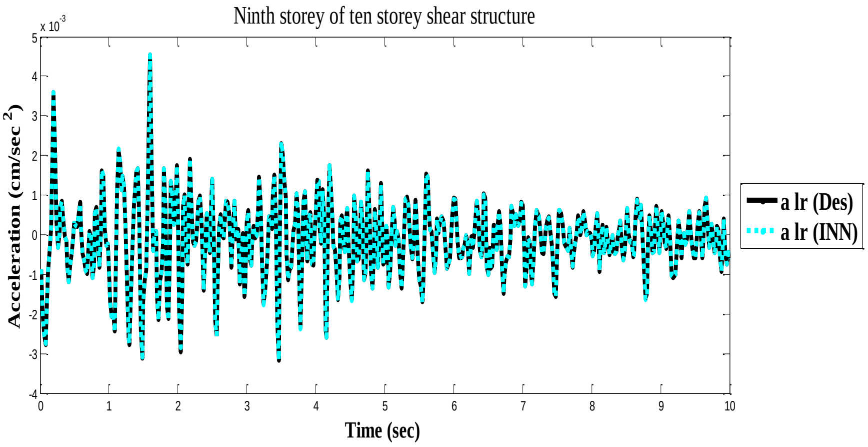

Figure 5.32(a): Lower values response comparison between the desired and INN for 120% of Uttarkashi earthquake at Barkot (NE) of ninth storey

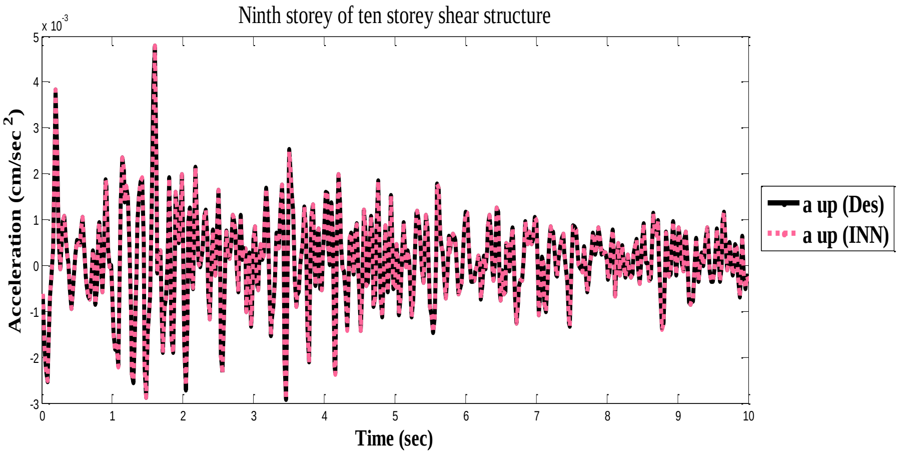

Figure 5.32(b): Upper values response comparison between the desired and INN for 120% of Uttarkashi earthquake at Barkot (NE) of ninth storey

The investigation done demonstrates that the developed model can handle the data in interval form. It is easier to use because training by one earthquake data, can very well give an idea of the response prediction subject to other earthquakes. Moreover, the model gives a faster prediction of the responses once it is trained by known earthquake data.

Table 5.4: Comparison between the Desired and INN Peak Acceleration values (Testing Values)

Storey	Intensities	Lower Peak value (cm/sec^2)			Upper Peak value (cm/sec^2)		
		Desired	INN	Error %	Desired	INN	Error %
4th storey (Uttarkashi at Barkot NE)	80%	0.0029843	0.0029884	-0.001	0.0031291	0.0031282	0.000
	120%	0.0044764	0.0044761	0.000	0.0046937	0.0046924	0.000
9th storey (Uttarkashi at Barkot NE)	80%	0.0030329	0.0030333	-0.001	0.0031929	0.0031974	-0.000
	120%	0.0045493	0.0045449	0.000	0.0047894	0.0047811	0.001

5.11 Conclusion

A method has been proposed here by creating a trained black box in terms of INN and FNN containing the characteristics of the multi-storey structure and of the earthquake motion. It is worth mentioning that the uncertain form of data in terms of interval and fuzzy shows the actual essence of the practical problem undertaken. After training of the INN and FNN model, it is shown that the black box can very well predict the dynamic response in interval form for other earthquakes that are not used in the training. The difficulty faced in the present problem is when the data are in bipolar form. The interval and fuzzy model cannot handle the negative data easily due to it computational complexities. So as to avoid these complexities first the fuzzified data are converted to *h*-level form, then a transformation is used which changes these bipolar data to unipolar form. One may note that while doing the training, the bipolar data in [-1, 1] are converted first to unipolar form [0, 1] by means of a transformation and once the training is done again the unipolar data are converted back to its bipolar form by using the inverse transformation. The trained INN and FNN architecture is then used to simulate and test the structural response of different floors for various intensity earthquake data. As mentioned in the previous sections that although the simulation is done with numerically generated response data for particular earthquake (experimental) data but the idea may also be used for actual experimental data of the building response. So, by using the input and output as the ground motion and the floor response, one can train the model. Accordingly then the storey response may be predicted for future earthquakes using the trained model. In this way the safety of the structural systems may be predicted in term of interval and fuzzy bound in case of future earthquakes.

Chapter 6

System Identification by Cluster of Interval and Fuzzy Neural Network

In this Chapter interval and fuzzy neural network based modelling for the identification of structural parameters of uncertain multi-storey shear buildings has been proposed. Here the method is developed to identify uncertain structural mass, stiffness and damping matrices from the dynamic responses of the structure without any optimization processes that are generally used to solve inverse vibration problems. Uncertainty has been taken in term of interval and fuzzy numbers. The governing equations of motion are first solved by the classical method to get responses of the consecutive stories. Further the governing equations of motion are modified based on relative responses of consecutive stories in such a way that the new set of equations can be implemented in a cluster of Interval and Fuzzy Neural Networks. As such the model starts solving the nth floor by INN and FNN modelling to estimate the structural parameters. Subsequently series of INN and FNN models are used to estimate the parameters for (n-1)th storey to the first storey. One may note that single layer interval and fuzzy neural networks have been used for training for each cluster of the INN and FNN such that the converged weights give the uncertain structural parameters. The initial weights in the INN and FNN architecture are taken as the design parameters in uncertain (interval and fuzzy) form. In order to validate the present model various example problems of different multi-storey shear structures have been considered. Related results are incorporated in term of tables and graphs. Comparisons between theoretical and identified results are carried out and are found to be in good agreement.

6.1 System Identification of Structural Parameters in Interval Form

We have consider a shear building with n storey structural system governed by the following set of linear differential equations in interval form as

$$[\tilde{M}]\{\tilde{\ddot{Y}}\}_t + [\tilde{C}]\{\tilde{\dot{Y}}\}_t + [\tilde{K}]\{\tilde{Y}\}_t = \{\tilde{F}\}_t \tag{6.1}$$

where $\{\ddot{\tilde{Y}}\}_t$ and $\{\dot{\tilde{Y}}\}_t$ are the known acceleration and velocity vectors in interval form respectively. Moreover, $\{\tilde{M}\} = \left[\underline{M}\,,\overline{M}\right]$ is $n \times n$ interval mass matrix of the structure and is given by

$$\{\tilde{M}\} = \begin{bmatrix} \left[\underline{m}_1\,,\overline{m}_1\right] & 0 & \cdots & \cdots & 0 \\ 0 & \left[\underline{m}_2\,,\overline{m}_2\right] & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & 0 & \left[\underline{m}_{n-1}\,,\overline{m}_{n-1}\right] & 0 \\ 0 & \cdots & \cdots & 0 & \left[\underline{m}_n\,,\overline{m}_n\right] \end{bmatrix},$$

$\{\tilde{C}\} = \left[\underline{C}\,,\overline{C}\right]$ represents $n \times n$ damping matrix of the structure in interval form and is written as

$$\{\tilde{C}\} = \begin{bmatrix} \left[\underline{c}_1,\overline{c}_1\right] + \left[\underline{c}_2\,,\overline{c}_2\right] & -\left[\underline{c}_2\,,\overline{c}_2\right] & 0 & \cdots & 0 \\ -\left[\underline{c}_2\,,\overline{c}_2\right] & \left[\underline{c}_2\,,\overline{c}_2\right] + \left[\underline{c}_3\,,\overline{c}_3\right] & -\left[\underline{c}_3\,,\overline{c}_3\right] & \cdots & 0 \\ \cdots & \cdots & & & \cdots \\ 0 & \cdots & -\left[\underline{c}_{n-1}\,,\overline{c}_{n-1}\right] & \left[\underline{c}_{n-1}\,,\overline{c}_{n-1}\right] + \left[\underline{c}_n\,,\overline{c}_n\right] & -\left[\underline{c}_n\,,\overline{c}_n\right] \\ 0 & & \cdots & -\left[\underline{c}_n,\overline{c}_n\right] & \left[\underline{c}_n\,,\overline{c}_n\right] \end{bmatrix},$$

and $\{\tilde{K}\} = \left[\underline{K}\,,\overline{K}\right]$ is $n \times n$ stiffness matrix of the structure in interval form which may be obtained as

$$\{\tilde{K}\} = \begin{bmatrix} \left[\underline{k}_1,\overline{k}_1\right] + \left[\underline{k}_2\,,\overline{k}_2\right] & -\left[\underline{k}_2\,,\overline{k}_2\right] & 0 & \cdots & 0 \\ -\left[\underline{k}_2\,,\overline{k}_2\right] & \left[\underline{k}_2\,,\overline{k}_2\right] + \left[\underline{k}_3\,,\overline{k}_3\right] & -\left[\underline{k}_3\,,\overline{k}_3\right] & \cdots & 0 \\ \cdots & \cdots & & & \cdots \\ 0 & \cdots & -\left[\underline{k}_{n-1}\,,\overline{k}_{n-1}\right] & \left[\underline{k}_{n-1}\,,\overline{k}_{n-1}\right] + \left[\underline{k}_n\,,\overline{k}_n\right] & -\left[\underline{k}_n\,,\overline{k}_n\right] \\ 0 & & \cdots & -\left[\underline{k}_n\,,\overline{k}_n\right] & \left[\underline{k}_n\,,\overline{k}_n\right] \end{bmatrix}.$$

The solution of free vibration equation i.e. for Eq. (6.1) with uncertain (interval) mass and stiffness gives the corresponding interval eigenvalues and eigenvectors Chakraverty [200]. The eigen values and eigen vectors are denoted by $\tilde{\lambda}_i$ and $\{\tilde{A}\}_i = \{\underline{A},\overline{A}\}_i$, $i = 1,\ldots, n$ respectively where $\tilde{\omega}_i^2 \left(= \tilde{\lambda}_i\right)$ are the system's interval natural frequency. The above free vibration equation will be an interval eigenvalue problem. The interval eigenvalue and vector are obtained by considering different sets of lower and upper stiffness and mass values. Although there exist different methods to handle interval eigenvalue problems but here the above procedure has been followed to handle the inverse of the matrices in crisp form separately as lower and upper values. And that is why now we will replace the '~' from all notations and will consider the case for lower form first and similarly for upper form. Hence the modal matrix for lower form $\{\underline{A}\}$ is written as

$$[\,\underline{A}\,]=[\,\{\,\underline{A}\,\}_1 \quad \{\,\underline{A}\,\}_2 \quad \cdots \quad \{\,\underline{A}\,\}_n\,]$$

The diagonal matrix consisting of eigenvalues in lower form is denoted as $\underline{\lambda}_i$, as $[\,\underline{\lambda}\,]_{n\times n}$, a new set of co-ordinates in lower form $\{\,\underline{x}\,\}$ related to the co-ordinates $\{\,\underline{Y}\,\}$ is introduced by the well known transformation

$$\{\underline{Y}\}=[\,\underline{A}\,]\;\{\,\underline{x}\,\}.$$

Proceeding by transforming into normal coordinates $(\{\underline{Y}\}=[\underline{A}]\{\underline{x}\})$ and premultiplying Eq. (6.1) with $[\underline{A}]^T$ we get

$$[\underline{A}]^T[\underline{M}][\underline{A}]\{\underline{\ddot{x}}\}+[\underline{A}]^T[\underline{C}][\underline{A}]\{\underline{\dot{x}}\}+[\underline{A}]^T[\underline{K}][\underline{A}]\{\underline{x}\}=[\underline{A}]^T\{\underline{F}\} \qquad (6.2)$$

Rewriting Eq. (6.2) in terms of generalized mass and spectral matrices, we obtain

$$[\underline{P}]\{\underline{\ddot{x}}\}+[\underline{A}]^T[\underline{C}][\underline{A}]\{\underline{\dot{x}}\}+[\underline{S}]\{\underline{x}\}=[\underline{A}]^T\{\underline{F}\} \qquad (6.3)$$

where $[\underline{P}]=[\underline{A}]^T[\underline{M}][\underline{A}]$ and $[\underline{S}]=[\underline{A}]^T[\underline{K}][\underline{A}]$

Thus we get the uncoupled equation as

$$\underline{P}_i\,\underline{\ddot{x}}_i+\underline{\overline{C}}_i\,\underline{\dot{x}}_i+\underline{S}_i\,\underline{x}_i=\underline{\overline{F}}_i \quad \text{for } i=1,2,....,n \qquad (6.4)$$

where $[\underline{A}]^T[\underline{C}][\underline{A}]=[\underline{\overline{C}}]$ and $[\underline{A}]^T\{\underline{F}\}=\{\underline{\overline{F}}\}$

The final solution may be obtained by solving the above differential equations and is written in the form

$$(\{\underline{Y}\}=[\underline{A}]\{\underline{x}\})$$

Similarly we can get the solution for upper form too. After getting the solution using the above classical method, Eq. (6.1) is rewritten to get the following set of equations in interval form

$$\tilde{m}_1\ddot{\tilde{y}}_1 + (\tilde{c}_1 + \tilde{c}_2)\dot{\tilde{y}}_1 - \tilde{c}_2\dot{\tilde{y}}_2 + (\tilde{k}_1 + \tilde{k}_2)\tilde{y}_1 - \tilde{k}_2\tilde{y}_2 = \tilde{f}_1 \tag{6.5}$$

$$\tilde{m}_2\ddot{\tilde{y}}_2 - \tilde{c}_2\dot{\tilde{y}}_1 + (\tilde{c}_2 + \tilde{c}_3)\dot{\tilde{y}}_2 - \tilde{c}_3\dot{\tilde{y}}_3 - \tilde{k}_2\tilde{y}_1 + (\tilde{k}_2 + \tilde{k}_3)\tilde{y}_2 - \tilde{k}_3\tilde{y}_3 = \tilde{f}_2 \tag{6.6}$$

$$\vdots$$

$$\vdots$$

$$\begin{aligned}&\tilde{m}_{n-1}\ddot{\tilde{y}}_{n-1} - \tilde{c}_{n-1}\dot{\tilde{y}}_{n-2} + (\tilde{c}_{n-1} + \tilde{c}_n)\dot{\tilde{y}}_{n-1} - \tilde{c}_n\dot{\tilde{y}}_n - \\ &\tilde{k}_{n-1}\tilde{y}_{n-2} + (\tilde{k}_{n-1} + \tilde{k}_n)\tilde{y}_{n-1} - \tilde{k}_n\tilde{y}_n = \tilde{f}_{n-1}\end{aligned} \tag{6.7}$$

$$\tilde{m}_n\ddot{\tilde{y}}_n - \tilde{c}_n\dot{\tilde{y}}_{n-1} + \tilde{c}_n\dot{\tilde{y}}_n - k_n\tilde{y}_{n-1} + \tilde{k}_n\tilde{y}_n = \tilde{f}_n \tag{6.8}$$

Eq. (6.8) is then written as

$$\tilde{m}_n\ddot{\tilde{y}}_n + \tilde{c}_n\left[\dot{\tilde{y}}_n - \dot{\tilde{y}}_{n-1}\right] + \tilde{k}_n\left[\tilde{y}_n - \tilde{y}_{n-1}\right] = \tilde{f}_n \tag{6.9}$$

The above equation may now be presented as

$$\tilde{m}_n\ddot{\tilde{y}}_n + \tilde{c}_n\dot{\tilde{d}}_n + \tilde{k}_n\tilde{d}_n = \tilde{f}_n \tag{6.10}$$

where $\dot{\tilde{d}}_n = \left[\dot{\tilde{y}}_n - \dot{\tilde{y}}_{n-1}\right]$ and $\tilde{d}_n = \left[\tilde{y}_n - \tilde{y}_{n-1}\right]$

Here $\dot{\tilde{d}}$ and $\tilde{d}_n$ defines the known relative velocity and displacement in interval form for n^{th} storey which is calculated by the above classical method. Single layer interval neural network is used to solve Eq. (6.10). For this interval neural network, the inputs are taken as structural acceleration, relative velocity and relative displacement for the n^{th} storey and outputs are taken as the applied force at time *t*. Using this single layer interval neural network, Eq. (6.10) is solved by a continuous training process with *n* training patterns and the converged weight matrix are obtained. This converged weight matrix gives the corresponding physical parameters such as $\tilde{m}_n, \tilde{c}_n$ and $\tilde{k}_n$ in interval form. Identified parameters of n^{th} storey are then used to identify the parameters of $(n-1)^{\text{th}}$ storey using Eq. (6.7). Proceeding in the same manner the parameters of $(n-2)^{\text{th}}$ storey are determined and the process goes on till the unknown parameters for the first storey are obtained. The cluster of interval neural network diagram for n storey structure is shown in Figure 6.1.

6.2 Learning Algorithm for Single Layer Interval Neural Network

A Neural Network in which at least one of its input, output and weight have interval values then the network is said to be an Interval Neural Network, Escarcina et al. [106]. Interval Neural Network (INN) is formed by processing units called interval neurons. In Interval Neural Networks, neurons are connected in a similar way as they are connected in traditional Neural Networks. The structure of a typical single layer INN is shown in Figure 6.2. The stepwise algorithm for the present INN model with interval computation (defined above) is shown below:

Step 1: The input weights $(\tilde{W}_{ij})$ and bias weights $\tilde{\theta}_i$ in interval form are initialized.

Step 2: The training pairs are considered in the form of $\{\tilde{Z}_1, \tilde{d}_1; \tilde{Z}_2, \tilde{d}_2; \ldots. \tilde{Z}_I, \tilde{d}_I\}$ where

$\tilde{Z}_I = ([\underline{z}_1, \overline{z}_1], [\underline{z}_2, \overline{z}_2], .., [\underline{z}_n, \overline{z}_n])$ are the inputs and

$\tilde{d}_I = ([\underline{d}_1, \overline{d}_1], [\underline{d}_2, \overline{d}_2], ..., [\underline{d}_n, \overline{d}_n])$ are the desired values for the given inputs in interval form.

Step 3: The output of the network is calculated for the input $\tilde{Z}_I$ as

$$\tilde{O}_J = [f(\underline{Y}_J), f(\overline{Y}_J)]$$

where $[\underline{Y}_J, \overline{Y}_J] = [\underline{W}_{ij}, \overline{W}_{ij}].[\underline{Z}_I, \overline{Z}_I] + [\underline{\theta}_i, \overline{\theta}_i]$

and f is the unipolar activation function defined by $f(\tilde{Y}_J) = 1 \big/ [1 + \exp(-\gamma \tilde{Y}_J)]$

Step 4: The weight is modified as

$$\tilde{W}_{ij}^{(New)} = \left[\underline{W}_{ij}^{(New)}, \overline{W}_{ij}^{(New)} \right] = \left[\underline{W}_{ij}^{(Old)}, \overline{W}_{ij}^{(Old)} \right] + \left[\Delta \underline{W}_{ij}, \Delta \overline{W}_{ij} \right]$$

where change in weights are calculated as

$$\Delta \tilde{W}_{ij} = [\Delta \underline{W}_{ij}, \Delta \overline{W}_{ij}] = \left[-\eta \frac{\partial \tilde{E}}{\partial \underline{W}_{ij}}, -\eta \frac{\partial \tilde{E}}{\partial \overline{W}_{ij}} \right]$$

In the similar fashion the bias weights are also updated.

Step 5: The error value is computed as

$$\tilde{E} = \frac{1}{2}\left[\left(\underline{d}_J - \underline{O}_J\right)^2 + \left(\overline{d}_J - \overline{O}_J\right)^2 \right]$$

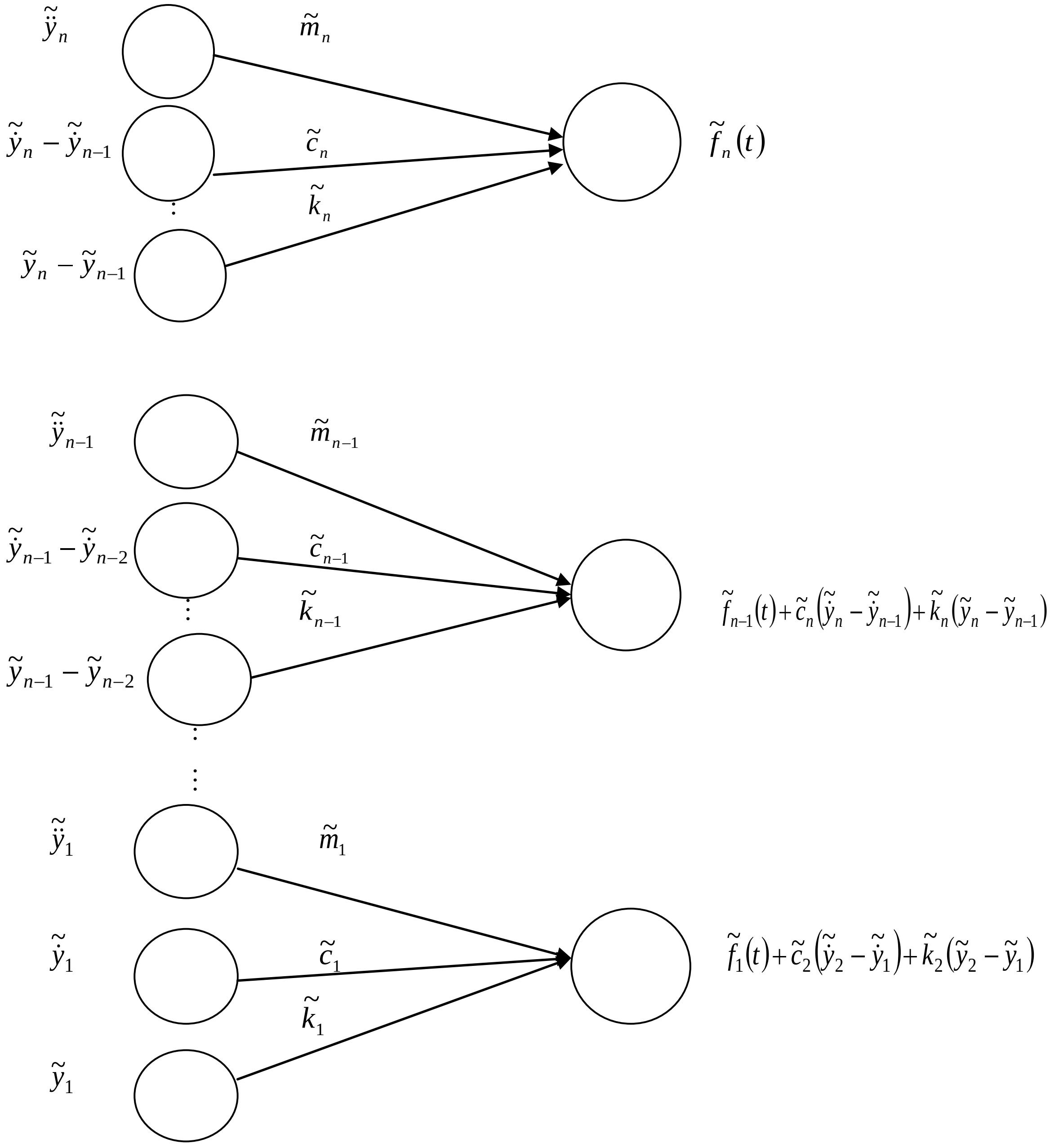
$\tilde{\ddot{y}}_n$
$\tilde{m}_n$
$\tilde{\dot{y}}_n - \tilde{\dot{y}}_{n-1}$
$\tilde{c}_n$
$\tilde{k}_n$
$\tilde{y}_n - \tilde{y}_{n-1}$
$\tilde{f}_n(t)$
$\tilde{\ddot{y}}_{n-1}$
$\tilde{m}_{n-1}$
$\tilde{\dot{y}}_{n-1} - \tilde{\dot{y}}_{n-2}$
$\tilde{c}_{n-1}$
$\tilde{k}_{n-1}$
$\tilde{y}_{n-1} - \tilde{y}_{n-2}$
$\tilde{f}_{n-1}(t)+\tilde{c}_n\left(\tilde{\dot{y}}_n - \tilde{\dot{y}}_{n-1}\right)+\tilde{k}_n\left(\tilde{y}_n - \tilde{y}_{n-1}\right)$
$\tilde{\ddot{y}}_1$
$\tilde{m}_1$
$\tilde{\dot{y}}_1$
$\tilde{c}_1$
$\tilde{k}_1$
$\tilde{y}_1$
$\tilde{f}_1(t)+\tilde{c}_2\left(\tilde{\dot{y}}_2 - \tilde{\dot{y}}_1\right)+\tilde{k}_2\left(\tilde{y}_2 - \tilde{y}_1\right)$

Figure 6.1: Proposed Cluster of Interval Neural Network Model for n Storey Shear Structure

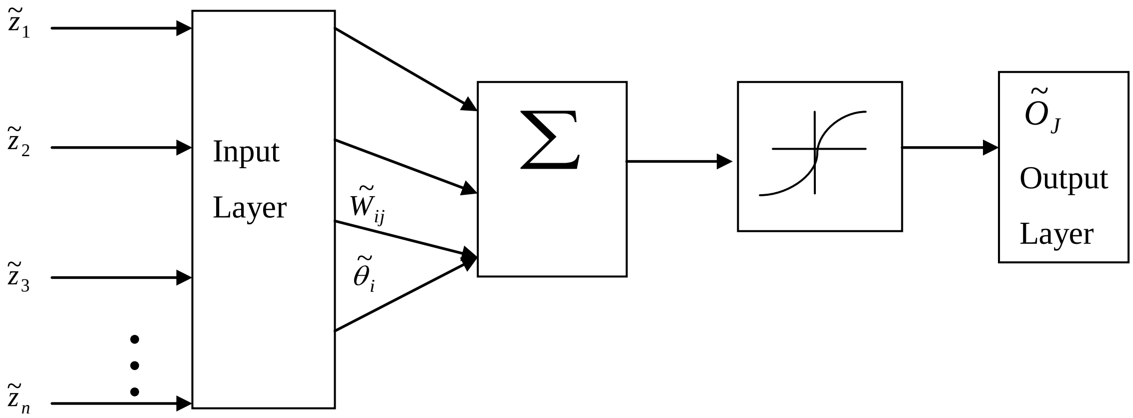

Figure 6.2: Single Layer Interval Neural Network

6.3 System Identification of Structural Parameters in Fuzzified Form

A shear building with n storey structural system governed by the following set of linear differential equations in fuzzified form is considered here and written as

$$[\hat{M}]\{\hat{\ddot{X}}\}_t + [\hat{C}]\{\hat{\tilde{X}}\}_t + [\hat{K}]\{\hat{X}\}_t = \{\hat{F}\}_t \tag{6.11}$$

where $\{\hat{X}\}_t = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, ..., \hat{x}_n\}_t^T$, $\{\hat{\ddot{X}}\}_t = \{\hat{\ddot{x}}_1, \hat{\ddot{x}}_2, \hat{\ddot{x}}_3, ..., \hat{\ddot{x}}_n\}_t^T$ and $\{\hat{\dot{X}}\}_t = \{\hat{\dot{x}}_1, \hat{\dot{x}}_2, \hat{\dot{x}}_3, ..., \hat{\dot{x}}_n\}_t^T$ indicates displacement, acceleration and velocity vectors of the consecutive n^{th} storey in fuzzified form respectively. Moreover, $\{\hat{M}\} = [\underline{M}, Mc, \overline{M}]$ is $n \times n$ mass matrix of the structure in fuzzified form and is given by

$$\{\hat{M}\} = \begin{bmatrix} [\underline{m}_1, m_1c, \overline{m}_1] & 0 & \cdots & \cdots & 0 \\ 0 & [\underline{m}_2, m_2c, \overline{m}_2] & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & 0 & [\underline{m}_{n-1}, m_{n-1}c, \overline{m}_{n-1}] & 0 \\ 0 & \cdots & \cdots & 0 & [\underline{m}_n, m_nc, \overline{m}_n] \end{bmatrix},$$

$\{\hat{C}\} = [\underline{C}, Cc, \overline{C}]$ represents $n \times n$ damping matrix of the structure in fuzzified form and is written as

$$\{\hat{C}\}=\begin{bmatrix} [\underline{c}_1, c_1, \bar{c}_1]+[\underline{c}_2, c_2c, \bar{c}_2] & -[\underline{c}_2, c_2c, \bar{c}_2] & 0 & \cdots & 0 \\ -[\underline{c}_2, c_2c, \bar{c}_2] & [\underline{c}_2, c_2c, \bar{c}_2]+[\underline{c}_3, c_3c, \bar{c}_3] & -[\underline{c}_3, c_3c, \bar{c}_3] & \cdots & 0 \\ \cdots & \cdots & & \cdots & \\ 0 & \cdots & -[\underline{c}_{n-1}, c_{n-1}c, \bar{c}_{n-1}] & [\underline{c}_{n-1}, c_{n-1}c, \bar{c}_{n-1}]+[\underline{c}_n, c_nc, \bar{c}_n] & -[\underline{c}_n, c_nc, \bar{c}_n] \\ 0 & & \cdots & -[\underline{c}_n, c_nc, \bar{c}_n] & [\underline{c}_n, c_nc, \bar{c}_n] \end{bmatrix},$$

and $\{\hat{K}\}=\left[\underline{K}, Kc, \overline{K}\right]$ is $n\times n$ stiffness matrix of the structure in fuzzified form which may be obtained as

$$\{\hat{K}\}=\begin{bmatrix} [\underline{k}_1, k_1c, \bar{k}_1]+[\underline{k}_2, k_2c, \bar{k}_2] & -[\underline{k}_2, k_2c, \bar{k}_2] & 0 & \cdots & 0 \\ -[\underline{k}_2, k_2c, \bar{k}_2] & [\underline{k}_2, k_2c, \bar{k}_2]+[\underline{k}_3, k_3c, \bar{k}_3] & -[\underline{k}_3, k_3c, \bar{k}_3] & \cdots & 0 \\ \cdots & \cdots & & \cdots & \\ 0 & \cdots & -[\underline{k}_{n-1}, k_{n-1}c, \bar{k}_{n-1}] & [\underline{k}_{n-1}, k_{n-1}c, \bar{k}_{n-1}]+[\underline{k}_n, k_nc, \bar{k}_n] & -[\underline{k}_n, k_nc, \bar{k}_n] \\ 0 & & \cdots & -[\underline{k}_n, k_nc, \bar{k}_n] & [\underline{k}_n, k_nc, \bar{k}_n] \end{bmatrix}.$$

Solution for Eq. (6.11) for free vibration with given mass and stiffness values in fuzzified form gives the corresponding fuzzy eigenvalues and eigenvectors. These are denoted respectively by $\hat{\lambda}_i$ and $\{\hat{A}\}_i = \{\underline{A}, Ac, \overline{A}\}_i$, $i = 1,..., n$ where $\hat{\omega}_i^2 \left(= \hat{\lambda}_i\right)$ are the system's natural frequency in fuzzified form. It may be noted that the free vibration equation will be a fuzzy eigen value problem. The fuzzy eigen value and vector are obtained then by considering different sets of lower, centre and upper stiffness and mass values. Here the above procedure has been used so that we may handle the inverse of the matrices in crisp form separately as lower, centre and upper values. And that is why now we will replace the '^' from all notations and will consider the case for lower form first and similarly for centre and upper form. Hence the modal matrix Chakraverty [200] for lower form $\{\underline{A}\}$ may be written as

$$[\underline{A}]=[\{\underline{A}\}_1 \quad \{\underline{A}\}_2 \quad \cdots \quad \{\underline{A}\}_n]$$

Denoting the diagonal matrix made up of the eigenvalues in lower form as $\underline{\lambda}_i$, as $[\underline{\lambda}]_{n\times n}$, a new set of co-ordinates in lower form $\{\underline{y}\}$ related to the co-ordinates $\{\underline{X}\}$ is introduced by the well known transformation

$$\{\underline{X}\}=[\underline{A}]\ \{\underline{y}\}.$$

Proceeding by transforming into normal coordinates $\left(\{\underline{X}\}=[\underline{A}]\{\underline{y}\}\right)$ and premultiplying Eq. (6.11) with $[\underline{A}]^T$ we get

$$[\underline{A}]^T[\underline{M}][\underline{A}]\{\underline{\ddot{y}}\}+[\underline{A}]^T[\underline{C}][\underline{A}]\{\underline{\dot{y}}\}+[\underline{A}]^T[\underline{K}][\underline{A}]\{\underline{y}\}=[\underline{A}]^T\{\underline{F}\} \tag{6.12}$$

Rewriting Eq. (6.12) in terms of generalized mass and spectral matrices, we obtain

$$[\underline{P}]\{\underline{\ddot{y}}\}+[\underline{A}]^T[\underline{C}][\underline{A}]\{\underline{\dot{y}}\}+[\underline{S}]\{\underline{y}\}=[\underline{A}]^T\{\underline{F}\} \tag{6.13}$$

where $[\underline{P}]=[\underline{A}]^T[\underline{M}][\underline{A}]$ and $[\underline{S}]=[\underline{A}]^T[\underline{K}][\underline{A}]$

Thus we get the uncoupled equation as

$$\underline{P}_i\,\underline{\ddot{y}}_i+\underline{\overline{C}}_i\,\underline{\dot{y}}_i+\underline{S}_i\,\underline{y}_i=\underline{\overline{F}}_i \text{ for } i=1,2,....,n \tag{6.14}$$

where $[\underline{A}]^T[\underline{C}][\underline{A}]=[\underline{\overline{C}}]$ and $[\underline{A}]^T\{\underline{F}\}=\{\underline{\overline{F}}\}$

The above differential equations can be solved and the final solution may be written by

$$\left(\{\underline{X}\}=[\underline{A}]\{\underline{y}\}\right)$$

In the similar manner we can proceed for centre and upper form too. After finding out the solution, Eq. (6.11) is rewritten to get the following set of equations in fuzzified form

$$\hat{m}_1\ddot{\hat{x}}_1+(\hat{c}_1+\hat{c}_2)\,\dot{\hat{x}}_1-\hat{c}_2\dot{\hat{x}}_2+\left(\hat{k}_1+\hat{k}_2\right)\hat{x}_1-\hat{k}_2\,\hat{x}_2=\hat{f}_1 \tag{6.15}$$

$$\hat{m}_2\ddot{\hat{x}}_2-\hat{c}_2\,\dot{\hat{x}}_1+(\hat{c}_2+\hat{c}_3)\dot{\hat{x}}_2-\hat{c}_3\,\dot{\hat{x}}_3-\hat{k}_2\,\hat{x}_1+\left(\hat{k}_2+\hat{k}_3\right)\hat{x}_2-\hat{k}_3\,\hat{x}_3=\hat{f}_2 \tag{6.16}$$

$$\vdots$$
$$\vdots$$

$$\begin{aligned}&\hat{m}_{n-1}\ddot{\hat{x}}_{n-1}-\hat{c}_{n-1}\,\dot{\hat{x}}_{n-2}+(\hat{c}_{n-1}+\hat{c}_n)\dot{\hat{x}}_{n-1}-\hat{c}_n\,\dot{\hat{x}}_n-\\&\hat{k}_{n-1}\,\hat{x}_{n-2}+\left(\hat{k}_{n-1}+\hat{k}_n\right)\hat{x}_{n-1}-\hat{k}_n\,\hat{x}_n=\hat{f}_{n-1}\end{aligned} \tag{6.17}$$

$$\hat{m}_n\ddot{\hat{x}}_n-\hat{c}_n\,\dot{\hat{x}}_{n-1}+\hat{c}_n\,\dot{\hat{x}}_n-\hat{k}_n\,\hat{x}_{n-1}+\hat{k}_n\,\hat{x}_n=\hat{f}_n \tag{6.18}$$

Eq. (6.18) is then written as

$$\hat{m}_n\hat{\ddot{x}}_n + \hat{c}_n\left[\hat{\dot{x}}_n - \hat{\dot{x}}_{n-1}\right] + \hat{k}_n\left[\hat{x}_n - \hat{x}_{n-1}\right] = \hat{f}_n \tag{6.19}$$

The above equation may now be presented as

$$\hat{m}_n\hat{\ddot{x}}_n + \hat{c}_n\hat{\dot{d}}_n + \hat{k}_n\hat{d}_n = \hat{f}_n \tag{6.20}$$

where $\hat{\dot{d}}_n = \left[\hat{\dot{x}}_n - \hat{\dot{x}}_{n-1}\right]$ and $\hat{d}_n = \left[\hat{x}_n - \hat{x}_{n-1}\right]$

Here $\hat{\dot{d}}_n$ and $\hat{d}_n$ indicate the known relative velocity and displacement in fuzzified form for n^{th} storey. Using the single layer fuzzy neural network, Eq. (6.20) is solved. Inputs for this fuzzy neural network are taken as structural acceleration, relative velocity and relative displacement for the n^{th} storey. Output of the network is taken as the applied force at time *t*. To solve Eq. (6.20) a continuous training process with *n* training patterns are done using fuzzy neural network and the converged weight matrix of the neural network is thus obtained. From this weight matrix the corresponding physical parameters such as $\hat{m}_n, \hat{c}_n$ and $\hat{k}_n$ in fuzzified form are obtained. Identified parameters of n^{th} storey are then used to identify the parameters of $(n-1)^{\text{th}}$ storey using Eq. (6.17). Proceeding in the same manner the parameters of $(n-2)^{\text{th}}$ storey are determined and the process goes on till the unknown parameters of the first storey are obtained. The cluster of fuzzy neural network diagram for n storey structure is shown in Figure 6.3.

6.4 Learning Algorithm for Single Layer Fuzzy Neural Network

A Neural Network is said to be a Fuzzy Neural Network if at least one of its input, output and weight sets have values in fuzzified form. In fuzzy Neural Networks, neurons are connected as they are connected in traditional Neural Networks and a typical single layer FNN is shown in Figure 6.4. Following are the steps in FNN using the fuzzy computation defined above.

Step 1: Initialize input weights $\left(\hat{W}_{ji}\right)$ and bias weights $\hat{\theta}_i$ in fuzzified form.

Step 2: Present the training pairs in the form $\left\{\hat{Z}_1, \hat{d}_1; \hat{Z}_2, \hat{d}_2; .... \hat{Z}_I, \hat{d}_J\right\}$ where

$\hat{Z}_I = \left(\left[\underline{z}_1, z_1c, \overline{z}_1\right], \left[\underline{z}_2, z_2c, \overline{z}_2\right],.., \left[\underline{z}_n, z_nc, \overline{z}_n\right]\right)$ are inputs and

$\hat{d}_J = \left(\left[\underline{d}_1, d_1c, \overline{d}_1\right], \left[\underline{d}_2, d_2c, \overline{d}_2\right],..., \left[\underline{d}_n, d_nc, \overline{d}_n\right]\right)$ are desired values for the given inputs in fuzzified form. The inputs and desired values can be written in *h*-level form as $\hat{Z}_I = \left(\left[[\underline{z}_1]_h, [\overline{z}_1]_h\right], \left[[\underline{z}_2]_h, [\overline{z}]_h\right],.., \left[[\underline{z}_n]_h, [\overline{z}_n]_h\right]\right)$,

$\hat{d}_J = \left(\left[[\underline{d}_1]_h, [\overline{d}_1]_h\right], \left[[\underline{d}_2]_h, [\overline{d}_2]_h\right],..., \left[[\underline{d}_n]_h, [\overline{d}_n]_h\right]\right)$

Step 3: Calculate the output of the network for the input $\hat{z}_I$ in *h*-level form

$$[\hat{O}_J]_h = f\left([Ne\hat{t}_J]_h\right)$$

where $\left([Ne\hat{t}_J]_h\right) = \sum_{j=1}^{J} [\hat{W}_{ji}]_h [\hat{O}_I]_h + [\hat{\theta}_i]_h \quad i = 1,2,..., I \quad \text{and} \quad j = 1,2,..., J$

and $\hat{f}$ is the fuzzy unipolar activation function defined by

$$f\left([Ne\hat{t}_J]_h\right) = 1 \Big/ \left[1 + \exp\left(-\gamma\, [Ne\hat{t}_J]_h\right)\right]$$

Step 4: The error value is then computed as

$$\hat{E} = \frac{h}{2}\left[\left([\underline{d}_J]_h - [\underline{O}_J]_h\right)^2 + \left([\overline{d}_J]_h - [\overline{O}_J]_h\right)^2\right]$$

Step 5: The weight is modified as

$$[\hat{W}_{ji}]_h(New) = \left[[\underline{W}_{ji}]_h(New), [\overline{W}_{ji}]_h(New)\right] = \left[[\underline{W}_{ji}]_h(Old), [\overline{W}_{ji}]_h(Old)\right] + \left[[\Delta\underline{W}_{ji}]_h, [\Delta\overline{W}_{ji}]_h\right]$$

$j = 1,2,..., J \quad \text{and} \quad i = 1,2,..., I$

where change in weights are calculated as

$$[\Delta\hat{W}_{ji}]_h = \left[[\Delta\underline{W}_{ji}]_h, [\Delta\overline{W}_{ji}]_h\right] = \left[-\eta\frac{\partial\hat{E}}{\partial[\underline{W}_{ji}]_h}, -\eta\frac{\partial\hat{E}}{\partial[\overline{W}_{ji}]_h}\right] \quad j = 1,2,..., J \quad \text{and} \quad i = 1,2,..., I$$

In the similar fashion the bias weights $[\hat{\theta}_i]_h$ is also updated.

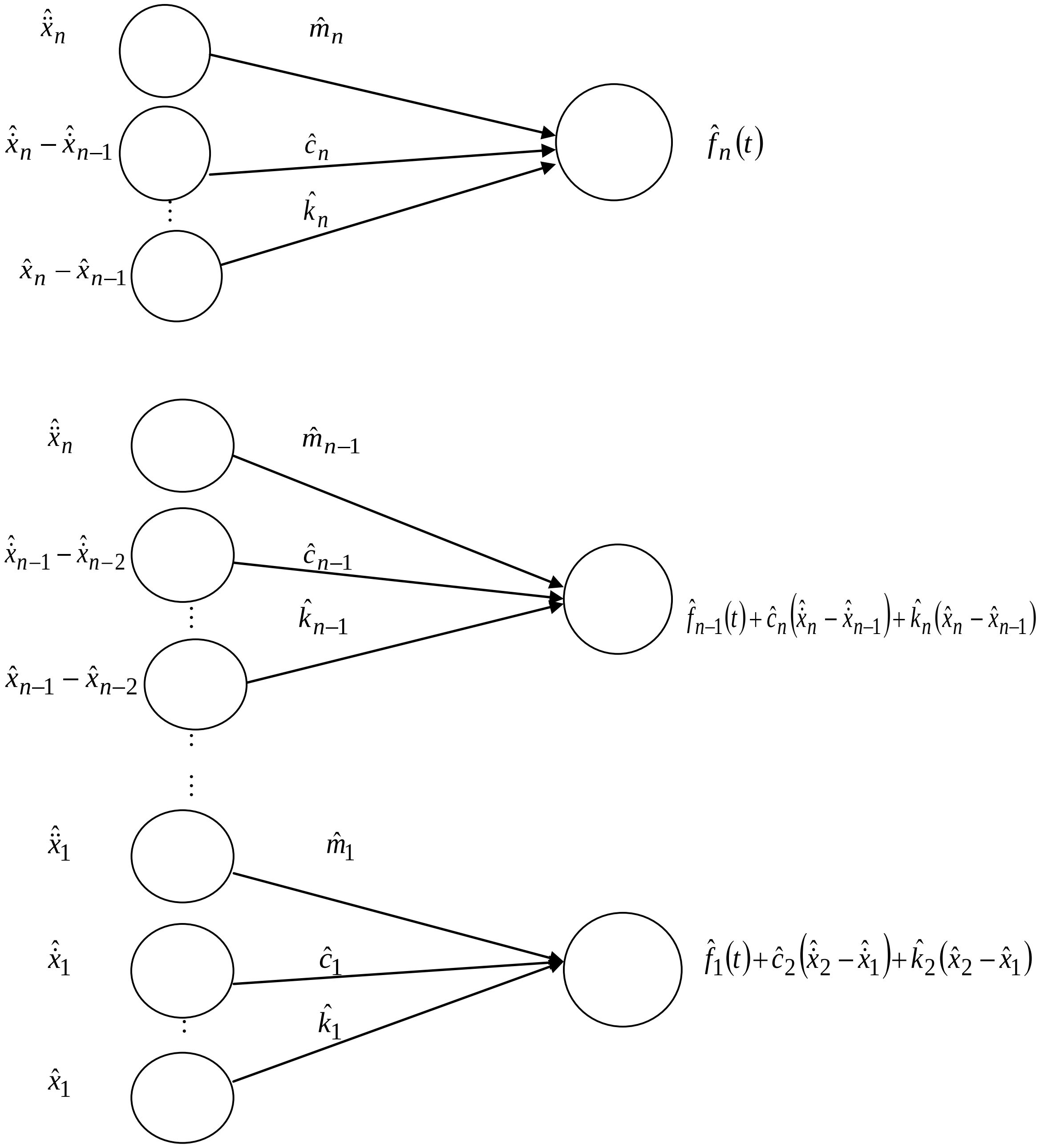
$\hat{\ddot{x}}_n$
$\hat{m}_n$
$\hat{\dot{x}}_n - \hat{\dot{x}}_{n-1}$
$\hat{c}_n$
$\hat{f}_n(t)$
$\hat{k}_n$
$\hat{x}_n - \hat{x}_{n-1}$
$\hat{\ddot{x}}_n$
$\hat{m}_{n-1}$
$\hat{\dot{x}}_{n-1} - \hat{\dot{x}}_{n-2}$
$\hat{c}_{n-1}$
$\hat{k}_{n-1}$
$\hat{f}_{n-1}(t) + \hat{c}_n(\hat{\dot{x}}_n - \hat{\dot{x}}_{n-1}) + \hat{k}_n(\hat{x}_n - \hat{x}_{n-1})$
$\hat{x}_{n-1} - \hat{x}_{n-2}$
$\hat{\ddot{x}}_1$
$\hat{m}_1$
$\hat{\dot{x}}_1$
$\hat{c}_1$
$\hat{f}_1(t) + \hat{c}_2(\hat{\dot{x}}_2 - \hat{\dot{x}}_1) + \hat{k}_2(\hat{x}_2 - \hat{x}_1)$
$\hat{k}_1$
$\hat{x}_1$

Figure 6.3: Proposed Cluster of Fuzzy Neural Network Model for n Storey Shear Structure

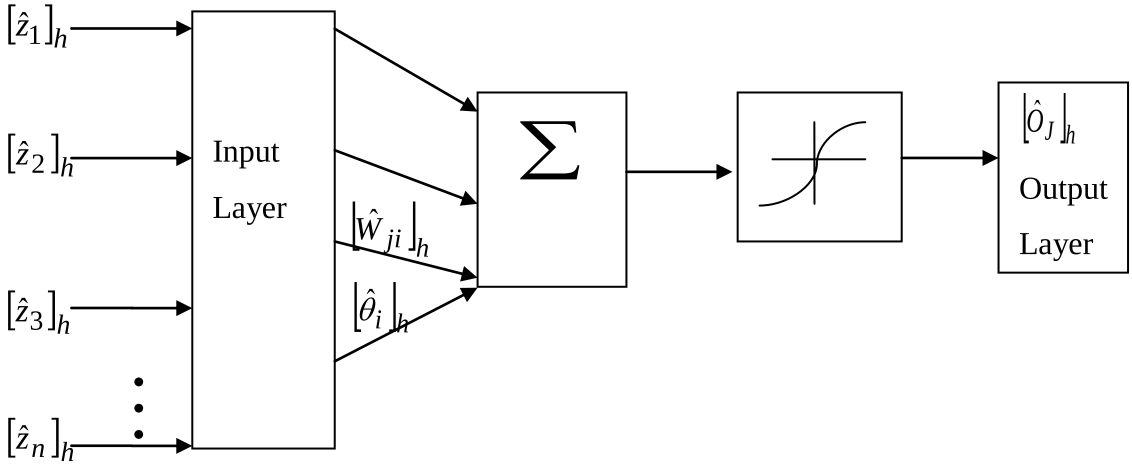
$[\hat{z}_1]_h$
$[\hat{z}_2]_h$
$[\hat{z}_3]_h$
$[\hat{z}_n]_h$
Input
Layer
$\lfloor \hat{W}_{ji} \rfloor_h$
$\lfloor \hat{\theta}_i \rfloor_h$
Σ
$\lfloor \hat{O}_J \rfloor_h$
Output
Layer

Figure 6.4: Single Layer Fuzzy Neural Network

6.5 Numerical Examples

6.5.1 Interval Case

The above method has been developed for different storey shear structures with damping and without damping cases. Interval neural network is trained till the desired accuracy is reached. The methodology has been discussed below by giving the results for following two cases.

Case (i) Without Damping: In this case three problems are considered which are

a) Two storey
b) Four storey
c) Eight storey

Case (ii) With Damping: In this case only one problem of three storey shear structure has been considered

a) Three storey

Case (i) (a) Two storey shear building:

Structural parameter of shear building has been identified using the direct method where the data are considered to be in interval form. The data are initially generated by taking the theoretical structural parametric values. These generated data are used first to train the neural network for n training patterns thus by establishing the converged weight matrix of the neural network. Corresponding component of the converged weight matrix gives the unknown or present structural parameters. Then the trained and theoretical data has been compared to show the efficiency of the proposed method. These ideas have been applied in all the cases. The initial structural parameter matrices in interval form are taken as: storey masses $\tilde{m}_1 = [1.5, 2.5]$, $\tilde{m}_2 = [1, 2]$ Kg, and the storey stiffnesses $\tilde{k}_1 = [390, 410]$ $\tilde{k}_2 = [290, 310]$Nm^{-1}. The harmonic force exerted in the shear building are assumed in interval form viz. $\tilde{f}_1(t) = [90\sin(1.6\pi t) + \pi, 110\sin(1.6\pi t) + \pi]$ and $\tilde{f}_2(t) = [90\sin(1.6\pi t), 110\sin(1.6\pi t)]$N. Table 6.1 includes the comparison of identified structural parameters with the theoretical parameters. The epoch versus mass and stiffness for two storeys are plotted in Figures 6.5-6.8 to show that how the structural parameters converge.

Table 6.1: Identified mass and stiffness parameters of two storey shear building in interval form under the forced vibration test (without damping)

Parameter	Storey	Theoretical	Identified	Centered Theoretical	Centered Identified
Mass (Kg)	M_1	[1.5, 2.5]	[1.6136, 2.4988]	2	1.9866
	M_2	[1, 2]	[1.0915, 2.2869]	1.5	1.6859
Stiffness (N m^{-1})	K_1	[390, 410]	[389.9947,409.9999]	400	399.9994
	K_2	[290, 310]	[290.008, 309.9951]	300	299.1889

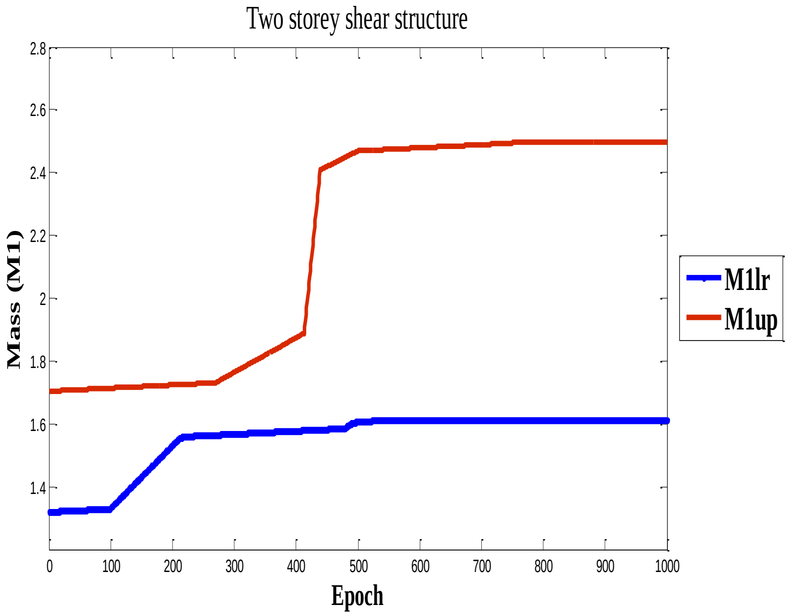

Figure 6.5: Convergence of interval mass parameter (M_1) with respect to number of epoch for two storey shear structure (without damping)

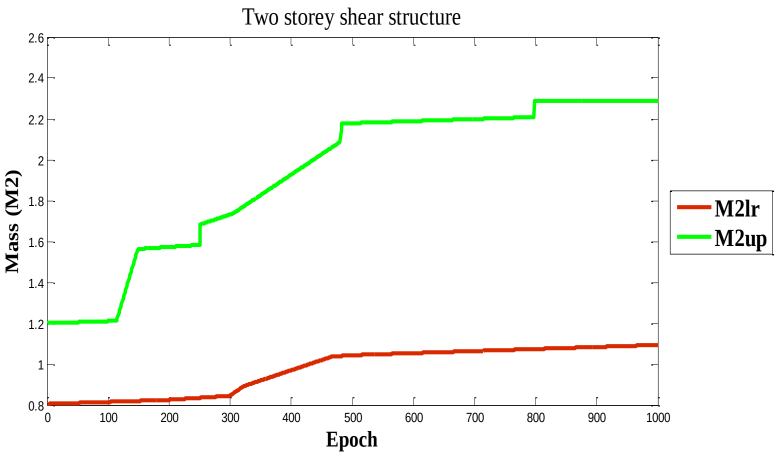

Figure 6.6: Convergence of interval mass parameter (M_2) with respect to number of epoch for two storey shear structure (without damping)

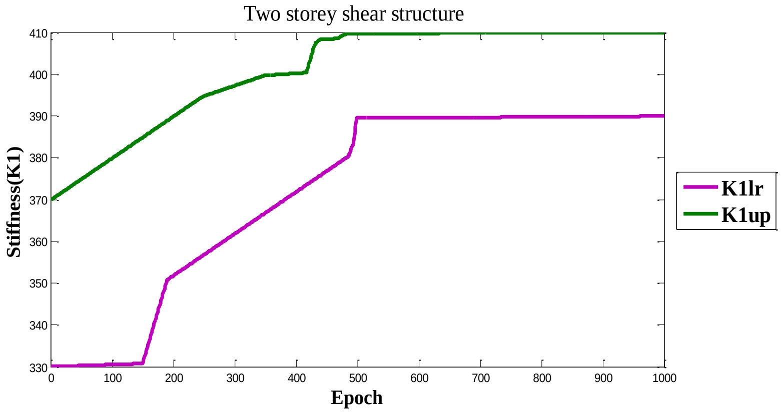

Figure 6.7: Convergence of interval stiffness parameter (K_1) with respect to number of epoch for two storey shear structure (without damping)

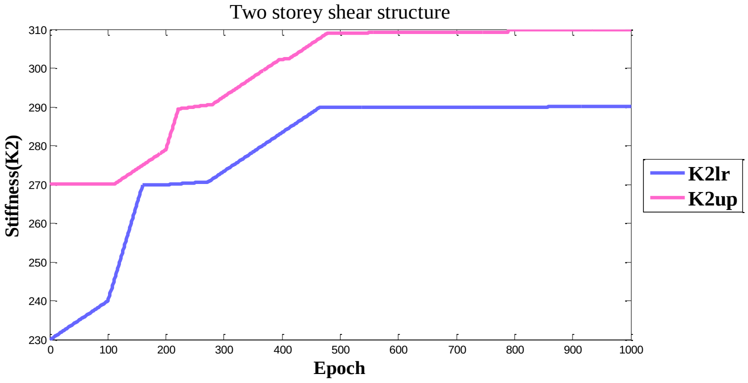

Figure 6.8: Convergence of interval stiffness parameter (K_2) with respect to number of epoch for two storey shear structure (without damping)

Case (i) (b) Four storey shear buildings:

In this problem the structural parameters in interval form are: the storey masses $\tilde{m}_1 = [3, 4]$, $\tilde{m}_2 = [3, 4], \tilde{m}_3 = [2.5, 3.5]$, $\tilde{m}_4 = [2, 3]$Kg, and the storey stiffnesses $\tilde{k}_1 = [1190, 1210]$, $\tilde{k}_2 = [1190, 1210]$, $\tilde{k}_3 = [790, 810]$ $\tilde{k}_4 = [590, 610]$Nm^{-1}. The harmonic forces exerted in the shear building are taken as $\tilde{f}_1(t) = [90\sin(1.6\pi t) + \pi, 110\sin(1.6\pi t) + \pi]$, $\tilde{f}_2(t) = [90\sin(1.6\pi t), 110\sin(1.6\pi t)]$, $\tilde{f}_3(t) = [0.5\sin(3.2\pi t), 1.5\sin(3.2\pi t)]$ and $\tilde{f}_4(t) = [1.0\sin(3.2\pi t), 3\sin(3.2\pi t)]$N. Dynamic displacements of four DOFs with duration of 1sec are stored and are used to train the interval neural network for identification procedure. Comparisons between the identified and theoretical parameters of the structures in interval form are incorporated in Table 6.2.

Table 6.2: Identified mass and stiffness parameters of four storey shear building in interval form under the forced-vibration test (without damping)

Parameter	Storey	Theoretical	Identified	Centered Theoretical	Centered Identified
Mass (Kg)	M_1	[3, 4]	[2.8, 3.999]	3.5	3.5000
	M_2	[3, 4]	[2.9, 3.899]	3.5	3.5000
	M_3	[2.5, 3.5]	[2.3577, 3.3336]	3	3.0825
	M_4	[2, 3]	[1.7690, 2.9828]	2.5	2.4627
Stiffness (N m^{-1})	K_1	[1190, 1210]	[1190.00, 1210.00]	1200	1199.0988
	K_2	[1190, 1210]	[1190.00, 1210.00]	1200	1199.0008
	K_3	[790, 810]	[789.998, 809.999]	800	799.9976
	K_4	[590, 610]	[589.9983, 610.000]	600	599.9999

Case (i) (c) Eight storey shear buildings:

In this case the structural parameters viz., the storey masses and stiffnesses of the structure in interval form are given in Table 6.3. The harmonic force in interval form exerted in the eight storey of the building is $\tilde{f}_8(t) = [1.0\sin(3.2\pi t), 3.0\sin(3.2\pi t)]$N. The dynamic displacements of eight DOFs with duration of 1 s are stored and are used to train the interval neural network for identification procedure. The identified structural parameters are compared with the theoretical structural parameters of this structural system. Corresponding result are shown in Table 6.3.

Table 6.3: Identified mass and stiffness parameters of eight-storey shear building in interval form under the forced-vibration test (without damping)

Parameter	Storey	Theoretical	Identified	Centered Theoretical	Centered Identified
Mass (Kg)	M_1	[3, 4]	[2.899, 3.999]	3.5	3.5000
	M_2	[2.5, 3.5]	[2.500, 3.500]	3	3.0000
	M_3	[2.5, 3.5]	[2.1858, 3.500]	3	2.9435
	M_4	[2.5, 3.5]	[2.4453, 3.500]	3	2.8900
	M_5	[2, 3]	[1.6021, 3.00]	2.5	2.4378
	M_6	[2, 3]	[1.6644, 3.00]	2.5	2.4998
	M_7	[2, 3]	[1.8965, 3.00]	2.5	2.3492
	M_8	[1.5, 2.5]	[1.2234, 2.500]	2	1.9928
Stiffness (N m^{-1})	K_1	[1190, 1210]	[1190.00, 1210.00]	1200	1199.9879
	K_2	[1190, 1210]	[1190.00, 1210.00]	1200	1200.0000
	K_3	[790, 810]	[789.998, 810.000]	800	799.9985
	K_4	[790, 810]	[789.999, 810.000]	800	799.9997
	K_5	[790, 810]	[789.9993, 810.000]	800	799.9992
	K_6	[590, 610]	[589.999, 610.000]	600	599.9955
	K_7	[590, 610]	[589.9997, 610.00]	600	599.8989
	K_8	[590, 610]	[589.9999, 610.00]	600	599.9998

Case (ii) (a) Three storey shear buildings:

Structural parameters in interval form are taken as: the storey masses $\tilde{m}_1 = [2.5, 3.5]$, $\tilde{m}_2 = [2.5, 3.5]$, $\tilde{m}_3 = [1.5, 2.5]$Kg and the storey stiffnesses $\tilde{k}_1 = [1190, 1210]$, $\tilde{k}_2 = [790, 810]$ $\tilde{k}_3 = [590, 610]$Nm^{-1}; the damping values $\tilde{c}_1 = [7, 9]$, $\tilde{c}_2 = [4, 6]$, $\tilde{c}_3 = [3, 5]$ N s m^{-1}. The harmonic forces exerted in the shear building in interval form are considered as $\tilde{f}_1(t) = [90\sin(1.6\pi t) + \pi, 110\sin(1.6\pi t) + \pi]$, $\tilde{f}_2(t) = [90\sin(1.6\pi t), 110\sin(1.6\pi t]$ and $\tilde{f}_3(t) = [0.5\sin(3.2\pi t), 1.5\sin(3.2\pi t)]$N. Comparisons between the identified and theoretical structural parameters are incorporated in Table 6.4.

Table 6.4: Identified mass and stiffness parameters of three-storey shear building in interval form under the forced-vibration test (with damping)

Parameter	Storey	Theoretical	Identified	Centered Theoretical	Centered Identified
Mass (Kg)	M_1	[2.5, 3.5]	[2.50, 3.50]	3	3.0000
	M_2	[2.5, 3.5]	[2.2451, 3.3727]	3	2.8881
	M_3	[1.5, 2.5]	[1.4321, 2.331]	2	1.6813
Stiffness (N m^{-1})	K_1	[1190, 1210]	[1190.00, 1210.00]	1200	1200.000
	K_2	[790, 810]	[789.991, 809.992]	800	799.9997
	K_3	[590, 610]	[589.997, 609. 998]	600	599.9990
Damping (N s m^{-1})	C_1	[7, 9]	[7.00, 9.00]	8	8.0000
	C_2	[4, 6]	[3.9858, 5.9903]	5	4.9948
	C_3	[3, 5]	[2.9991, 4.9938]	4	3.9835

6.5.2 Fuzzy Case

The above developed method has been used for different storey shear structures with damping and without damping cases. Fuzzy neural network training is done till a desired accuracy is achieved. The methodology has been discussed by giving the results for following two cases.

Case (i) Without Damping: In this case three problems are considered which are

a) Two storey
b) Four storey
c) Eight storey

Case (ii) With Damping: In this case only one problem of three storey shear structure has been considered

a) Three storey

Case (i) (a) Two storey shear buildings:

Structural parameter of shear building has been identified using the direct method where the data are considered to be in fuzzified form. To generate the data initially the theoretical values of the structural parameters are taken. These generated data are used first to train the neural network for n training patterns thus by establishing the converged weight matrix of the neural network. Corresponding component of the converged weight matrix gives the unknown or present structural parameters. Then the trained and theoretical data has been compared to show the efficiency of the proposed method. These ideas have been applied in all the cases. The initial structural parameter matrices in fuzzified form are taken as: storey masses $\hat{m}_1 = [1.5, 2, 2.5]$, $\hat{m}_2 = [1, 1.5, 2]$ Kg, and the storey stiffnesses $\hat{k}_1 = [390, 400, 410]$, $\hat{k}_2 = [290, 300, 310]$ Nm^{-1}. The harmonic force exerted in the shear building are assumed in fuzzified form viz. $\hat{f}_1(t) = [90\sin(1.6\pi t) + \pi, 100\sin(1.6\pi t) + \pi, 110\sin(1.6\pi t) + \pi]$ and $\hat{f}_2(t) = [90\sin(1.6\pi t), 100\sin(1.6\pi t), 110\sin(1.6\pi t)]$N. Table 6.5 includes the comparison of identified structural parameters with the theoretical parameters.

Table 6.5: Identified mass and stiffness parameters of two-storey shear building in fuzzified form under the forced-vibration test (without damping)

Parameter	Storey	Theoretical	Identified
Mass (Kg)	M_1	[1.5, 2, 2.5]	[1.4650, 1.9866, 2.4939]
	M_2	[1,1.5, 2]	[1.0457, 1.6859, 2.3077]
Stiffness (N m^{-1})	K_1	[390, 400, 410]	[389.9991, 399.9994, 409.9997]
	K_2	[290, 300, 310]	[290.4673, 299.1889, 308.5296]

Case (i) (b) Four storey shear buildings:

In this problem the structural parameters in fuzzified form are: the storey masses $\hat{m}_1 = [3, 3.5, 4], \hat{m}_2 = [3, 3.5, 4]$, $\hat{m}_3 = [2.5, 3, 3.5]$, $\hat{m}_4 = [2, 2.5, 3]$Kg, and the storey stiffnesses $\hat{k}_1 = [1190, 1200, 1210]$, $\hat{k}_2 = [1190, 1200, 1210]$, $\hat{k}_3 = [790, 800, 810]$ $\hat{k}_4 = [590, 600, 610]$Nm^{-1}. The harmonic forces exerted in the shear building are taken as $\hat{f}_1(t) = [90\sin(1.6\pi t) + \pi, 100\sin(1.6\pi t) + \pi, 110\sin(1.6\pi t) + \pi]$, $\hat{f}_2(t) = [90\sin(1.6\pi t), 100\sin(1.6\pi t), 110\sin(1.6\pi t)]$ $\hat{f}_3(t) = [0.5\sin(3.2\pi t), 1.0\sin(3.2\pi t), 1.5\sin(3.2\pi t)]$and $\hat{f}_4(t) = [1.0\sin(3.2\pi t), 2.0\sin(3.2\pi t), 3.0\sin(3.2\pi t)]$N. Dynamic displacements of four DOFs with duration of 1s are stored and are used to train the fuzzy neural network for identification procedure. Comparisons between the identified and theoretical parameters of the structures in fuzzified form are incorporated in Table 6.6.

Table 6.6: Identified mass and stiffness parameters of four storey shear building in fuzzified form under the forced-vibration test (without damping)

Parameter	Storey	Theoretical	Identified
Mass (Kg)	M_1	[3, 3.5, 4]	[2.859, 3.549, 4.001]
	M_2	[3, 3.5, 4]	[2.900, 3.501, 3.898]
	M_3	[2.5, 3.0, 3.5]	[2.2771, 3.0825, 3.2257]
	M_4	[2, 2.5, 3]	[1.7690, 2.4627, 2.9828]
Stiffness (N m^{-1})	K_1	[1190, 1200, 1210]	[1190.00, 1199.0988, 1210.00]
	K_2	[1190, 1200, 1210]	[1190.00, 1199.008, 1210.051]
	K_3	[790, 800, 810]	[789.9977, 799.9974, 810.001]
	K_4	[590, 600, 610]	[589.9983, 599.991, 610.000]

Case (i) (c) Eight storey shear buildings:

In this case the structural parameters viz., the storey masses and stiffnesses of the structure in fuzzified form are given in Table 6.7. The harmonic force in fuzzified form exerted in the eight storey of the building is $\hat{f}_8(t) = [1.0\sin(3.2\pi t), 2.0\sin(3.2\pi t), 3.0\sin(3.2\pi t)]$N. The dynamic displacements of eight DOFs with duration of 1 s are stored and are used to train the fuzzy neural network for identification procedure. The identified structural parameters are compared with the theoretical structural parameters of this structural system. Corresponding result are shown in Table 6.7.

Table 6.7: Identified mass and stiffness parameters of eight storey shear building in fuzzified form under the forced-vibration test (without damping).

Parameter	Storey	Theoretical	Identified
Mass (Kg)	M_1	[3, 3.5, 4]	[2.899, 3.500, 3.999]
	M_2	[2.5, 3, 3.5]	[2.500, 3.00, 3.500]
	M_3	[2.5, 3, 3.5]	[2.3425, 2.9435, 3.500]
	M_4	[2.5, 3, 3.5]	[2.4504, 2.8900, 3.500]
	M_5	[2, 2.5, 3]	[1.6672, 2.4378, 3.00]
	M_6	[2, 2.5, 3]	[1.6916, 2.4998, 3.00]
	M_7	[2, 2.5, 3]	[1.9706, 2.3492, 3.00]
	M_8	[1.5, 2, 2.5]	[1.2336, 1.9928, 2.500]
Stiffness (N m^{-1})	K_1	[1190, 1200, 1210]	[1190.00, 1199.9879, 1210.00]
	K_2	[1190, 1200, 1210]	[1190.00, 1200.001, 1210.00]
	K_3	[790, 800, 810]	[789.998, 799.9985, 810.000]
	K_4	[790, 800, 810]	[789.999, 799.9997, 810.000]
	K_5	[790, 800, 810]	[789.9998, 799.9992, 810.000]
	K_6	[590, 600, 610]	[589.9994, 599.9955, 610.000]
	K_7	[590, 600, 610]	[589.9997, 599.8989, 610.00]
	K_8	[590, 600, 610]	[589.9988, 599.8999 , 610.00]

Case (ii) (a) Three storey shear buildings:

Structural parameters in fuzzified form are taken as: the storey masses $\tilde{m}_1 = [2.5, 3, 3.5]$, $\tilde{m}_2 = [2.5, 3, 3.5]$, $\tilde{m}_3 = [1.5, 2, 2.5]$Kg and the storey stiffnesses $\tilde{k}_1 = [1190, 1200, 1210]$, $\tilde{k}_2 = [790, 800, 810]$, $\tilde{k}_3 = [590, 600, 610]$Nm^{-1}; the damping values $\tilde{c}_1 = [7, 8, 9]$, $\tilde{c}_2 = [4, 5, 6]$, $\tilde{c}_3 = [3, 4, 5]$N s m^{-1}. The harmonic forces exerted in the shear building in fuzzified form are considered as $\tilde{f}_1(t) = [90\sin(1.6\pi t) + \pi, 100\sin(1.6\pi t) + \pi, 110\sin(1.6\pi t) + \pi]$, $\tilde{f}_2(t) = [90\sin(1.6\pi t), 100\sin(1.6\pi t), 110\sin(1.6\pi t)]$ and $\tilde{f}_3(t) = [0.5\sin(3.2\pi t), 1.0\sin(3.2\pi t), 1.5\sin(3.2\pi t)]$ N. Comparisons between the identified and theoretical structural parameters are incorporated in Table 6.8. The epoch versus mass, stiffness and damping for three storey are plotted in Figures 6.9-6.17 to show that how the structural parameters converge.

Table 6.8: Identified mass and stiffness parameters of three storey shear building in fuzzified form under the forced vibration test (with damping)

Parameter	Storey	Theoretical	Identified
Mass (Kg)	M_1	[2.5, 3, 3.5]	[2.5101, 3.001,3.498]
	M_2	[2.5, 3, 3.5]	[2.596, 2.8881, 3.373]
	M_3	[1.5, 2, 2.5]	[1.5737, 1.6837, 2.139]
Stiffness (N m^{-1})	K_1	[1190, 1200, 1210]	[1190.95, 1199.001, 1210.95]
	K_2	[790, 800, 810]	[789.991, 799.957, 809.992]
	K_3	[590, 600, 610]	[589.993, 599.990, 609. 995]
Damping (N s m^{-1})	C_1	[7, 8, 9]	[6.599, 7.895, 8.999]
	C_2	[4, 5, 6]	[3.999, 4.999, 5.999]
	C_3	[3, 4, 5]	[2.9980, 3.9835, 4.9867]

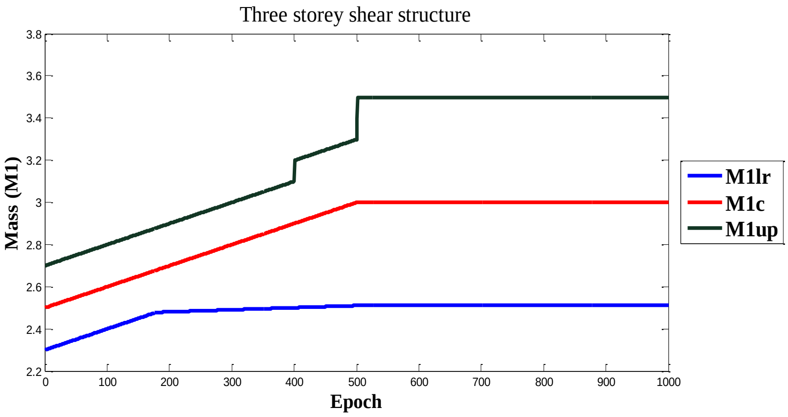

Figure 6.9: Convergence of fuzzified mass parameter ($\hat{M}_1$) with respect to number of epoch for three storey shear structure (with damping)

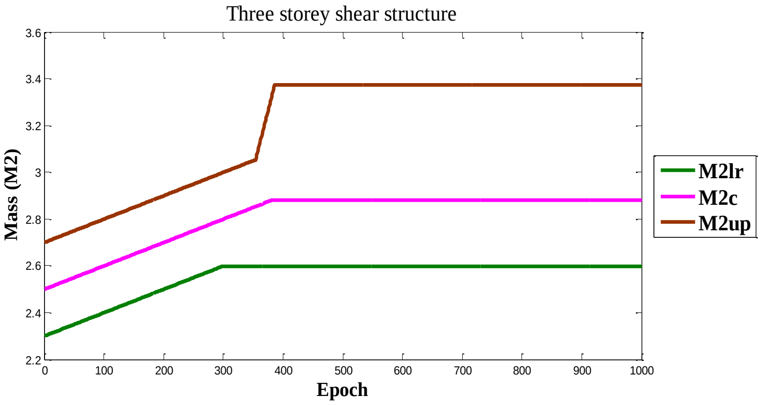

Figure 6.10: Convergence of fuzzified mass parameter ($\hat{M}_2$) with respect to number of epoch for three storey shear structure (with damping)

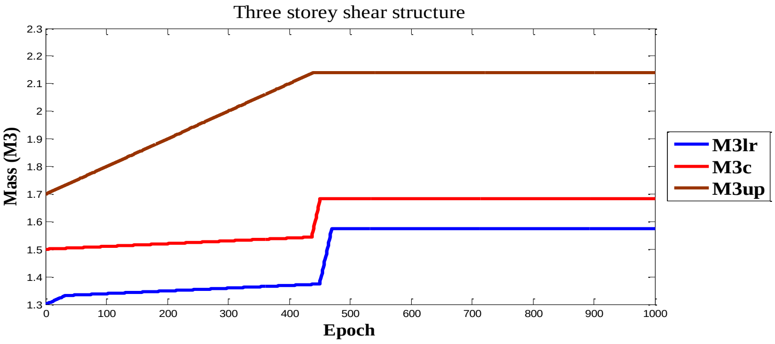

Figure 6.11: Convergence of fuzzified mass parameter ($\hat{M}_3$) with respect to number of epoch for three storey shear structure (with damping)

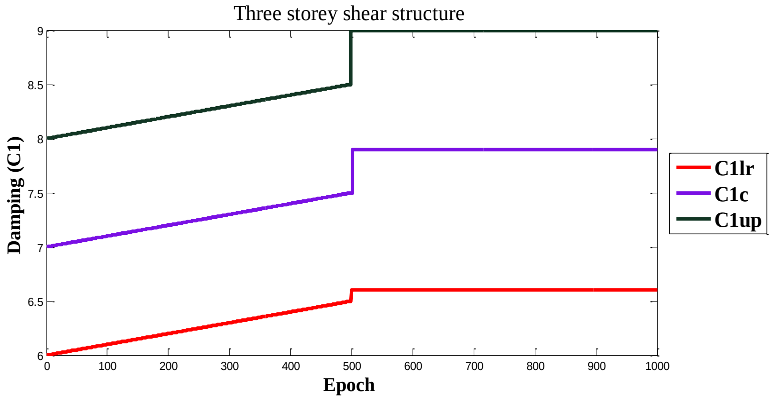

Figure 6.12: Convergence of fuzzified damping parameter (C_1) with respect to number of epoch for three storey shear structure (with damping)

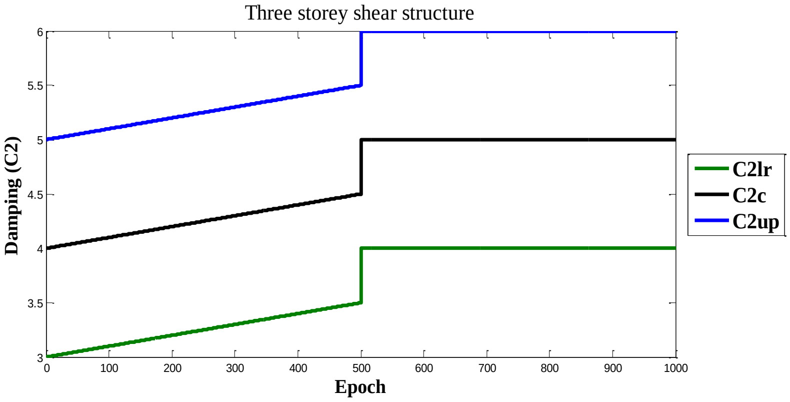

Figure 6.13: Convergence of fuzzified damping parameter (C_2) with respect to number of epoch for three storey shear structure (with damping)

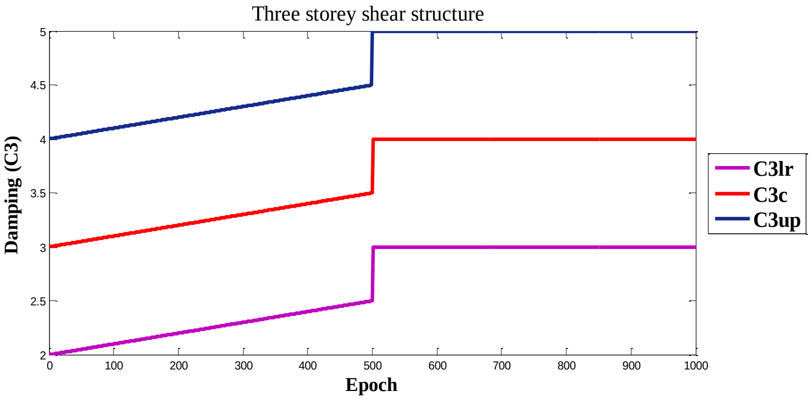

Figure 6.14: Convergence of fuzzified stiffness parameter (C_3) with respect to number of epoch for three storey shear structure (with damping).

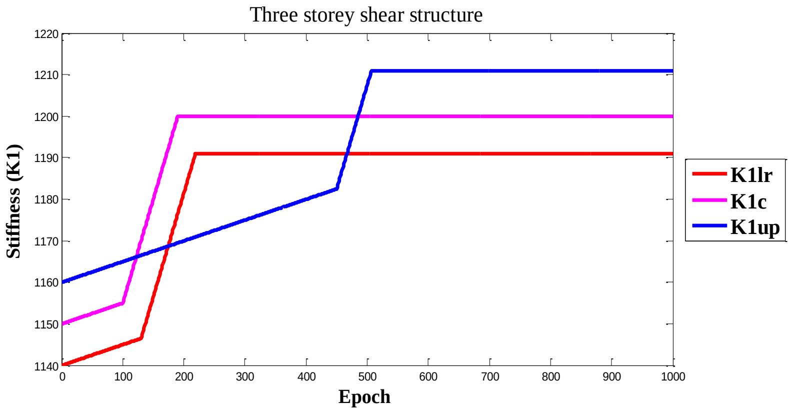

Figure 6.15: Convergence of fuzzified stiffness parameter (K_1) with respect to number of epoch for three storey shear structure (with damping).

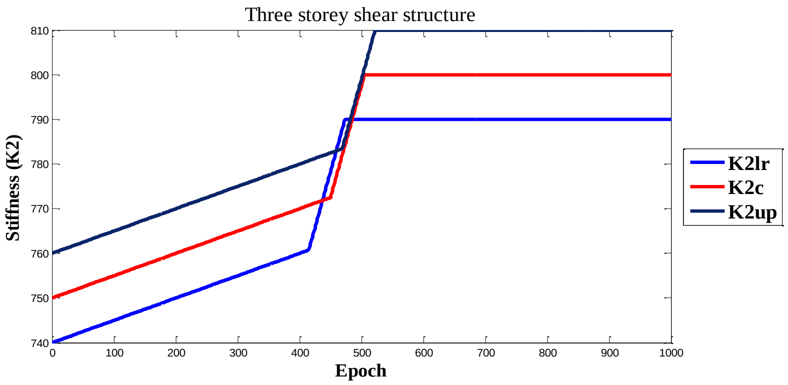

Figure 6.16: Convergence of fuzzified stiffness parameter (K_2) with respect to number of epoch for three storey shear structure (with damping).

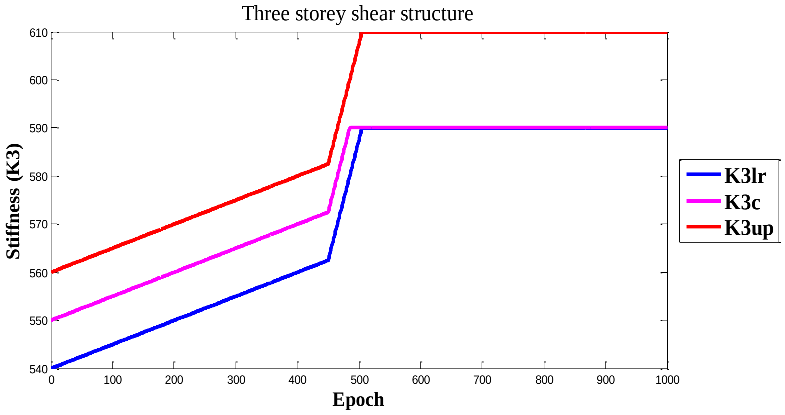

Figure 6.17: Convergence of fuzzified stiffness parameter (K_3) with respect to number of epoch for three storey shear structure (with damping).

6.6 Conclusion

This paper uses the powerful soft computing technique viz. single layer Interval and Fuzzy Neural Network for identification of uncertain structural parameters. If the available information and/or data are uncertain or nonprobabilistic in nature, then the mathematical model needs to be developed accordingly. As such, interval and fuzzy neural network has been developed which can handle uncertain (interval and fuzzified) data. Here, direct method for system identification of uncertain multistory shear structures from their dynamic responses has been proposed in interval and fuzzified form. Governing equations of motion are modified based on relative responses of consecutive stories and are implemented in a series/cluster of interval and fuzzy neural network models. Various example problems have been solved and related results are reported to show the reliability and powerfulness of the method. It is worth mentioning that the cluster of INN and FNN may directly estimate the structural parameters in interval and fuzzified form. The interval and fuzzy estimates are certainly useful for design engineers by knowing the bounds of the structural parameters.

Chapter 7

Functional Link Neural Network Based System Identification from Frequency Data

The objective of this Chapter is to estimate structural parameters by developing a novel Functional Link Neural Network (FLNN) model. Functional Link Neural Network model is more efficient than Multi layer Neural Network (MNN) as computation is less because hidden layer is not required. Here single layer neural network with multi-input and multi-output with feed forward neural network and principle of error back propagation has been used to identify structural parameters. The hidden layer is excluded by enlarging the input patterns with the help of Chebyshev, Legendre and Hermite polynomials. Comparison of results among MNN, Chebyshev Neural Network (ChNN), Legendre Neural Network (LeNN), Hermite Neural Network (HNN) and desired are considered and it is found that Functional Link Neural Network models are more effective than MNN.

7.1 Modelling for System Identification of Multi-Storey Shear Buildings

System Identification methods in particular uses the values of the parameters initially given to the structure by an engineer. The governing equation of motion for n-storey shear structure without damping may be written by n coupled second order ordinary differential equations as

$$[M]\{\ddot{Y}\}+[K]\{Y\}=\{0\} \tag{7.1}$$

Here $[M]$ is a $n\times n$ mass matrix which is given as

$$[M]=\begin{bmatrix} m_1 & 0 & \dots & \dots & 0 \\ 0 & m_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & m_{n-1} & 0 \\ 0 & \dots & \dots & 0 & m_n \end{bmatrix}$$

The $n\times n$ stiffness matrix may be written as

$$[K]=\begin{bmatrix} k_1+k_2 & -k_2 & 0 & \dots & 0 \\ -k_2 & k_2+k_3 & -k_3 & \dots & 0 \\ \dots & \dots & & & \dots \\ 0 & \dots & -k_{n-1} & k_{n-1}+k_n & k_n \\ 0 & & \dots & -k_n & k_n \end{bmatrix}$$

and $\{Y\}=\{y_1, y_2, \dots, y_n\}^T$ is the displacement vector.

The above free vibration equation is solved for finding the frequency and mode shapes. Putting $\{Y\}=\{\phi\}e^{i\,\omega\,t}$ in free vibration equation (7.1) we get the following characteristic equation

$$\left([K]-[M][\omega]^2\right)\{\phi\}=\{0\} \tag{7.2}$$

where $[\omega]^2=[\lambda]$ are the eigenvalues or the frequency and $\{\phi\}$ are mode shapes of the structure respectively. The frequencies are considered as the input and the stiffness parameters are considered as the output for FLNN.

7.2 Functional Link Neural Network

FLNN consists of a single layer structure where the hidden nodes are excluded and the inputs patterns are expanded to higher dimensional patterns by means of functional expansion block. Architecture of the FLNN model comprises of two parts viz. functional expansion part and learning part respectively. Due to the absence of hidden layer, FLNNs are computationally more efficient and faster than MNN. It is worth mentioning that MNN is well known and the detail training and model etc. may be found in various books and papers, Zurada [186]. As such here, we will include details about the FLNN only. In FLNN, the input vector is expanded to higher dimensional vector by some suitable functions. Let us consider *i* input vector denoted as $X=\{x_1, x_2, \dots, x_i\}^T$. Thus, the enhanced input vector can be expressed as $X^T=U_j(X)$, where $U=[u_1(X), u_2(X), \dots, u_N(X)]$ $\{U\}_{j=1}^{N}$ are set of functions. These functions are replaced here by set of orthogonal polynomials. Orthogonal polynomials used for the present FLNN model are Chebyshev, Legendre and Hermite polynomials. Here basic orthogonal polynomials or any other orthogonal polynomials may be used to train this FLNN. For, basic orthogonal polynomials for training we may start with algebraic polynomials and then othogonalising them using Grahm Schmidt Orthogonalisation procedure. These are certainly simple but sometimes suffer from instability. As such, instead of these basic orthogonal polynomials, Chebyshev, Legendre and Hermite polynomials have been used here. Structure of single layer multi-input and multi-output FLNN model that has been considered and its learning algorithms have been discussed below.

7.3 Learning Algorithm of Functional Link Neural Network (FLNN)

In FLNN, the weights are updated to minimize a given cost function. Here Feed forward and error back propagation algorithm has been used for learning and for updating the weights of FLNN. Inputs $x_i = \lambda_i$ are the frequencies and outputs $O_J = k_J$ are the stiffness parameters. As such, linear sum Z_J can be calculated as

$$Z_J = \sum_{j=1}^{N} w_j U_j(X) + \theta_j \tag{7.3}$$

$$J = i = \text{number of input vectors and } j \text{ is the order of the polynomial}$$

where w_j are the weights, θ_j are the bias and $U_j(X)$ are expanded inputs vector. These $U_j(X)$ are considered here as Chebyshev, Legendre and Hermite polynomials.

The net output is given as

$$O_J = f(Z_J)$$

Here bipolar sigmoidal function has been used as the activation function and defined as

$$f = \frac{1-\exp(Z_J)}{1+\exp(Z_J)}$$

Cost function used for minimization of error is defined as

$$E = \frac{1}{2}\left[d_j - O_j\right]^2 = \frac{1}{2}e_j^2 \tag{7.4}$$

where d_J is the desired output, O_J is the target output and e_J is the error value. The error value is computed to obtain the desired accuracy. Weights are updated as follows:

$$w_j(New) = w_j(Old) + \Delta w_j \tag{7.5}$$

where change in weights are calculated as

$$\Delta w_j = \left[-\eta \frac{\partial E}{\partial w_j}\right] = \left[-\eta\ (d_J - O_J)\left(1 - O_J{}^2\right) U_i(X)\right] \tag{7.6}$$

Here η is the learning parameter. We follow the same procedure to update the bias θ_j. The multi-input and multi-output single layer FLNN architecture is shown in Figure 7.1, where $\{U_j\}_{j=1}^N$ are set of Chebyshev, Legendre and Hermite orthogonal polynomials.

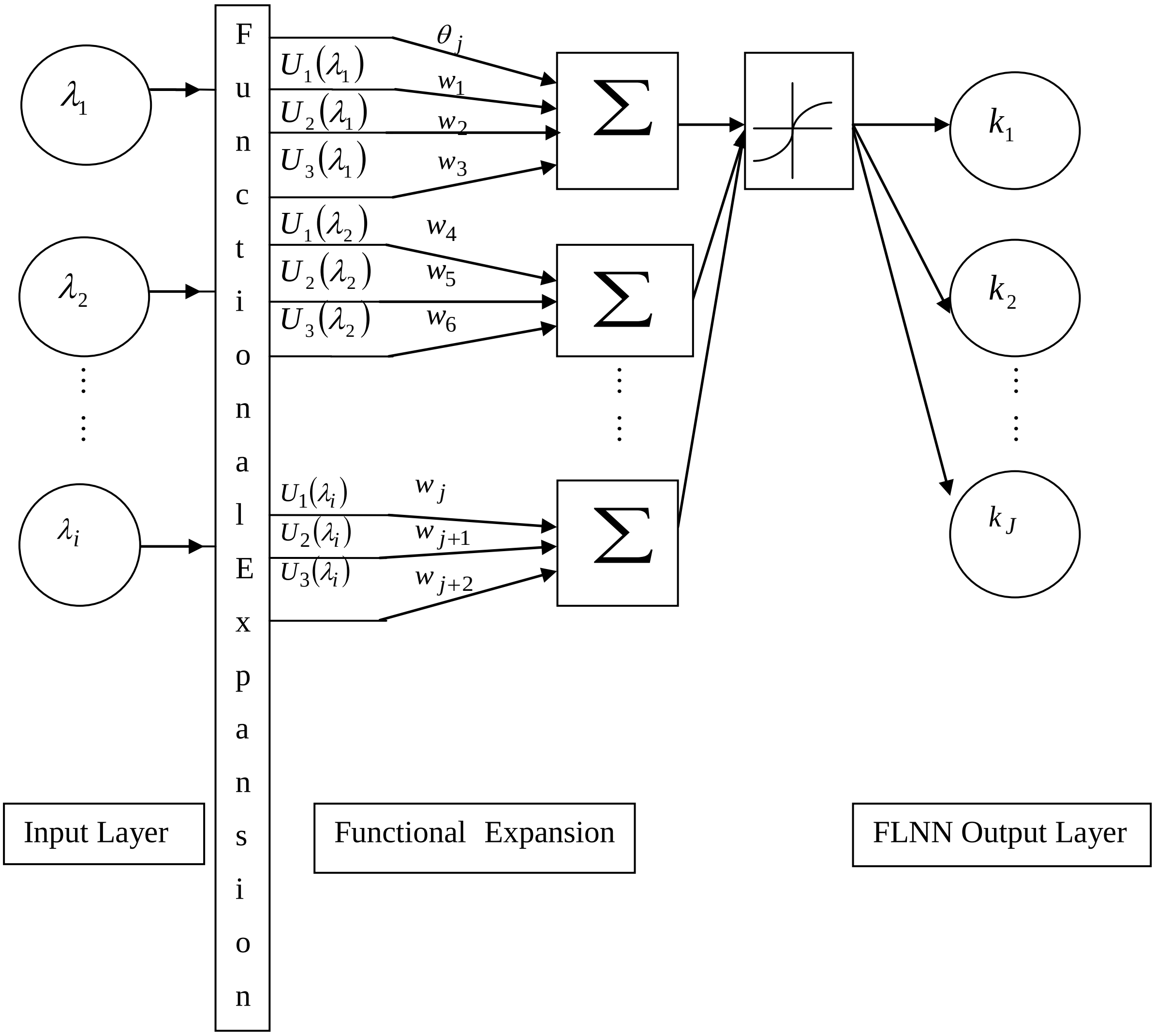

Figure 7.1: Single Layer Multi-Input and Multi-Output Functional Link Neural Network

7.3.1 Structure of Chebyshev Neural Network

Here we have developed a single layer multi-input and multi-output Chebyshev Neural Network (ChNN) model to study System Identification problems. The structure of ChNN comprises of multiple input nodes, a functional expansion block consists of Chebyshev polynomials and multiple output nodes. Functional expansion part and learning part are the two parts of Chebyshev neural network. Each input vector is expanded to several terms of Chebyshev polynomials so that they may be viewed as a new input vector in the functional expansion part. Chebyshev polynomials are a set of orthogonal polynomials. Let us consider *i* input vector denoted as $X = \{x_1, x_2, \ldots, x_i\}^T$. Chebyshev polynomial of *j*th order is denoted by $T_j(x)$, where -1 < *x* < 1. First four Chebyshev polynomials may be written as (Gerald [202] and Bhatt and Chakraverty [203])

$$\begin{aligned}
&T_0(x) = 1\\
&T_1(x) = x\\
&T_2(x) = 2x^2 - 1\\
&T_3(x) = 4x^3 - 3x\\
&T_4(x) = 8x^4 - 8x^2 + 1
\end{aligned}$$

Higher order Chebyshev polynomials may be generated by the well known recursive formula

$$T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x) \tag{7.7}$$

7.3.2 Structure of Legendre Neural Network

In Legendre neural network (LeNN), the suitable functions are taken as Legendre orthogonal polynomials in the expansion block. In functional expansion, each input is expanded to several terms using Legendre polynomials to have a new input vector. Here we have developed a single layer multi-input and multi-output Legendre Neural Network (LeNN) model to study System Identification problems. The structure of LeNN comprises of multiple input nodes (which is a functional expansion block obtained by Legendre polynomials) and multiple output nodes. LeNN involves less computations and offers faster training compared to MNN. The Legendre polynomial of *j*th order is denoted by $L_j(x)$, where -1 <*x*< 1. The first five Legendre polynomials may be written as in (Gerald [202] and Bhatt and Chakraverty [203])

$$\begin{aligned}
&L_1(x) = 1\\
&L_2(x) = x\\
&L_3(x) = \frac{1}{2}\left(3x^2 - 1\right)\\
&L_4(x) = \frac{1}{2}\left(5x^3 - 3x\right)\\
&L_5(x) = \frac{1}{8}\left(35x^4 - 30x^2 + 3\right)
\end{aligned}$$

Legendre polynomials of various orders may be obtained by the well known recursive rule

$$L_{j+1}(x) = \frac{1}{j+1}\left[(2j+1)xL_j(x) - jL_{j-1}(x)\right] \tag{7.8}$$

7.3.3 Structure of Hermite Neural Network

The structure of Hermite Neural Network (HNN) comprises of multiple input nodes (which now is a functional expansion block obtained from Hermite polynomials) and multiple output nodes. Here, again each input data is expanded to several terms using Hermite polynomials and these are now considered as a new input vector. A single layer multi-input and multi-output Hermite Neural Network (HNN) model is also developed for solving system identification problems. Hermite polynomial of j^{th} order is denoted by $H_j(x)$, where $-\infty < x < \infty$. The First five Hermite polynomials can be written as (Gerald [202] and Bhatt and Chakraverty [203])

$$\begin{aligned} H_1(x) &= 1 \\ H_2(x) &= 2x \\ H_3(x) &= 4x^2 - 2 \\ H_4(x) &= 8x^3 - 12x \\ H_5(x) &= 16x^4 - 48x^2 + 12 \end{aligned}$$

Various orders of Hermite polynomials may be generated by the well known recursive formula

$$H_{j+1}(x) = 2xH_j(x) - 2jH_{j-1}(x) \tag{7.9}$$

7.4 Results and Discussion

The procedure is demonstrated using few example problems. It may be mentioned that the frequency may practically be obtained by some experiments. But due to non availability of experimental data here the analyses have been done by numerical simulation. Present problem is first solved as forward vibration problem and set of frequency parameters are computed from initial design (stiffness) parameters. These designed stiffness parameters are randomized and training sets of frequency parameters are generated beforehand. Now the developed Functional Link neural network is trained with these patterns till the desired accuracy is reached. The orthogonal polynomials considered for present Functional Link neural network are Chebyshev, Legendre and Hermite polynomials. The results for different storey shear building, comparison tables and figures have been presented and discussed in the subsequent paragraphs.

7.4.1. Chebyshev Neural Network Based Results

Here examples of two, five and ten storey shear buildings are considered. For present Functional Link neural network Chebyshev polynomial upto 4th order are taken.

Example 1. Two storey shear buildings:

The inputs are taken as frequency and outputs are taken as stiffness parameters. Here two problems have been solved. For the first problem the storey masses are taken as $m_1 = m_2 = 36,000$ and the initial designed stiffness parameter as $k_1 = 100,000$ and $k_2 = 20,000$. Then for the second problem same masses are considered as that of the first problem and the stiffness are taken as $k_1 = 50,000$ and $k_2 = 20,000$. From these initial designed values of stiffness and masses we have generated 50 sets of data for frequency. These 50 training patterns are used for training the ChNN model. For these two problems different orders of Chebyshev polynomials are used for function expansion to obtain minimum error and to reach an accuracy of 0.001. Figure 7.2(a) shows the comparisons between the ChNN, multi layer neural network (MNN) and the desired for first floor and Figure 7.2(b) shows the comparisons between the ChNN, multi layer neural network (MNN) and the desired for second floor with 10 data for first problem. Comparison between the ChNN with fourth order polynomial, multi layer neural network (MNN) and the desired for 10 data are given in Table 7.1(a). The result between the ChNN with second order polynomial and the desired with 10 data have been shown in Table 7.1(b). Tables are shown for second problem. It is found that the ChNN takes less computation time than the MNN. The CPU time for the first problem using ChNN is 132.367 secs and with MNN is 526.332 secs. Similarly for second problem the CPU time for ChNN is 81.753 secs and for MNN it is 277.550 secs.

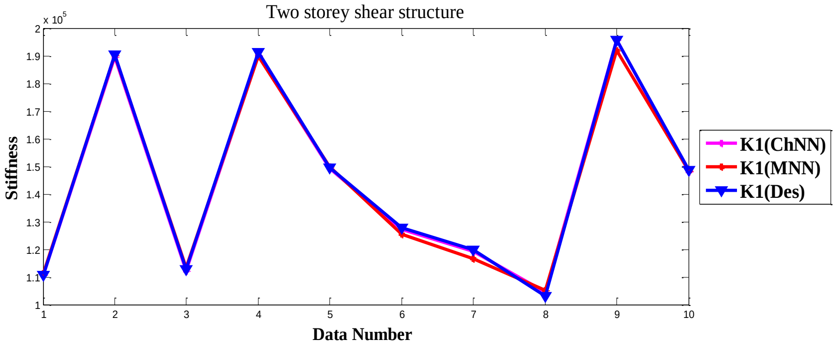

Figure 7.2(a): Comparison among ChNN, MNN and desired values of k_1 for a two storey shear buildings

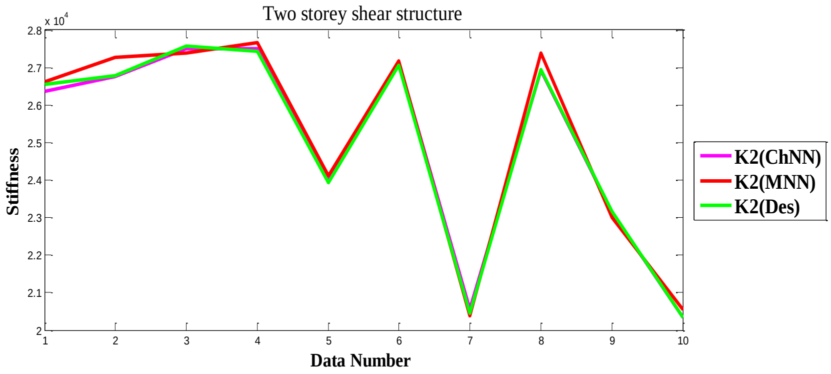

Figure 7.2(b): Comparison among ChNN, MNN and desired values of k_2 for a two storey shear buildings

Table 7.1(a): Comparison among ChNN (4th order polynomial), MNN and desired values of k_1 and k_2 for a two storey shear buildings

Data No.	k_1(ChNN)	k_1 (MNN)	k_1 (Des)	k_2 (ChNN)	k_2 (MNN)	k_2 (Des)
1	57469.3279	57411.7494	57464.7003	26107.958	26010.9841	26160.4468
2	62825.8206	62023.7934	62875.4127	24791.5608	25420.4664	24732.8885
3	92122.5463	91812.6547	92135.8628	23575.2357	23235.933	23516.5951
4	62786.9365	63303.2088	62714.1089	28335.1854	27456.8037	28308.2863
5	90718.5495	91555.9348	90714.2413	25805.8226	26867.7371	25852.6409
6	62195.4481	61271.8187	62176.2484	25493.6863	25991.0228	25497.2361
7	96494.7137	96137.939	96463.1812	29154.2619	28712.255	29171.9366
8	67421.5183	68115.7209	67499.1883	22884.5472	23019.0868	22858.3902
9	60029.3617	60044.0714	59829.7625	27580.1925	26905.8705	27572.0023
10	62575.7691	62485.4718	62554.1929	27498.2423	27144.8748	27537.2909

Table 7.1(b): Comparison between the ChNN (2nd order polynomial) and desired values of k_1 and k_2 for a two storey shear buildings

Data No.	k_1(ChNN)	k_1 (Des)	k_2 (ChNN)	k_2 (Des)
1	57528.0526	57464.7003	25525.1895	26160.4468
2	62903.8683	62875.4127	25303.1474	24732.8885
3	92188.8276	92035.8628	23033.4168	23516.5951
4	62799.80	62714.1089	28250.8314	28308.2863
5	90672.7283	90714.2413	25782.2893	25852.6409
6	62264.1615	62176.2484	25528.2606	25497.2361
7	96311.0601	96463.1812	29253.6982	29171.9366
8	67550.7045	67499.1883	22796.4967	22858.3902
9	60151.4052	59829.7625	27421.2854	27572.0023
10	62444.6976	62554.1929	27406.7464	27537.2909

Example 2. Five storey shear buildings:

Two problems have been solved for five storey shear buildings too. Masses for the first problem is considered as $m_1 = m_2 = m_3 = m_4 = m_5 = 36{,}000$ and the initial stiffness parameter as $k_1 = 50{,}000$ and $k_2 = k_3 = k_4 = k_5 = 20{,}000$. For second problem the masses are taken as $m_1 = 106{,}000,\ m_2 = 102{,}000,\ m_3 = 90{,}000,\ m_4 = m_5 = 80{,}000$ and the stiffness parameters are taken as $k_1 = 650{,}0000,\ k_2 = 450{,}0000,\ k_3 = k_4 = 220{,}0000,\ k_5 = 150{,}0000$. We have generated 40 sets of data of frequency parameters. Again different orders of Chebyshev polynomials are taken to optimize the error so as to get the required accuracy of 0.001. For first problem, comparison among ChNN, MNN and the desired values for 10 typical data for fifth floor has been plotted in Figure 7.3. After training, 10 trained data among 40 data are given in tables. Comparison of the ChNN and the desired values for second problem for first to third floors are incorporated in Table 7.2(a) and for fourth and fifth floor in Table 7.2(b).

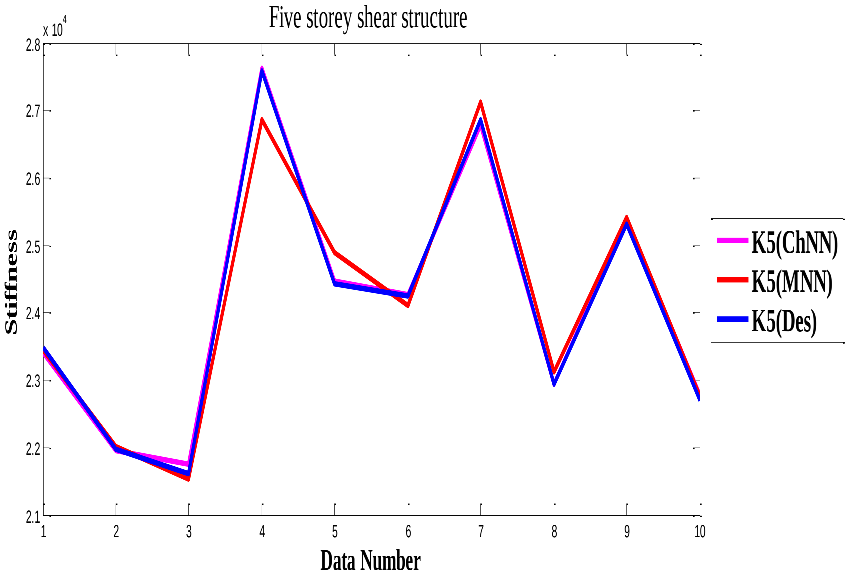

Figure 7.3: Comparison among ChNN, MNN and desired values of k_5 for a five storey shear buildings

Table 7.2(a): Comparison between the ChNN (with 4th order polynomial) and desired values of k_1, k_2 and k_3 for a five storey shear buildings

Data No.	k_1 (ChNN)	k_1 (Des)	k_2 (ChNN)	k_2 (Des)	k_3 (ChNN)	k_3 (Des)
1	6881531.6286	6881569.296	5355840.5633	5355821.8396	3589678.5908	3589606.5782
2	7017115.8831	7017164.5028	6432151.1137	6432105.4992	3052967.4767	3052911.0714
3	8295709.788	8295731.3697	5740122.9586	5740110.1083	3872579.1716	3872540.8414
4	7686709.9245	7686723.7208	5890704.5615	5890779.9038	3662727.0359	3662774.1323
5	7507624.4942	7507680.1637	5940326.3145	5940329.224	2920072.5968	2920062.0842
6	7725635.3312	7725619.1777	5193705.9015	5193790.3806	3108467.4182	3108424.7437
7	8138837.0332	8138844.4845	5533973.5232	5533980.8411	2972706.8871	2972779.7972
8	7563772.0232	7563778.339	5613303.7103	5613389.264	3751104.4253	3751109.2845
9	6904186.4148	6904150.1931	4812952.7444	4812990.4382	3668523.5351	3668542.2114
10	7407732.3345	7407786.9314	5624166.4634	5624112.0911	3060557.3664	3060555.6901

Table 7.2(b): Comparison between the ChNN (with 4th order polynomial) and desired values of k_4 and k_5 for a five storey shear buildings

No. of Data	k_4 (ChNN)	k_4 (Des)	k_5 (ChNN)	k_5 (Des)
1	3587542.3795	3587505.1483	3168395.9688	3168378.1221
2	4090483.6141	4090426.9743	1531271.7699	1531289.3854
3	3768496.8374	3768465.1965	3227407.2961	3227421.7301
4	3611193.2463	3611143.7163	1656141.9502	1656138.1061
5	2418663.7157	2418668.479	2838033.1023	2838085.1802
6	2979813.0581	2979861.3143	2500423.598	2500422.6486
7	3381877.9267	3381809.4608	1935961.4321	1935987.5975
8	3118793.0405	3118760.0959	2643205.0016	2643231.4508
9	2300606.817	2300679.9735	1744356.3896	1744378.3018
10	2657340.6508	2657375.168	2842360.1153	2842332.466

Example 3. Ten storey shear buildings:

Finally, ten storey shear building have been considered and two problems are taken into consideration. The masses for each storey are considered as $m_1 = m_2 = \cdots = m_{10} = 36{,}000$. The initial stiffness parameter for first problem is considered as $k_1 = 50{,}000, k_2 = k_3 = \cdots k_{10} = 20{,}000$ and for second problem as $k_1 = 100{,}000, k_2 = 50{,}000, k_3 = \cdots k_{10} = 20{,}000$. 50 data sets of frequency parameters are generated for training. The training is done using Chebyshev polynomial of second order. The figure plots of eighth and tenth storey for comparing the ChNN, MNN and the desired values for 10 data are shown in Figures 7.4(a) and 7.4(b) for the first problem. Table 7.3 shows the comparison between ChNN and desired for eighth and tenth floor for second problem with 10 data. The CPU time for the first problem using ChNN is 445.945 secs and with MNN it is 1134.877 secs. Similarly for second problem the CPU time for ChNN is 216.391 secs and for MNN is 1120.057 secs.

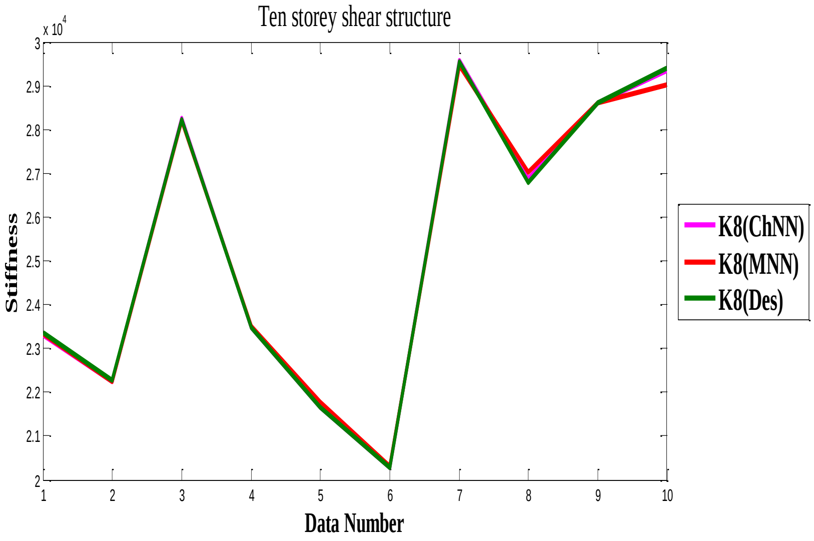

Figure 7.4(a): Comparison among ChNN, MNN and desired values of k_8 for a ten storey shear buildings

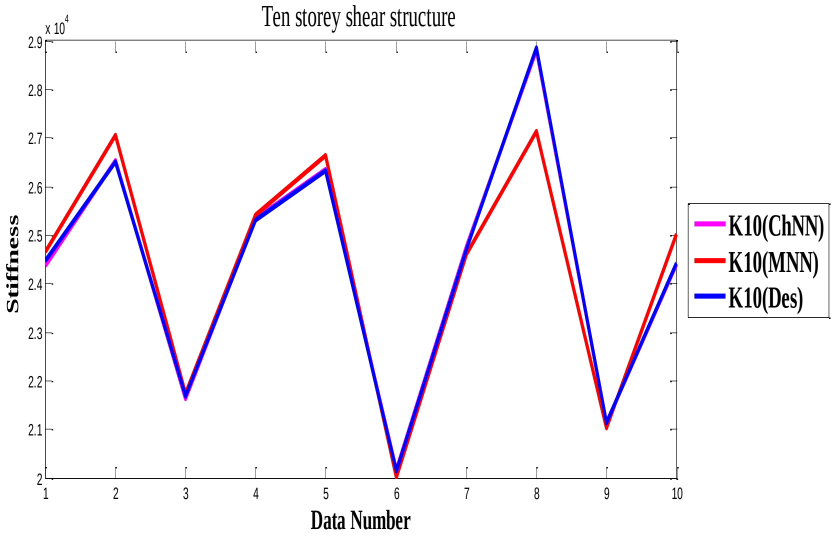

Figure 7.4(b): Comparison among ChNN, MNN and desired values of k_{10} for a ten storey shear buildings

Table 7.3: Comparison between the ChNN (with 2nd order polynomial) and desired values of k_8 and k_{10} for a ten storey shear buildings

Data No.	k_8 (ChNN)	k_8 (Des)	k_{10} (ChNN)	k_{10} (Des)
1	28439.8549	28407.1726	20751.0908	20758.5429
2	22511.5318	22542.8218	20552.6015	20539.5012
3	28132.7532	28142.8483	25318.4456	25307.9755
4	22490.4973	22435.2497	27722.4485	27791.6723
5	29235.2651	29292.6362	29372.4103	29340.1068
6	23494.9922	23499.8377	21291.9164	21299.0621
7	21921.5911	21965.9525	25622.4117	25688.2366
8	22508.426	22510.8386	24663.5239	24693.9064
9	26141.9975	26160.4468	19946.1473	20119.0207
10	24743.1849	24732.8885	23318.3256	23371.2264

7.4.2. Legendre and Hermite Neural Network Based Results

The orthogonal polynomials considered for present Functional Link neural network are Legendre and Hermite polynomial of 3rd order. In order to validate the present method, testing has been done for three and six storey shear buildings. After training, the converged weights are stored and these converged weights are used then for testing.

Example 1. Three storey shear building:

As mentioned earlier the frequency parameters are taken as the inputs and stiffness parameters are taken as the outputs. Here two problems have been solved. For the first problem the storey mass are taken as $m_1 = m_2 = m_3 = 36000$ and the initial stiffness parameters as $k_1 = 100000$, $k_2 = 50000$ and $k_3 = 20000$ Ref. [115]. Then for the second problem mass are considered as $m_1 = 106000$, $m_2 = 102000$ and $m_3 = 90000$ and the stiffness are taken as $k_1 = 6800000$, $k_2 = 4500000$ and $k_3 = 4500000$. Here, 50 sets of data for frequency have been obtained from these initial values of stiffness and masses. These 50 patterns are used for training the FLNN model. Legendre and Hermite polynomials of 3rd order are used as function expansion to obtain minimum error and to reach an accuracy of 0.001. Training with MNN is also done with different hidden nodes in the hidden layer. Figures 7.5(a)-7.5(c) represent the comparisons between the LeNN, HNN, MNN and desired for first, second and third floor with 10 data for first problem. In Figures 7.5(a)-7.5(c), the MNN is shown with 5, 10 and 15 hidden nodes. Comparison between the LeNN, HNN and desired for second problem with 10 data are given in Table 7.4. It is found that LeNN and HNN take less computation time than MNN. The CPU time for the first problem using LeNN is 207.29 secs and with HNN is 202.83 secs. The CPU time for MNN with 5 nodes is 405.55 secs, with 10 nodes are 584.12 and with 15 nodes are 672.91 secs. Similarly for second problem the CPU time for LeNN is 197.01 secs and for HNN it is 190.44 secs.

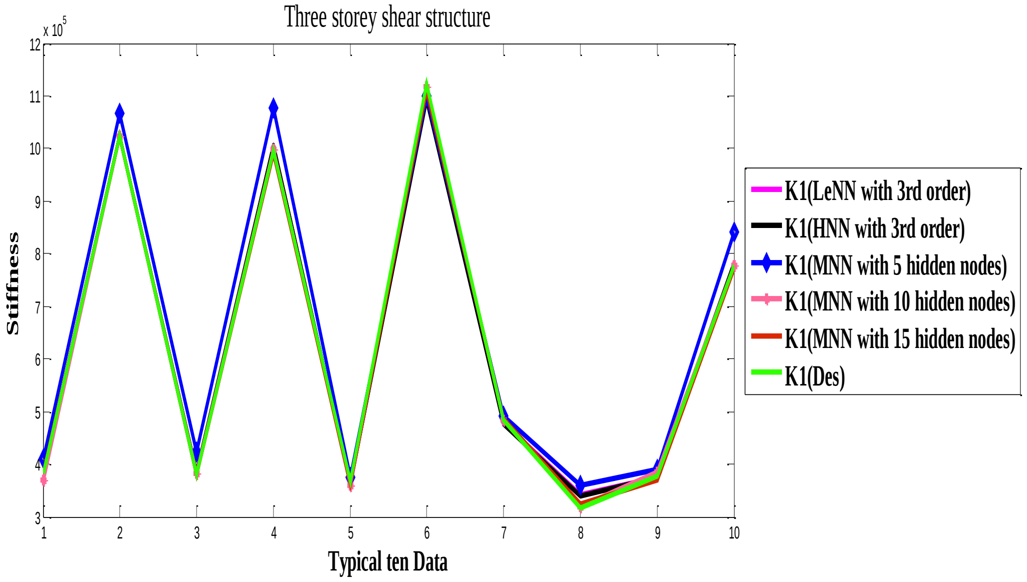

Figure 7.5(a): Comparison among LeNN, HNN, MNN and desired values of k_1 for three storey shear building

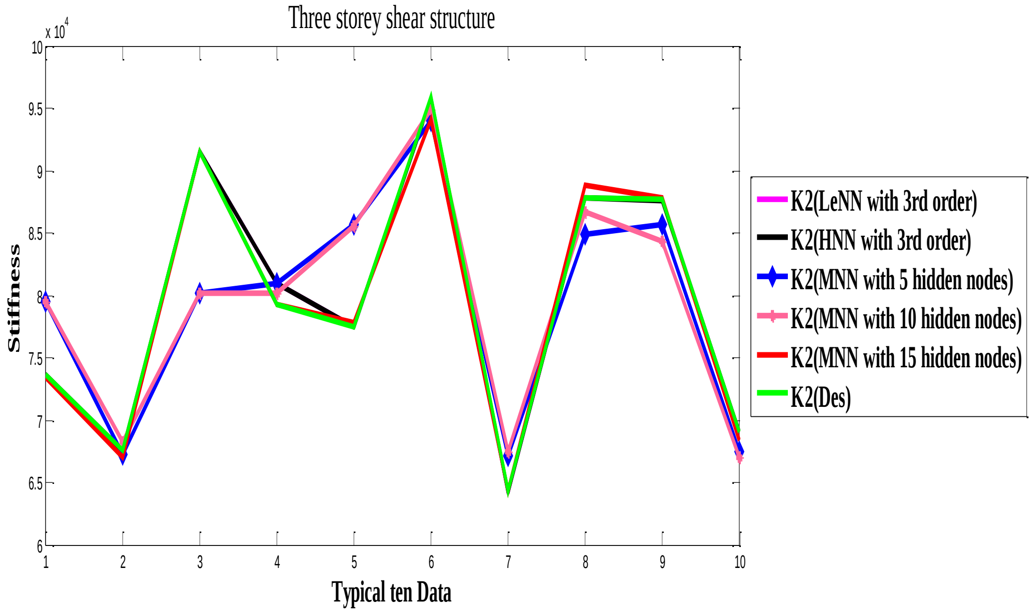

Figure 7.5(b): Comparison among LeNN, HNN, MNN and desired values of k_2 for three storey shear building

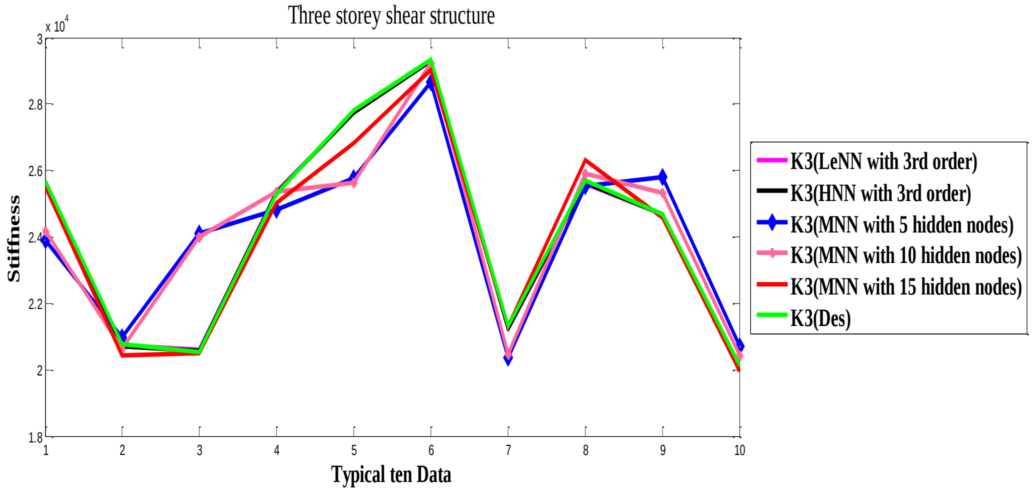

Figure 7.5(c): Comparison among LeNN, HNN, MNN and desired values of k_3 for three storey shear building

Table 7.4: Comparison among LeNN, HNN and desired values of k_1, k_2 and k_3 for three storey shear building

Typical Ten Data	k_1 (LeNN)	k_1 (HNN)	k_1 (Des)	k_2 (LeNN)	k_2 (HNN)	k_2 (Des)	k_3 (LeNN)	k_3 (HNN)	k_3 (Des)
1	8166891.815	8166879.071	8166831.733	6275596.137	6275550.341	6275541.908	5155138.085	5155137.549	5155130.868
2	8208099.075	8208056.278	8208094.860	5282337.258	5282334.683	5282365.990	5842512.778	5842584.511	5842528.740
3	7684621.666	7684644.146	7684610.826	6038223.915	6038253.902	6038228.774	5377268.866	5377294.804	5377289.965
4	6839178.554	6839118.267	6839155.247	5293559.749	5293552.063	5293583.034	6167087.557	6167012.382	6167001.191
5	7461742.628	7461791.407	7461715.760	6117012.838	6117058.356	6117028.191	6037790.441	6037798.108	6037708.504
6	7648662.380	7648642.835	7648618.993	6010110.639	6010149.282	6010154.198	4834564.691	4834558.095	4834507.091
7	7340517.277	7340531.536	7340540.846	5254741.044	5254764.92	5254791.089	6223938.823	6223997.93	6223960.957
8	7194163.625	7194110.573	7194107.596	4932043.569	4932077.61	4932037.831	6479793.211	6479720.851	6479744.307
9	8443480.397	8443451.051	8443442.369	6080830.900	6080854.309	6080814.435	5528899.2411	5528836.285	5528846.913
10	7659868.654	7659889.998	7659842.818	6398670.444	6398621.626	6398607.823	6268594.081	6268540.12	6268562.046

Example 2. Six storey shear building:

In this example, two problems of six storey shear building have been solved too. We assume masses for the first problem as $m_1 = m_2 = m_3 = m_4 = m_5 = m_6 = 36{,}000$ and the initial stiffness parameter as $k_1 = 100000$, $k_2 = 50{,}000$ and $k_3 = k_4 = k_5 = k_6 = 20{,}000$ Ref. [35]. For second problem the masses are taken as $m_1 = 106{,}000,\ m_2 = 102{,}000,\ m_3 = m_4 = 90{,}000,\ m_5 = m_6 = 80{,}000$ and the stiffness parameters are taken as

$k_1 = 6800000$, $k_2 = 4500000$, $k_3 = k_4 = 2200000$, $k_5 = k_6 = 1500000$. In this case, 40 sets of frequency parameter data are generated. Again Legendre and Hermite polynomials with 3rd order are taken to optimize the error so as to get the required accuracy of 0.001. Training with MNN is also done with different hidden nodes in the hidden layer. For first problem, comparison among LeNN, HNN, MNN and the desired values for 10 typical data among 40 data for sixth floor has been depicted in Figure 7.6. Figure 7.6 also shows MNN with 5, 10 and 15 hidden nodes. Typical 10 trained data among 40 data are also given in Tables 7.5(a) and 7.5(b) after training of the model. Comparison of the LeNN, HNN and the desired values for second problem for first to third floors are incorporated in Table 7.5(a) and for fourth to sixth floor in Table 7.5(b).

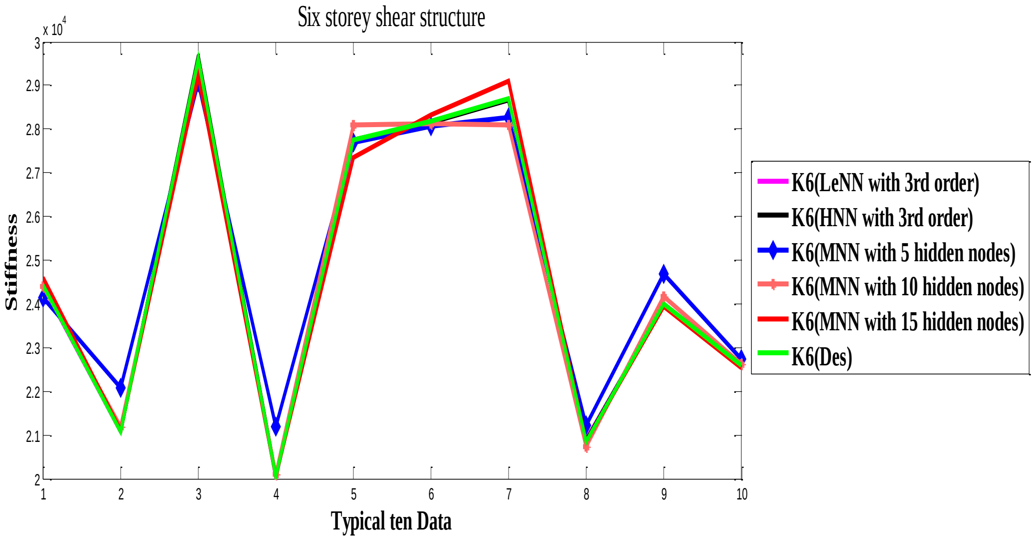

Figure 7.6: Comparison among LeNN, HNN, MNN and desired values of k_6 for six storey shear building

Table 7.5(a): Comparison among LeNN, HNN and desired values of k_1 , k_2 and k_3 for six storey shear building

Typical Ten Data	k_1 (LeNN)	k_1 (HNN)	k_1 (Des)	k_2 (LeNN)	k_2 (HNN)	k_2 (Des)	k_3 (LeNN)	k_3 (HNN)	k_3 (Des)
1	7660519.8106	7660546.539	7660555.690	4957397.7855	4957389.715	4957375.168	3542360.128	3542376.539	3542332.466
2	8187581.1215	8187551.797	8187505.148	6168315.1558	6168355.833	6168378.1221	3399119.4649	3399154.551	3399171.0962
3	8690486.5155	8690424.938	8690426.974	4531232.9757	4531207.659	4531289.3854	2311903.0292	2311905.159	2311952.3148
4	8368492.8618	8368471.663	8368465.196	6227442.581	6227446.419	6227421.7301	2312635.9764	2312660.667	2312686.0371
5	8211136.7641	8211151.927	8211143.716	4656157.5112	4656121.041	4656138.1061	2505031.302	2505036.048	2505001.274
6	7018636.1162	7018672.196	7018668.479	5838058.5836	5838065.092	5838085.1802	2239264.2552	2239297.234	2239242.1335
7	7579802.6927	7579803.165	7579861.314	5500469.8454	5500465.043	5500422.6486	3070319.9162	3070379.354	3070351.0914
8	7981885.9134	7981872.37	7981809.460	4935946.6738	4935936.303	4935987.5975	3864452.6906	3864442.281	3864442.9506
9	7718700.8278	7718731.693	7718760.095	5643207.4739	5643203.455	5643231.4508	3434791.0545	3434791.949	3434780.3429
10	7900611.1252	7900673.426	6900679.973	4744393.8144	4744393.957	4744378.301	3240252.9808	3240230.646	3240258.8306

Table 7.5(b): Comparison among LeNN, HNN and desired values of k_4, k_5 and k_6 for six storey shear building

Typical Ten Data	k_4 (LeNN)	k_4 (HNN)	k_4 (Des)	k_5 (LeNN)	k_5 (HNN)	k_5 (Des)	k_6 (LeNN)	k_6 (HNN)	k_6 (Des)
1	3927703.480	3927785.987	3927736.445	3333671.057	3333615.089	3333642.540	2533973.913	2533970.098	2533958.029
2	2395302.526	2395348.251	2395395.836	3473929.169	3473921.922	3473936.549	1842065.569	1842028.289	1842096.035
3	4016138.966	4016110.094	4016104.406	2510235.769	2510237.67	2510266.203	3377167.146	3377100.724	3377115.728
4	2416089.907	2416057.6	2416033.388	2042841.718	2042887.521	2042843.248	2680976.512	2680999.196	2680966.354
5	3233924.212	3233987.713	3233993.516	1701570.859	1701548.9	1701501.023	2381246.247	2381211.714	2381269.361
6	2486386.514	2486394.746	2486312.044	2515601.986	2515668.984	2515697.661	3383877.666	3383846.444	3383837.860
7	3318776.074	3318768.873	3318741.144	2671260.261	2671298.86	2671218.251	2811851.186	2811887.815	2811827.640
8	2209111.417	2209145.828	2209159.247	3025735.607	3025757.716	3025774.191	2403874.964	2403881.418	2403891.418
9	3733393.608	3733315.762	3733363.997	1665925.489	1665943.233	1665925.298	3179319.638	3179367.93	3179394.841
10	3897470.604	3897416.124	3897418.452	2823169.686	2823176.461	2823192.386	2565234.048	2565229.185	2565247.004

Example 3. Eight storey shear building:

Finally, eight storey shear building have been considered and two problems are taken into consideration. The masses for each storey are considered as $m_1 = m_2 = \cdots = m_8 = 36{,}000$. In the first problem, the initial stiffness parameter has been considered as $k_1 = 100{,}000,\ k_2 = 50{,}000,\ k_3 = \cdots k_8 = 20{,}000$ Ref. [115] and for second problem as $k_1 = 50{,}000,\ k_2 = k_3 = \cdots k_8 = 20{,}000$. 50 data sets of frequency parameters are generated for training. The training is done with Legendre, Hermite polynomial of 3rd order and MNN. The figure plots of seventh and eighth storey for comparing the LeNN, HNN, MNN and the desired values for 10 data are shown in Figures 7.7(a) and 7.7(b) for the first problem. Figures 7.7(a) and 7.7(b) show the MNN with 5, 10 and 15 hidden nodes. Table 7.6 depicts the comparison between LeNN, HNN and desired for seventh and eighth floor for second problem with 10 data. The CPU time for the first problem using LeNN is 266.366 secs and with HNN is 259.465 secs. The CPU time for MNN with 5 nodes is 722.58 secs, with 10 nodes is 1372.57secs and with 15 nodes is 1784.23 secs. Similarly for second problem the CPU time for LeNN is 229.33secs and for HNN it is 216.67 secs.

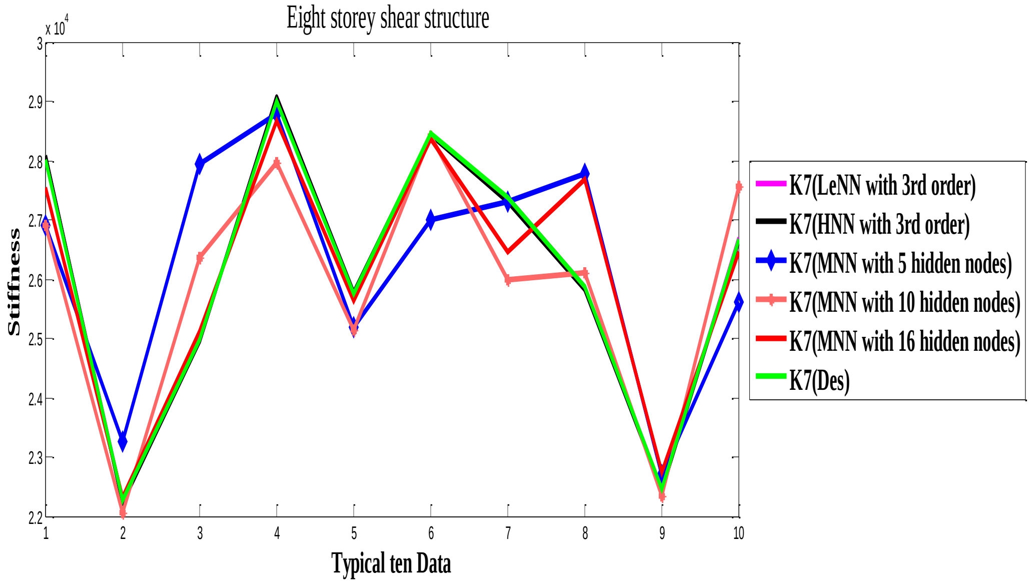

Figure 7.7(a): Comparison among LeNN, HNN, MNN and desired values of k_7 for eight storey shear building

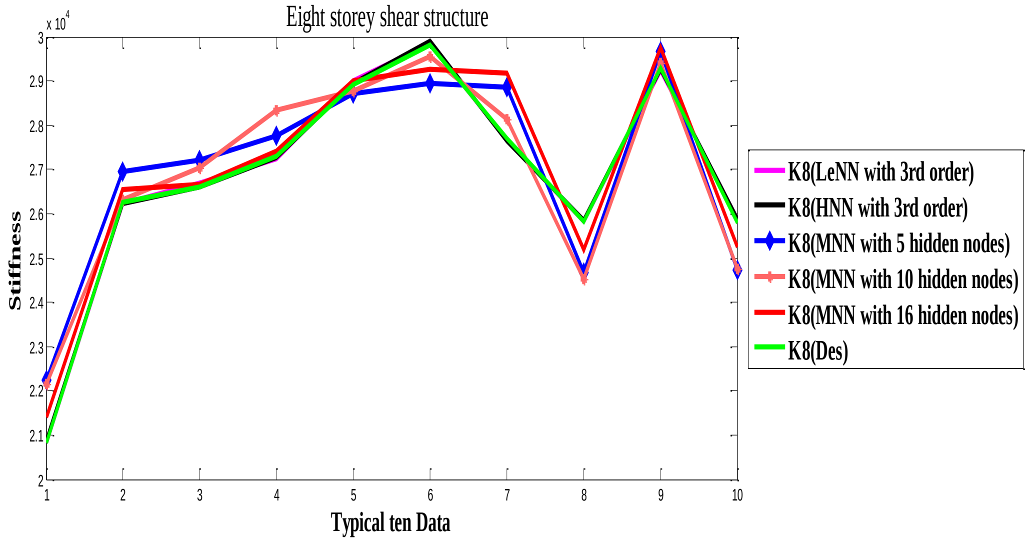

Figure 7.7(b): Comparison among LeNN, HNN, MNN and desired values of k_8 for eight storey shear building

Table 7.6: Comparison among LeNN, HNN and desired values of k_7 and k_8 for eight storey shear building

Typical Ten Data	k_7 (LeNN)	k_7 (HNN)	k_7 (Des)	k_8 (LeNN)	k_8 (HNN)	k_8 (Des)
1	2791576.5467	2791539.646	2791583.5342	2531520.0203	2531589.306	2531531.3064
2	2403225.2295	2403214.522	2403299.1454	3175663.0671	3175696.93	3175681.3817
3	2835888.5996	2835840.332	2835883.1989	3341549.659	3341520.379	3341580.216
4	2998304.5982	2998312.964	2998370.2578	2496425.2655	2496478.28	2496455.8589
5	2701787.9967	2701730.545	2701701.0524	2055232.9371	2055299.432	2055222.2445
6	2850861.5281	2850804.718	2850812.1359	2805085.2674	2805003.571	2805039.9215
7	2995958.358	2995962.706	2995954.9856	3334510.8628	3334523.808	3334597.6076
8	2433367.722	24333082.426	2433373.7681	2519609.9897	2519628.394	2519678.907
9	2800868.753	2800804.188	2800838.9384	3448332.152	3448309.02	3448382.9676
10	2312450.8118	2312437.821	2312462.3784	1894553.2781	1894503.591	1894557.8842

7.4.2.1 Testing

Further, for health monitoring and validation of the present method testing has been done. In Example 1, FLNN is trained with some random weights till the desired accuracy is reached and then the converged weights are stored. Now, to validate the model for checking the present health of a structure we consider a new structure with the same mass and with reduced stiffness parameters. Accordingly first the forward problem has been solved and frequency parameters are generated numerically. This is done because the experimental data is not available. Otherwise we may have the experimental frequencies which may be fed into the ANN model as input for the testing. For first scenario, three mass and stiffness parameter values for generating frequency parameters for three storey shear building (new) for testing are taken as $m_1 = m_2 = m_3 = 36000$ and $k_1 = 90000$, $k_2 = 40000$ and $k_3 = 10000$. Testing is done with the stored converged weights of the first problem of Example 1 considering the Legendre and Hermite polynomial of 3rd order as the functional link. Frequency parameters which are numerically generated from the above set of data are used as input for testing. As mentioned above, the converged weight from example 1 is taken directly and then new data are fed in the ANN model. Accordingly, comparison among the results from LeNN and HNN including desired for three storey shear buildings for typical 10 data are shown in Figures 7.8(a)-7.8(c). In second scenario, the mass and stiffness parameters are taken as $m_1 = 106{,}000, m_2 = 102{,}000, m_3 = m_4 = 90{,}000, m_5 = m_6 = 80{,}000$ $k_1 = 5800000$, $k_2 = 3500000$, $k_3 = k_4 = 1200000$, $k_5 = k_6 = 1000000$ for six storey shear buildings. For this case the stored converged weights of second problem of example 2 has been considered. Results have been computed from ANN models of Legendre and Hermite polynomial of 3rd order. Again comparison among the testing value of LeNN, HNN and desired are incorporated in Tables 7.7(a)-7.7(b) for typical 10 data.

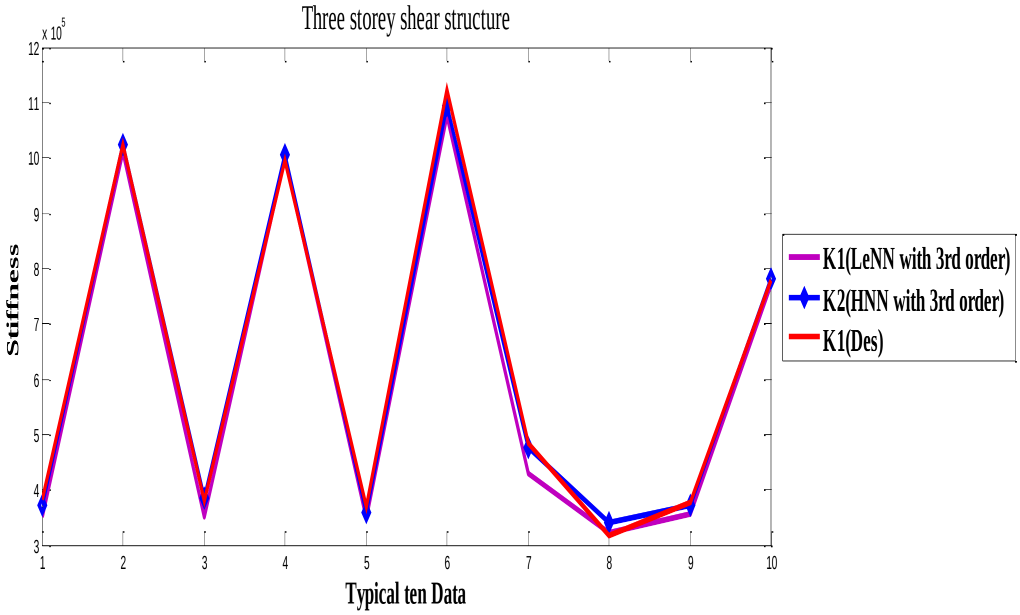

Figure 7.8(a): Comparison of LeNN, HNN results (testing) with desired of k_1 for three storey shear building

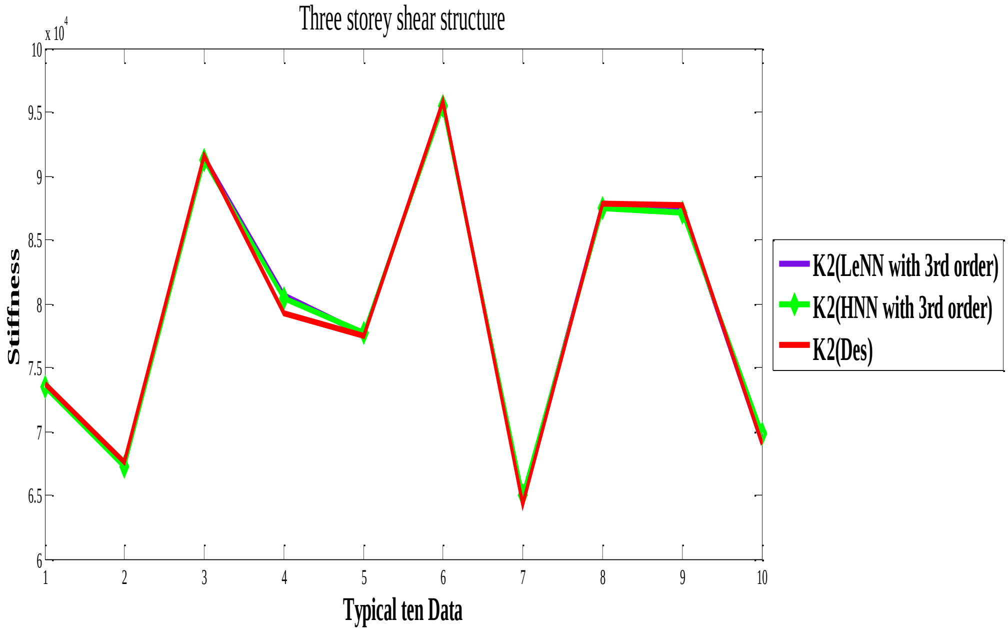

Figure 7.8(b): Comparison of LeNN, HNN results (testing) with desired of k_2 for three storey shear building

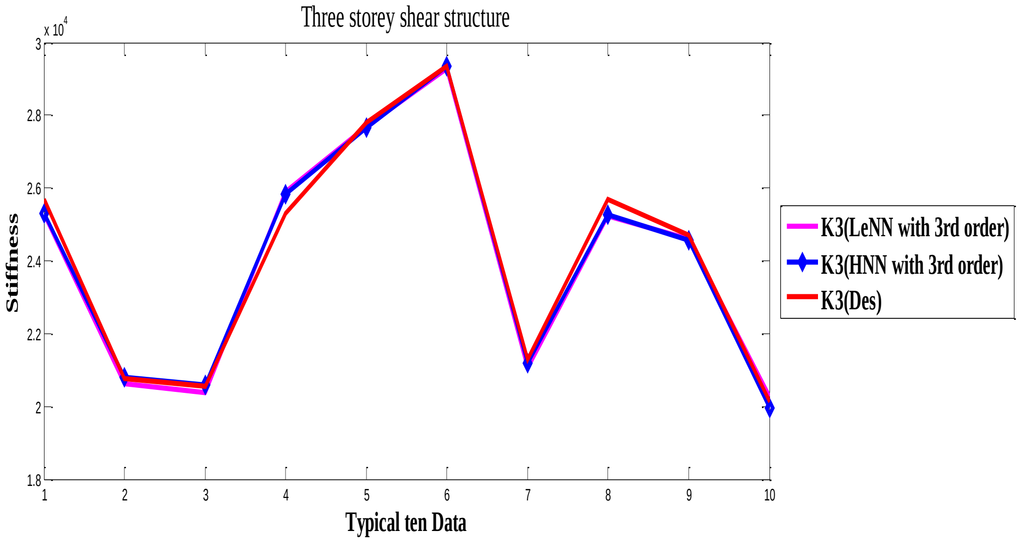

Figure 7.8(c): Comparison of LeNN, HNN results (testing) with desired of k_3 for three storey shear building

Table 7.7(a): Comparison of LeNN, HNN results (testing) with desired for k_1, k_2 and k_3 for six storey shear building

Typical Ten Data	k_1 (LeNN)	k_1 (HNN)	k_1 (Des)	k_2 (LeNN)	k_2 (HNN)	k_2 (Des)	k_3 (LeNN)	k_3 (HNN)	k_3 (Des)
1	7281705.604	7282873.366	7660555.6901	5091749.662	5089261.651	4957375.168	3539528.459	3541402.185	3542332.466
2	8104009.615	8104875.152	8187505.1483	6066462.148	6067141.075	6168378.1221	3862561.586	3862561.436	3399171.0962
3	8195975.586	8196247.486	8690426.9743	4540216.562	4540321.951	4531289.3854	2327918.52	2328029.272	2311952.3148
4	8062392.067	8062514.448	8368465.1965	6102717.864	6103253.256	6227421.7301	2352503.563	2352574.631	2312686.0371
5	7754437.444	7755830.379	8211143.7163	4721776.4	4720806.137	4656138.1061	2506250.835	2506681.515	2505001.274
6	7264481.185	7263072.887	7018668.479	5871558.789	5872129.429	5838085.1802	2206694.29	2206639.986	2239242.1335
7	7825538.897	7825685.674	7579861.3143	5442858.437	5444127.648	5500422.6486	3032317.191	3032011.805	3070351.0914
8	8400479.015	8402258.504	7981809.4608	4985943.012	4984873.106	4935987.5975	3743572.516	3744188.781	3864442.9506
9	7896787.265	7895762.805	7718760.0959	5635557.522	5635810.56	5643231.4508	3407731.331	3407582.893	3434780.3429
10	7279749.18	7277968.604	6900679.9735	4615826.117	4617312.229	4744378.3018	3320756.347	3319320.697	3240258.8306

Table 7.7(b): Comparison of LeNN, HNN results (testing) with desired for k_4, k_5 and k_6 for six storey shear building

Typical Ten Data	k_4 (LeNN)	k_4 (HNN)	k_4 (Des)	k_5 (LeNN)	k_5 (HNN)	k_5 (Des)	k_6 (LeNN)	k_6 (HNN)	k_6 (Des)
1	3933514.201	3933479.69	3927736.4459	3241254.953	3242415.979	3333642.5405	2562988.979	2561371.222	2533958.0294
2	2398954.564	2398902.74	2395395.8363	3445665.722	3445501.232	3473936.5496	1826701.711	1826731.373	1842096.0351
3	3958539.361	3958995.533	4016104.4064	2499196.676	2499157.629	2510266.2036	3347717.284	3348124.663	3377115.7287
4	2391670.448	2391765.131	2416033.3883	2039332.931	2039209.906	2042843.2488	2785002.809	2784979.29	2680966.3543
5	3252574.428	3252524.877	3233993.5162	1679590.263	1679863.346	1701501.0238	2356833.659	2356101.71	2381269.3615
6	2499213.546	2499244.339	2486312.0442	2520703.931	2520820.591	2515697.6617	3223676.972	3223543.746	3383837.8606
7	3382507.364	3382265.099	3318741.1448	2672972.389	2672980.792	2671218.2514	2666212.46	2666891.145	2811827.6405
8	2212209.103	2212189.412	2209159.2479	3024918.314	3025327.75	3025774.1918	2425152.55	2424474.889	2403891.4185
9	4015817.54	4015821.66	3733363.9972	1665849.608	1665916.812	1665925.2982	3154115.642	3154314.578	3179394.8414
10	3754633.14	3754783.794	3897418.4529	2878838.706	2877757.334	2823192.3862	2667374.57	2669492.059	2565247.0049

7.5 Conclusion

A novel method is presented here to estimate stiffness parameters of multistorey shear building using single layer multi-input and multi-output Functional Link neural network. Training with multilayer neural network (MNN) is also done and it is noted that ChNN, LeNN and HNN takes less computation time and gives good result as compared to MNN. The models need to have the knowledge of the initial design parameters namely stiffness and mass of the said problem and using these initial design parameters, frequencies for a structure may be computed by numerical simulation. These frequencies are used as inputs to train the FLNN model. Present structural parameter values of the shear structure may be identified by using the proposed FLNN. The converged FLNN model will have the capability to estimate the present stiffness parameter values for each floor. It may be seen that the present FLNN model are easy to implement with low computational complexity and are also more efficient than MNN. Various example problems of two, three, five, six, eight and ten storey shear buildings have been analyzed to show the efficacy and usefulness of the present FLNN model. Testing is also been done for three and six storey shear buildings with the stored converged weights of FLNN which validate the novelty of the present methods.

Chapter 8

Functional Link Neural Network Based System Identification through Seismic Data

In this Chapter application of Functional Link Neural Network (FLNN) for structural response prediction of tall buildings due to seismic loads have been proposed. The ground acceleration data is taken as input and structural responses of different floors of multi-storey shear buildings are considered as output. It is worth mentioning that handling of large earthquake data has become a great challenge in the design of tall structures viz. that of shear buildings. As such here, functional expansion block in FLNN has been used along with efficient Chebyshev and Legendre polynomials. Training is done with one earthquake data and testing is done with different intensities of other earthquake data and it is seen that FLNN can very well predict the structural response of different floors of multi-storey shear building subject to earthquake data. Results of FLNN are compared with Multilayer Neural Network (MNN) and it is found that the FLNN gives better accuracy and takes less computation time compared to MNN which shows the computational efficiency of FLNN over MNN. Numerical examples of two, five and ten storey buildings are considered and corresponding results are presented in the form of Tables and Plots.

8.1 Modelling For Response Analysis for Multi-Degree of Freedom System

When a multi-storey building is subjected to base excitation, then the governing equation of motion is written as

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\}=-[M]\{\chi\}\,\ddot{a} \tag{8.1}$$

where $[M]$ is a $n\times n$ mass matrix, $[K]$ is a $n\times n$ stiffness matrix of the structure and $[C]$ represents $n\times n$ damping matrix and $\{\chi\}$ is the influence co-efficient vector. Here $\{x\}$ is the displacement relative to the ground, $\{\ddot{x}\}$ is the response acceleration, $\{\dot{x}\}$ is the response velocity and $\ddot{a}$ is the earthquake ground acceleration. The global mass, stiffness and damping matrices are denoted as $[M]$, $[K]$ and $[C]$respectively and are given as below:

$$[M]=\begin{bmatrix} m_1 & 0 & \dots & \dots & 0 \\ 0 & m_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & m_{n-1} & 0 \\ 0 & \dots & \dots & 0 & m_n \end{bmatrix},$$

$$[K]=\begin{bmatrix} k_1+k_2 & -k_2 & 0 & \dots & 0 \\ -k_2 & k_2+k_3 & -k_3 & \dots & 0 \\ \dots & \dots & \cdots & \dots & \cdots \\ 0 & \dots & -k_{n-1} & k_{n-1}+k_n & -k_n \\ 0 & \cdots & \cdots & -k_n & k_n \end{bmatrix}$$

and

$$[C]=\begin{bmatrix} c_1+c_2 & -c_2 & 0 & \dots & 0 \\ -c_2 & c_2+c_3 & -c_3 & \dots & 0 \\ \dots & \dots & \cdots & \dots & \cdots \\ 0 & \dots & -c_{n-1} & c_{n-1}+c_n & -c_n \\ 0 & \cdots & \cdots & -c_n & c_n \end{bmatrix}.$$

Eq. (8.1) is a set of n-coupled ordinary differential equations. Modal analysis technique can be used to solve Eq. (8.1) if the system is linear with proportional damping. The modal analysis technique becomes more efficient for earthquake response analysis. Hence Eq. (8.1) can be reduced to n-modal equations as in Chakraverty et al. [89]

$$\ddot{x}_r + 2\xi_r\,\omega_r\,\dot{x}_r + \omega_r^2 x_r = -\ddot{a}\gamma_r \; r=1,2,\dots,n \tag{8.2}$$

where *n* (≤*N*) is the number of significant modes, ξ_r is the damping ratio and the modal coordinate x_r is related to the displacement of the i-th mass as

$$v_i = \sum_{r=1}^{n} \varphi_{ir} x_r \tag{8.3}$$

where φ_{ir} is the *i*-th component of the *r*-th mode-shape vector and γ_r is the modal participation factor. Eq. (8.3) represents the equation of motion of *n* SDOF system and the response is obtained from Duhamel integral. The Duhamel integral is written as Chakraverty [200]

$$x_r = -\frac{\gamma_r}{\omega_{Dr}} \int_0^t \ddot{a}(\tau)\exp\left[-\xi_r\,\omega_r(t-\tau)\right]\sin\left[\omega_r(t-\tau)\right]d\tau \tag{8.4}$$

Here, $\omega_{Di} = \omega_i \sqrt{1-\xi_i^2}$, where ω_{Di} ,ω_r and ξ_r are damped frequency, free vibration frequency and damping ratio respectively. The time history response of the *i*-th mass is then determined from Eq. (8.2) as

$$v_i\ (t) = \phi_{i1}\, x_1(t) + \phi_{i2}\, x_2\,(t) + \cdots \tag{8.5}$$

The floor responses of multi-storey shear structures viz. displacement is obtained from the above Eq. (8.5). The FLNN architecture is constructed with Chebyshev and Legendre polynomials, considering the ground acceleration data as input and the storey response as output, obtained from the above solution for each time step.

8.2 Functional Link Neural Network

Functional Link Neural Network is a single layer neural network in which the hidden layer is removed and the input vectors are expanded to higher dimensional vectors by some orthogonal polynomials. The FLNN model consists of mainly two parts, one is functional expansion part and the other is learning part. Due to the absence of hidden layer, FLNN is computationally more efficient and faster than MNN. Let us consider *i* input vector denoted as $Y = \{y_1, y_2, \ldots, y_i\}^T$. Thus, the enhanced input vector can be expressed as $Y^T = U(Y)$, where $U = [u_1(Y), u_2(Y), \ldots, u_N(Y)]$. Here $\{U\}_{k=1}^N$ are set of suitable functions. These functions may be considered as set of orthogonal polynomials viz. Chebyshev and Legendre polynomials. Structure of single layer FLNN model with single input-output is considered here and their learning algorithms have been discussed below. The single input-output FLNN architecture is shown in Figure 8.1.

8.2.1 Learning Algorithm of Functional Link Neural Network (FLNN)

In FLNN, the weights are updated to minimize a given cost function. Feed forward and error back propagation algorithm is used for learning. Error back propagation algorithm has been used to update the weights of FLNN. Inputs $Y_i = \ddot{a}_i$ are the ground acceleration and outputs $O_k = x_k$ are the structural responses of each floor of a multi-storey building. The linear sum S_k can be calculated as

$$\begin{gathered} S_k = \sum_{k=1}^{N} w_k U_k(Y) + \theta_k \\ k = i = \text{number of input nodes} \end{gathered} \tag{8.6}$$

where w_k are the weights, θ_k are the bias and $U_k(Y)$ are the expanded input vectors.

These $U_k(Y)$ are considered here as Chebyshev and Legendre polynomials.
The net output is given as

$$O_k = f(S_k)$$

As the earthquake data are both positive and negative, therefore bipolar sigmoidal function has been used as the activation function. The bipolar activation function is defined as

$$f = \frac{1 - \exp(S_k)}{1 + \exp(S_k)}$$

Cost function used for minimization of error is defined as

$$E = \frac{1}{2}[d_k - O_k]^2 = \frac{1}{2}e_k^2 \quad k = i \tag{8.7}$$

where d_k is the desired output, O_k is the target output and e_k is the error value. The error value is computed to obtain the desired accuracy. Weights are updated as follows:

$$w_k(New) = w_k(Old) + \Delta w_k \tag{8.8}$$

where change in weights are calculated as

$$\Delta w_k = \left[-\eta \frac{\partial E}{\partial w_k}\right] = \left[-\eta (d_k - O_k)\left(1 - O_k{}^2\right) U_k(Y)\right] \tag{8.9}$$

where η is the learning parameter. The same procedure to update the bias θ_k .

8.2.2 Structure of Chebyshev Neural Network

The structure of ChNN comprises of one input node, a functional expansion block consisting of Chebyshev polynomials and one output node. Each input vector is expanded to several terms of Chebyshev orthogonal polynomials so that they may be viewed as a new input vector in the functional expansion part. Chebyshev polynomial of k^{th} order is denoted by $T_k(y)$, where -1 <y< 1. First five Chebyshev polynomials may be written as

$$T_1(y) = 1$$
$$T_2(y) = y$$
$$T_3(y) = 2y^2 - 1$$
$$T_4(y) = 4y^3 - 3y$$
$$T_5(y) = 8y^4 - 8y^2 + 1$$

Higher order Chebyshev polynomials may be generated by the well known recursive formula

$$T_{k+1}(y) = 2xT_k(y) - T_{k-1}(y) \tag{8.10}$$

8.2.3 Structure of Legendre Neural Network

In Legendre neural network (LeNN), the suitable functions are taken as Legendre orthogonal polynomials in the expansion block. In functional expansion, each input is expanded to several terms using Legendre polynomials to have a new input vector. Here we have considered a single layer single input-output Legendre Neural Network (LeNN) model to find structural response. The Legendre polynomial of k^{th} order is denoted by $L_k(y)$, where -1 <y< 1. As such first five Legendre polynomials may be written as

$$\begin{aligned}
&L_1(y)=1\\
&L_2(y)=y\\
&L_3(y)=\frac{1}{2}\left(3y^2-1\right)\\
&L_4(y)=\frac{1}{2}\left(5y^3-3y\right)\\
&L_5 y=\frac{1}{8}\left(35y^4-30y^2+3\right)
\end{aligned}$$

Legendre polynomials of higher order may be obtained by the well known recursive rule

$$L_{k+1}(y)=\frac{1}{k+1}\left[(2k+1)\,y\,L_k(y)-k\,L_{k-1}(y)\right] \tag{8.11}$$

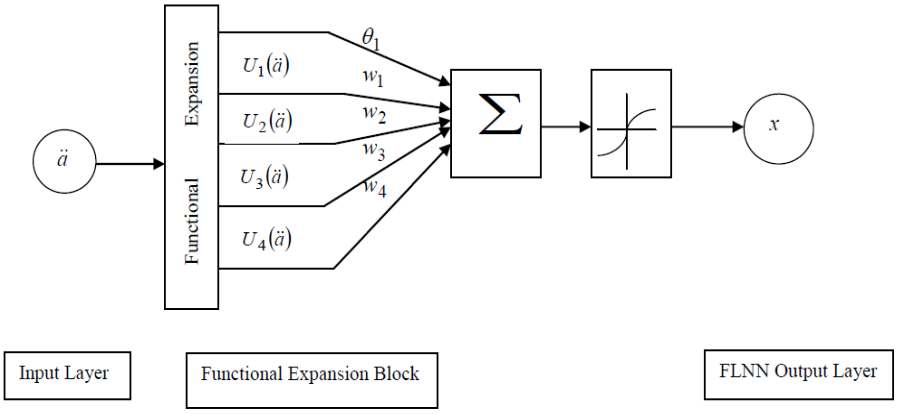

Figure 8.1 Single Layer Functional Link Neural Network

8.3 Results and Discussions

8.3.1 Training Case

The novel aim of the proposed method is to estimate the structural response of multi-degree freedom system from earthquake ground acceleration data. FLNN is trained and tested with different order polynomials. On the other hand, MNN is also trained and tested with different number of nodes in the hidden layer. Training is done with random weights and converged weights are stored for FLNN and MNN. With these stored weights testing has been done in order to get an accuracy of 0.001. For present investigation, examples of two, five and ten storey shear buildings have been considered for training with Chamoli earthquake data and structural response for ninth floor is predicted (testing) subject to different intensities of Uttarkashi earthquake data.

Example 1. Two storey shear buildings:

In this case, Chamoli earthquake data at Barkot (NE direction) has been considered for training with peak acceleration value as 19.58 cm/sec^2. A 2DOF system with natural frequency parameters 19.544 and 51.167 and damping ratio as 5% critical in both natural modes has been taken. Two separate models for two floors are used in training with a total time range of 0-10 secs. (500 data set). Responses of first and second storey are obtained numerically by solving the Duhamel integral that is Eq. (8.4) considering the ground acceleration data of Chamoli earthquake at Barkot (NE). First the training is done by Chebyshev neural network taking earthquake ground acceleration as input and structural response as output with random weights. After training by ChNN the converged weights are stored. Using the stored converged weights of ChNN, Legendre neural network is trained for different orders of polynomials. LeNN is also trained with random weights. It is found that LeNN trained with stored converged weights gives better accuracy and takes less computation time than trained with random weights. It is also seen that as we increase the order of these two orthogonal polynomials we get better accuracy. The training is also done with MNN. Out of 500 data set used for training, results for 10 typical data set are shown in all the tables below. Comparison of desired, ChNN with different order polynomials and MNN values for 10 data set for first floor of the two storey is shown in Table 8.1(a). Table 8.1(b) shows the results for desired, ChNN with different order polynomials and MNN values for 10 data set for second floor of the two storey. Tables 8.1(a) and 8.1(b) show how increase in the order of polynomials shows better accuracy for this case. The result comparison between desired and LeNN with stored converged weights and random weights (for first floor) has been incorporated in Table 8.1(c). Similarly for second floor the result is given in Table 8.1(d). Tables 8.1(c) and 8.1(d) show how learning with converged weights take less computation time and better accuracy than learning with random weights. The CPU time for ChNN is 86.30 secs.. LeNN with converged weights is 75.49 secs., random weights is 78.69 secs. and MNN is 284.20 secs. It is found that ChNN and LeNN take less computation time than MNN.

Table 8.1(a): Comparison between ChNN (with different order polynomial) and MNN for first storey

Data Set	Desired (cm/sec^2)	$T_2(y)$ (cm/sec^2)	$T_3(y)$ (cm/sec^2)	$T_4(y)$ (cm/sec^2)	$T_5(y)$ (cm/sec^2)	MNN (cm/sec^2)
1	-0.00044194	-0.00049865	-0.00052216	-0.00044154	-0.00044113	-0.00042112
2	-0.0018653	-0.0018279	-0.0017278	-0.0018951	-0.0018643	-0.0017858
3	-0.0013531	-0.0014483	-0.0013881	-0.0013437	-0.0013564	-0.0013726
4	-0.00055874	-0.00063637	-0.00065088	-0.00055988	-0.00055866	-0.00054473
5	-0.00021603	-0.00022461	-0.00026186	-0.00021852	-0.00021655	-0.00019099
6	0.00019361	0.00028091	0.0002343	0.00019321	0.00019369	0.00018613
7	0.0017298	0.0017846	0.0018271	0.0017799	0.0017229	0.0017509
8	0.0026065	0.0022142	0.0022834	0.0026496	0.0026015	0.0025816
9	0.0016876	0.0017565	0.0017965	0.0016834	0.0016895	0.0016834
10	0.00030355	0.00041458	0.00036915	0.00030257	0.00030388	0.00027847

Table 8.1(b): Comparison between ChNN (with different order polynomial) and MNN for second storey

Data Set	Desired (cm/sec^2)	$T_2(y)$ (cm/sec^2)	$T_3(y)$ (cm/sec^2)	$T_4(y)$ (cm/sec^2)	$T_5(y)$ (cm/sec^2)	MNN (cm/sec^2)
1	-0.00048403	-0.00054613	-0.00057188	-0.00048073	-0.00048486	-0.00046126
2	-0.002043	-0.0020019	-0.0018923	-0.002055	-0.002046	-0.0019557
3	-0.001482	-0.0015862	-0.0015203	-0.001417	-0.001494	-0.0015034
4	-0.00061195	-0.00069697	-0.00071286	-0.00061938	-0.00061195	-0.00059664
5	-0.00023661	-0.000246	-0.0002868	-0.00023838	-0.00023698	-0.00020914
6	0.00021205	0.00030766	0.00025662	0.00021278	0.00021288	0.00020392
7	0.0018945	0.0019546	0.0020011	0.0018493	0.0018998	0.0019178
8	0.0028547	0.002425	0.0025008	0.0028829	0.0028516	0.0028275
9	0.0018483	0.0019238	0.0019676	0.0018991	0.0018413	0.0018436
10	0.00033246	0.00045406	0.0004043	0.00033662	0.00033244	0.00030502

Table 8.1(c): Comparison between desired and LeNN with stored converged weights and random weights for first storey

Data Set	Desired (cm/sec^2)	With converged weights				With Random weights			
		$L_2(y)$ (cm/sec^2)	$L_3(y)$ (cm/sec^2)	$L_4(y)$ (cm/sec^2)	$L_5(y)$ (cm/sec^2)	$L_2(y)$ (cm/sec^2)	$L_3(y)$ (cm/sec^2)	$L_4(y)$ (cm/sec^2)	$L_5(y)$ (cm/sec^2)
1	-0.00044194	-0.00044865	-0.00044827	-0.00044145	-0.00044198	-0.00049867	-0.00052218	-0.00044156	-0.00044117
2	-0.0018653	-0.0018641	-0.0018576	-0.0018641	-0.0018651	-0.0018279	-0.0017279	-0.0018955	-0.0018648
3	-0.0013531	-0.0013852	-0.0013279	-0.0013497	-0.0013533	-0.0014485	-0.0013883	-0.0013439	-0.0013566
4	-0.00055874	-0.00055988	-0.00055831	-0.00055805	-0.00055877	-0.00063639	-0.00065085	-0.00055984	-0.00055869
5	-0.00021603	-0.00021705	-0.00021924	-0.00021655	-0.00021607	-0.00022463	-0.00026188	-0.00021855	-0.00021654
6	0.00019361	0.00019332	0.00019745	0.00019393	0.00019369	0.00028093	0.0002345	0.00019326	0.00019366
7	0.0017298	0.0017253	0.0017289	0.0017223	0.0017292	0.0017848	0.0018273	0.0017797	0.0017227
8	0.0026065	0.0026387	0.0026683	0.0026002	0.0026064	0.0022144	0.0022835	0.0026498	0.0026017
9	0.0016876	0.0016672	0.0016871	0.0016863	0.0016878	0.0017567	0.0017967	0.0016838	0.0016893
10	0.00030355	0.00030086	0.00030769	0.00030326	0.00030355	0.00041459	0.00036917	0.00030259	0.00030385

Table 8.1(d): Comparison between desired and LeNN with stored converged weights and random weights for second storey

Data Set	Desired (cm/sec^2)	With converged weights				With Random weights			
		$L_2(y)$ (cm/sec^2)	$L_3(y)$ (cm/sec^2)	$L_4(y)$ (cm/sec^2)	$L_5(y)$ (cm/sec^2)	$L_2(y)$ (cm/sec^2)	$L_3(y)$ (cm/sec^2)	$L_4(y)$ (cm/sec^2)	$L_5(y)$ (cm/sec^2)
1	-0.00048403	-0.00048613	-0.00048572	-0.00048463	-0.00048401	-0.00054615	-0.00057185	-0.00048075	-0.00048488
2	-0.002043	-0.002019	-0.002054	-0.002056	-0.002048	-0.0020016	-0.0018925	-0.002057	-0.002044
3	-0.001482	-0.001486	-0.001484	-0.001488	-0.001484	-0.0015865	-0.0015205	-0.001419	-0.001498
4	-0.00061195	-0.00061697	-0.00061048	-0.00061157	-0.00061197	-0.00069699	-0.00071289	-0.00061934	-0.00061197
5	-0.00023661	-0.0002366	-0.00023697	-0.00023664	-0.00023665	-0.000247	-0.0002865	-0.00023835	-0.00023695
6	0.00021205	0.00021766	0.00021211	0.00021253	0.00021203	0.00030768	0.00025665	0.00021276	0.00021285
7	0.0018945	0.0018546	0.0018702	0.0018901	0.0018943	0.0019548	0.0020014	0.0018497	0.0018996
8	0.0028547	0.002825	0.0028843	0.0028545	0.0028548	0.002427	0.0025009	0.0028826	0.0028518
9	0.0018483	0.0018238	0.0018362	0.0018498	0.0018482	0.0019239	0.0019677	0.0018993	0.0018415
10	0.00033246	0.00033406	0.00033175	0.00033273	0.00033248	0.00045408	0.0004048	0.00033665	0.00033245

Example 2. Five storey shear buildings:

Same earthquake data as mentioned in example 1 has been taken in this example. Again damping is assumed as 5% critical for this case. The training is done for 500 data set with a total time range of 0-10 secs.. ChNN and LeNN are trained with different order polynomials. LeNN is trained with the stored converged weights of ChNN. MNN is trained with15 hidden nodes in the hidden layer. Comparison between desired and ChNN of order 5 for fourth floor has been plotted in Figure 8.3(a). Figure 8.3(b) shows comparison between desired and LeNN (of order 5) for fourth floor. Similarly, comparison between desired and MNN for fourth floor has been shown in Figure 8.3(c). Comparison between desired and ChNN (of order 5) for fifth floor is shown Figure 8.3(d). The results of desired and LeNN (with order 5) for fifth floor is plotted in Figure 8.3(e). Finally Figure 8.3(f) depicts result comparison between desired and MNN for fifth floor. The CPU time for this case for ChNN, LeNN and MNN are 3659.84 secs., 3002.36 secs. and 256393.21 secs. respectively.

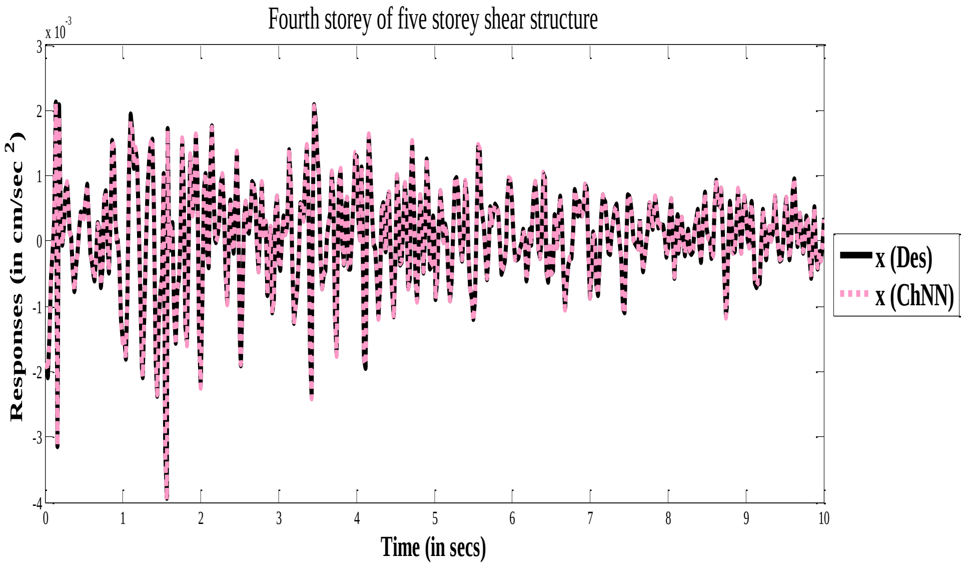

Figure 8.3(a): Comparison between desired and ChNN results for fourth floor

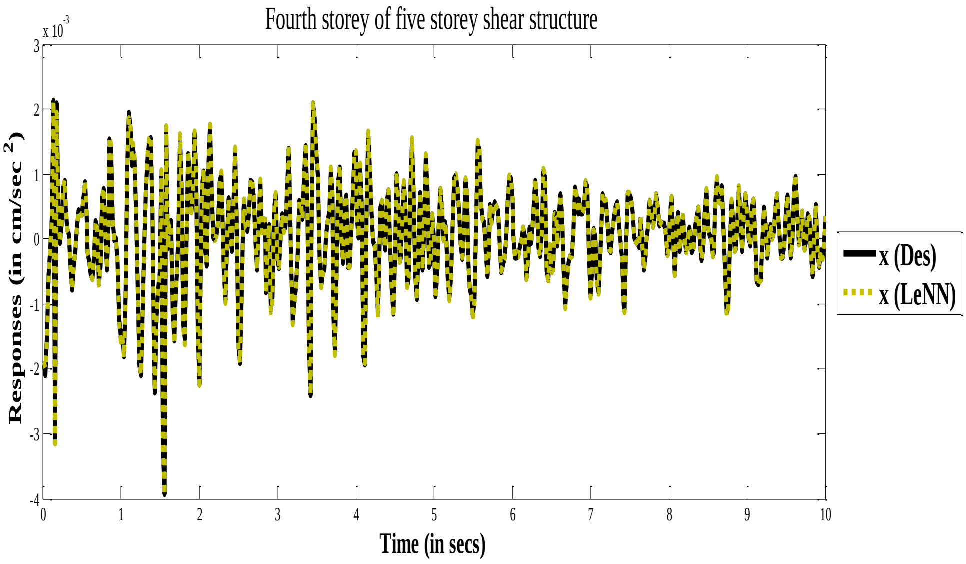

Figure 8.3(b): Comparison between desired and LeNN results for fourth floor

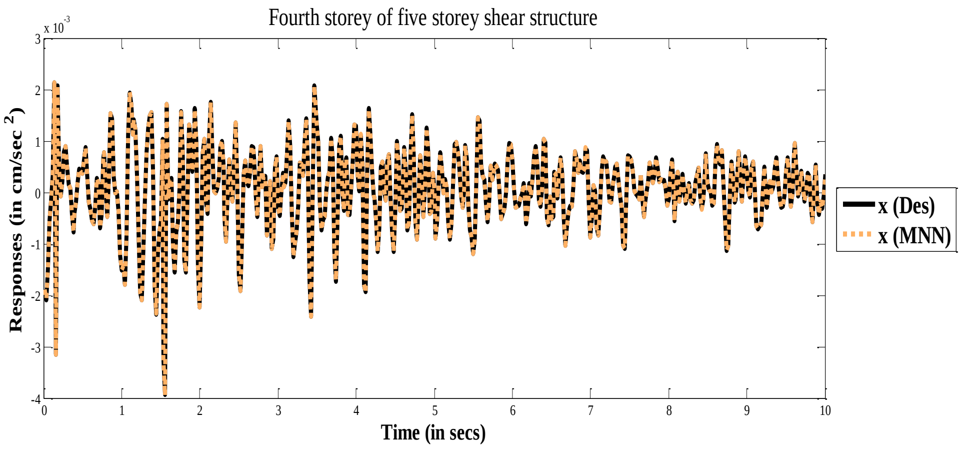

Figure 8.3(c): Comparison between desired and MNN results for fourth floor

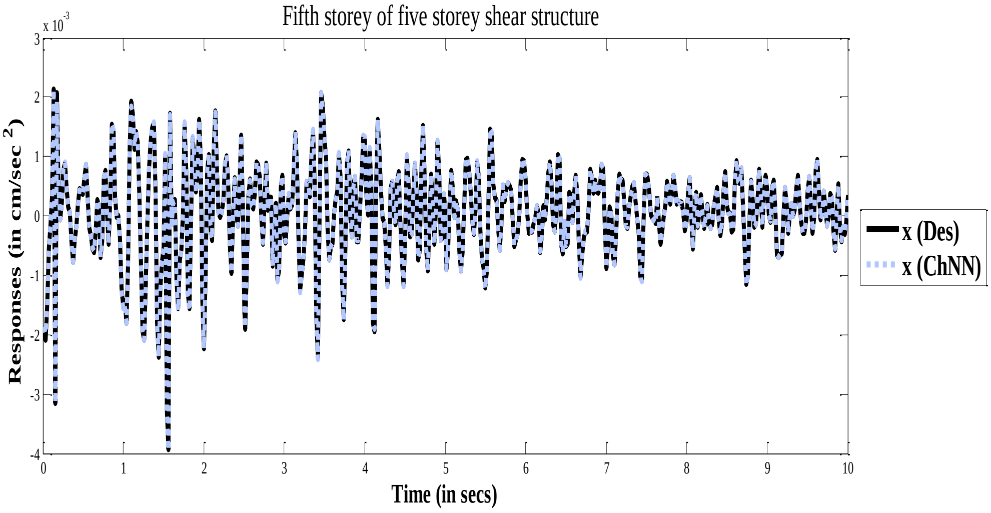

Figure 8.3(d): Comparison between desired and ChNN results for fifth floor

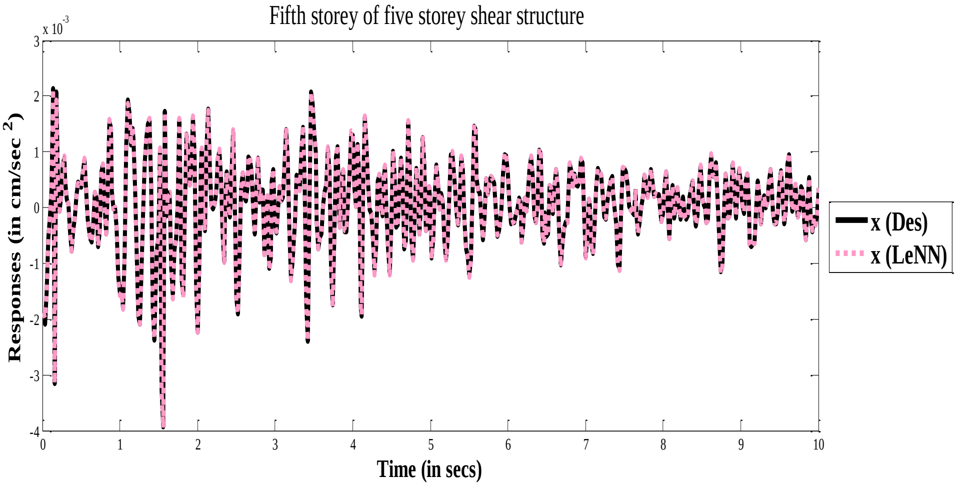

Figure 8.3(e): Comparison between desired and LeNN results for fifth floor

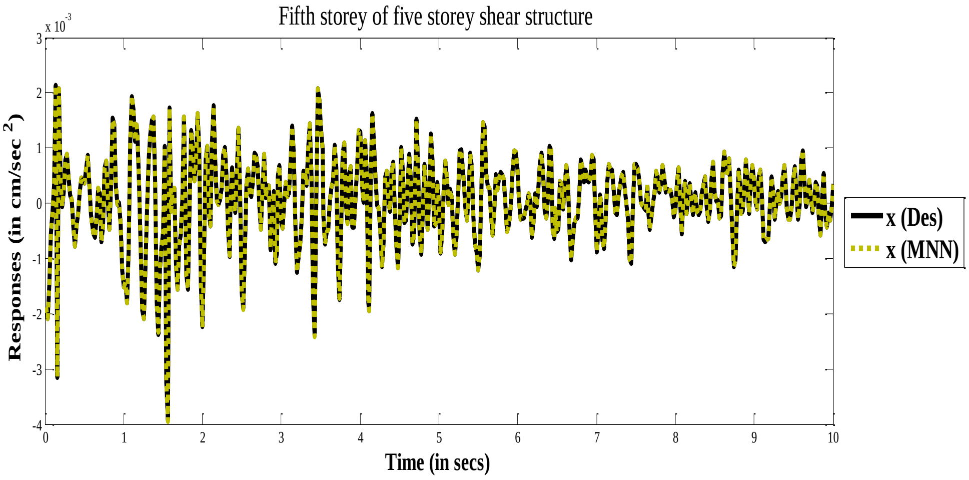

Figure 8.3(f): Comparison between desired and MNN results for fifth floor

Example 3. Ten storey shear buildings:

In this case, the damping ratio is again assumed as 5% critical for all natural modes for the entire storey. In the similar manner using same earthquake data as in cases 1 and 2, the training has been done for ten storey building for 500 data set with a total time of 0-10 secs.. Here ChNN and LeNN of order 5 have been considered and MNN has been trained with 15 hidden nodes. Comparison between desired, ChNN (of order 5), LeNN (with order 5) and MNN for eighth floor has been plotted separately in Figures 8.4(a)-8.4(c). Also results for tenth floor from ChNN, LeNN and MNN are shown separately in Figures 8.4(d)-8.4(f) respectively. The CPU time for this case for ChNN, LeNN and MNN are 4412.06 secs, 3012.16 secs and 328225.19 secs respectively.

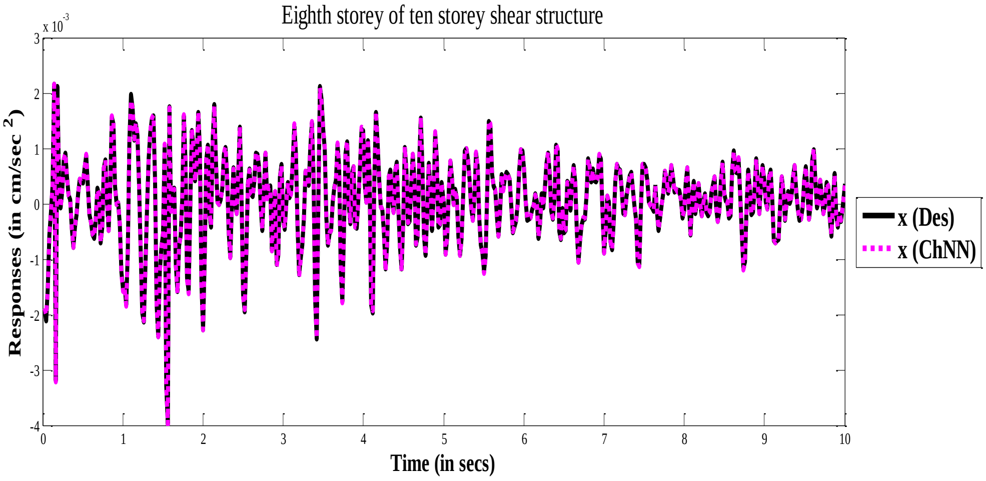

Figure 8.4(a): Comparison between desired and ChNN results for eighth floor

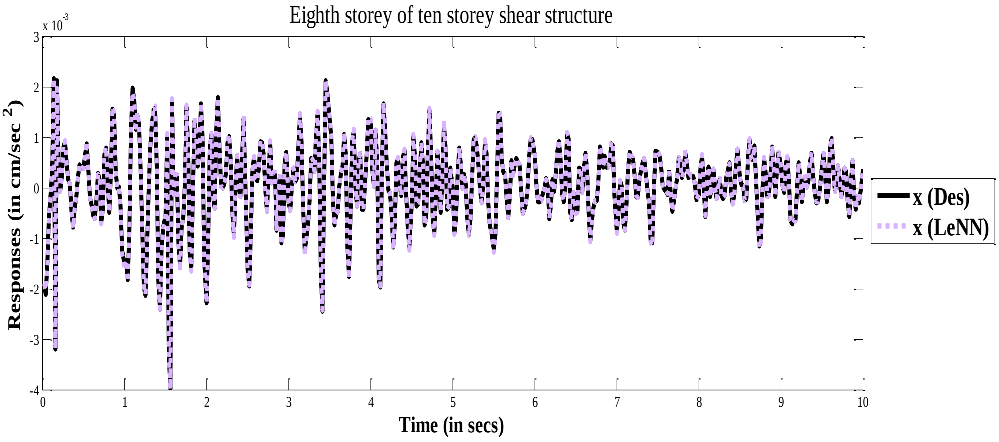

Figure 8.4(b): Comparison between desired and LeNN results for eighth floor

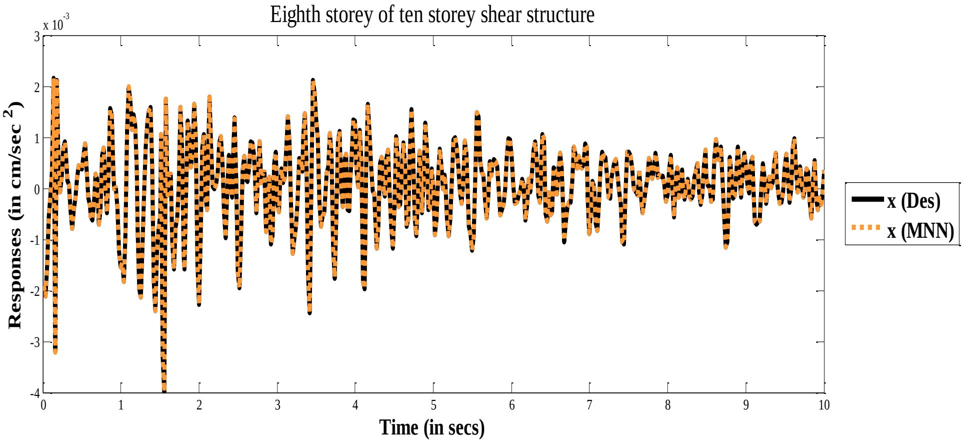

Figure 8.4(c): Comparison between desired and MNN results for eighth floor

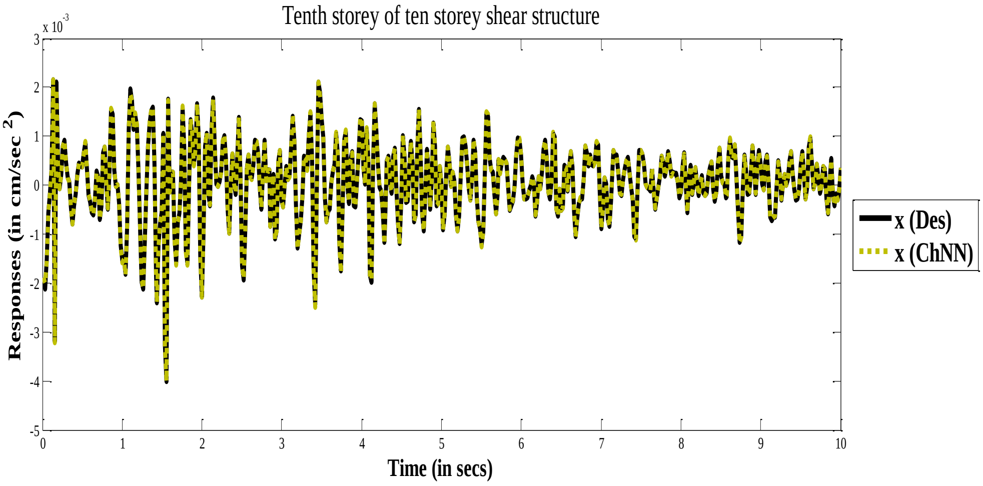

Figure 8.4(d): Comparison between desired and ChNN results for tenth floor

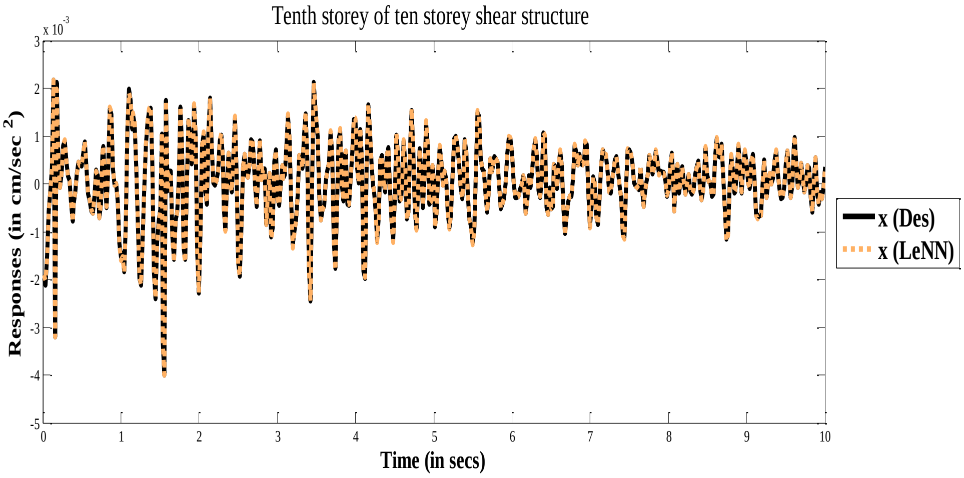

Figure 8.4(e): Comparison between desired and LeNN results for tenth floor

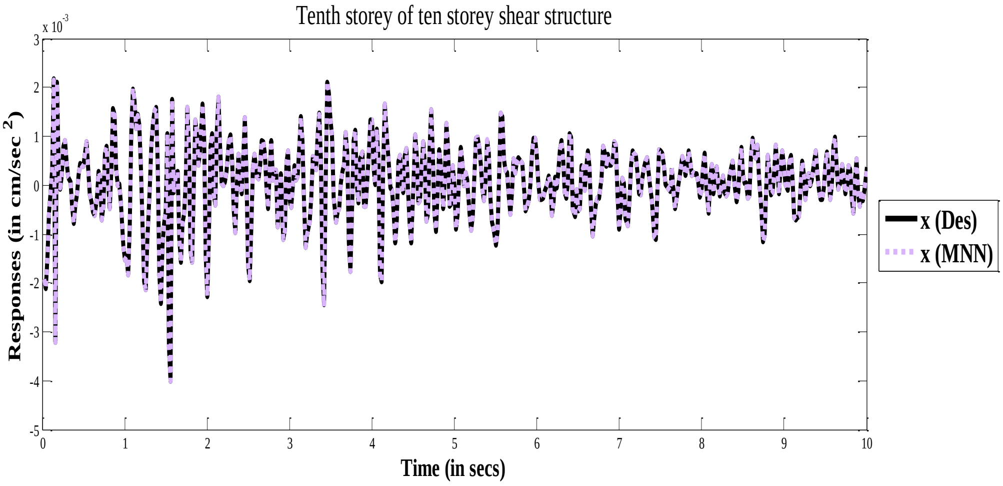

Figure 8.4(f): Comparison between desired and MNN results for tenth floor

8.3.2 Testing Case

Finally testing is done with various intensities (80% and 120%) of Uttarkashi earthquake data. First training with FLNN and MNN is done with Chamoli earthquake data for ten storey shear buildings and weights were stored. These stored converged weights are then used to predict storey response for 80% and 120% of the Uttarkashi earthquake data. Training is done with 500 data set with a time range of 0-10 secs but testing is done with 100 data set with a total time of 0-2 secs. Responses of ninth floor are found using the stored weights of FLNN and MNN. For testing orthogonal polynomials of order 5 are considered for FLNN and 10 hidden nodes are taken for MNN. The peak response values (testing values) from desired, ChNN, LeNN and MNN for 100 data set with different intensities of Uttarkashi earthquake are given in Table 8.2. Result comparisons among desired, ChNN, LeNN and MNN for ninth floor with 80% intensity of Uttarkashi earthquake for 100 data set (testing data) with time 0-2 secs are shown in Figures 8.5(a)-8.5(c) respectively. Similarly the results of desired, ChNN, LeNN and MNN for ninth floor with 120 % intensity of Uttarkashi earthquake for 100 data set (testing data) with time 0-2 secs. are plotted separately in Figures 8.6(a)-8.6(c). From these figures (Figures 8.5(a)-8.5(c) and Figures 8.6(a)-8.6(c)) and Table 8.2 one may conclude that FLNN can very well be used for response prediction of multistorey shear buildings subject to various intensities of seismic loads. It is also seen that computation is faster in FLNN as compared to MNN and it can give better accuracy even with the absence of hidden layer.

Table 8.2: Comparison among the desired, ChNN, LeNN and MNN peak response values (testing values)

Intensities (Uttarkashi at Barkot NE)	Desired (cm/sec^2)	ChNN (cm/sec^2)	LeNN (cm/sec^2)	MNN (cm/sec^2)
80%	0.0031215	0.0031213	0.0031225	0.003015
120%	0.004687	0.004682	0.004683	0.004685

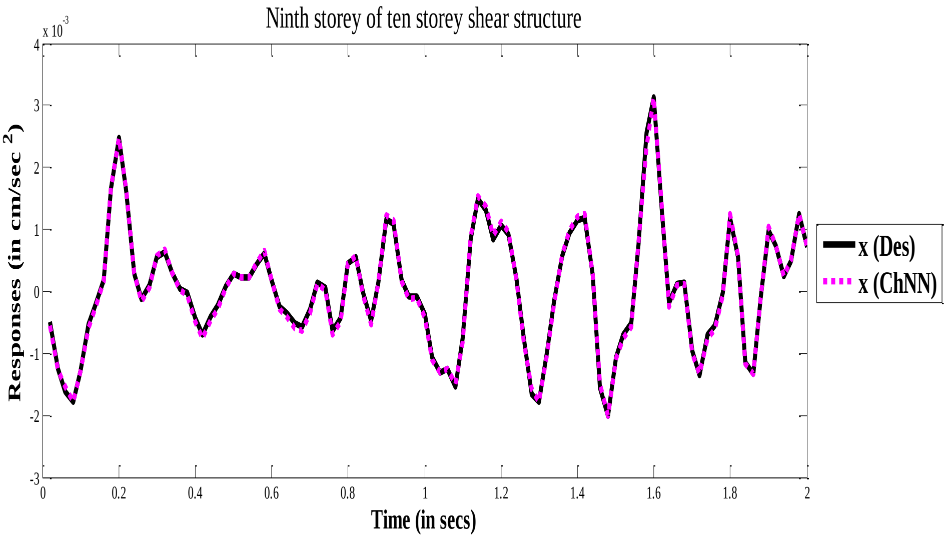

Figure 8.5(a): Comparison between desired and ChNN testing values (80%) for ninth floor

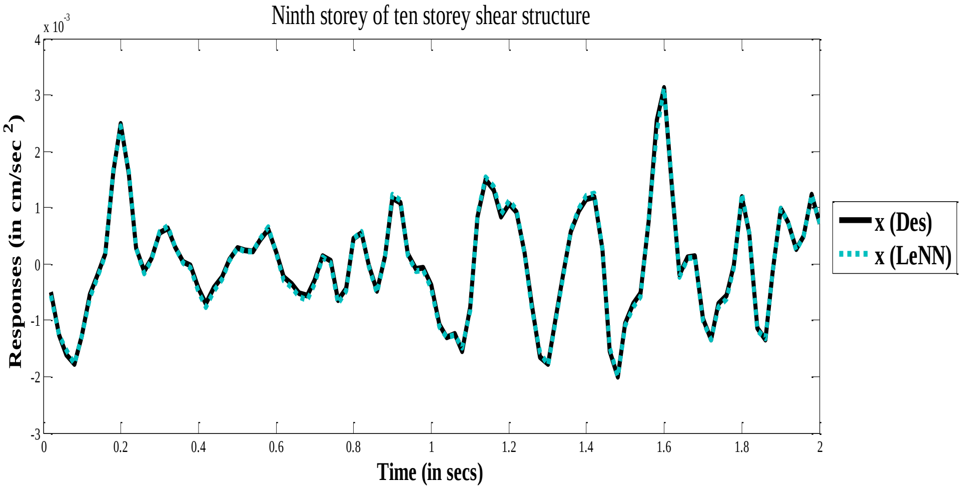

Figure 8.5(b): Comparison between desired and LeNN testing values (80%) for ninth floor

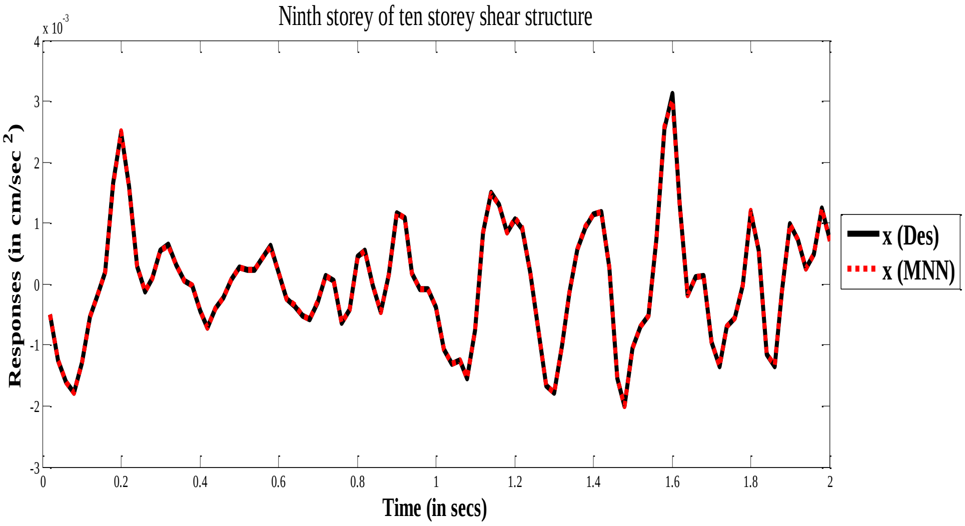

Figure 8.5(c): Comparison between desired and MNN testing values (80%) for ninth floor

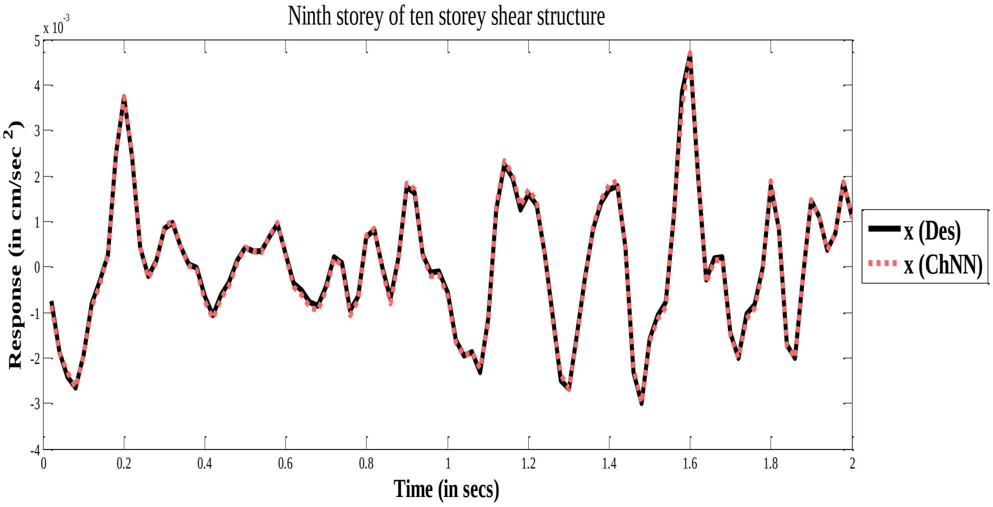

Figure 8.6(a): Comparison between desired and ChNN testing values (120%) for ninth floor

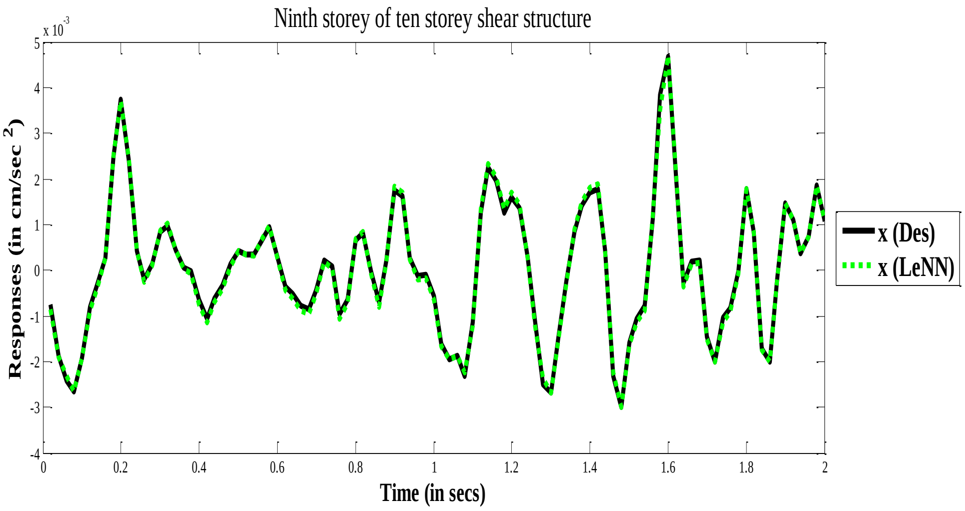

Figure 8.6(b): Comparison between desired and LeNN testing values (120%) for ninth floor

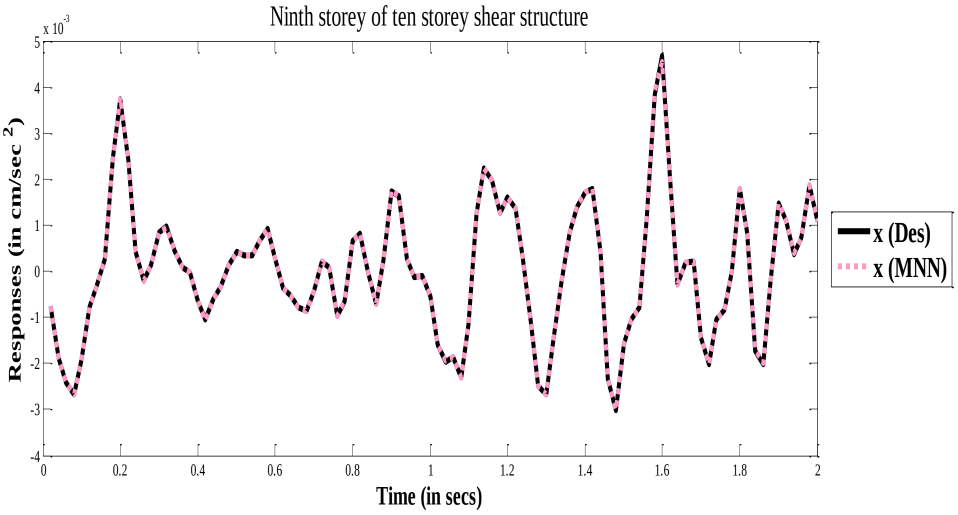

Figure 8.6(c): Comparison between desired and MNN testing values (120%) for ninth floor

8.4 Conclusion

Present paper estimates the structural response of multi storey shear buildings subject to seismic loads using FLNN. Training is done with ChNN, LeNN and MNN taking the input as ground acceleration and output as structural response of each floor. ChNN is trained with random weights for different order of polynomials and it is seen that for higher order of polynomial we get better accuracy. LeNN is trained for different order of polynomials by two ways, one way is by using the stored converged weights of ChNN and the other way is by considering the random weights. After the training is complete it is found that LeNN with stored converged weights gives better accuracy than random weights. Training is done with Chamoli earthquake data but for testing different intensities of Uttarkashi earthquake data have been used. With the stored weights (of training), testing is done to show the powerfulness of the present model. In testing it is found that FLNN can predict structural response for different storey subject to different earthquake loads. It is worth mentioning that FLNN gives better accuracy in all the cases and also found to be computationally more efficient than MNN as it takes less computation time.

Chapter 9

Interval Based Functional Link Neural Network for System Identification via Response Data

This Chapter presents a procedure to identify uncertain structural parameters of multistorey shear buildings by interval functional link neural network. The structural parameters are identified using response of the structure with both ambient and forced vibration. Here interval functional link neural network have been used to train interval data. The polynomials used in the functional link are Chebyshev and Legendre polynomials. These polynomials are taken in interval form.

9.1 Analysis and Modelling with Interval Case

The floor masses for this application problem are assumed to be $\left[\underline{m}_1, \overline{m}_1\right], \left[\underline{m}_2, \overline{m}_2\right], \ldots, \left[\underline{m}_n, \overline{m}_n\right]$ and the stiffness $\left[\underline{k}_1, \overline{k}_1\right], \left[\underline{k}_2, \overline{k}_2\right], \ldots, \left[\underline{k}_n, \overline{k}_n\right]$ are the structural parameters which are to be identified. It may be seen that all the mass and stiffness parameters are taken in interval form. The interval n-storey shear structure is already shown in Figure 3.1. Corresponding dynamic equation of motion for n-storey (supposed as n degrees of freedom) shear structure without damping may be written as

$$\left[\tilde{M}\right]\left\{\ddot{\tilde{X}}\right\}+\left[\tilde{K}\right]\left\{\tilde{X}\right\}=\left\{\tilde{F}(t)\right\} \tag{9.1}$$

where, $\left[\tilde{M}\right]=\left[\underline{M}, \overline{M}\right]$ and $\left[\tilde{K}\right]=\left[\underline{K}, \overline{K}\right]$ are interval mass and stiffness matrices and $\left\{\tilde{F}(t)\right\}=\left\{\underline{F}(t), \overline{F}(t)\right\}$ is the interval horizontal displacement forcing function.

Let us consider that the initial conditions in interval form are given by Eq. (9.2) and (9.3) as

$$\left\{\tilde{x}(0)\right\}=\left\{\underline{x}(0), \overline{x}(0)\right\}=\left\{\tilde{x}_1(0) \quad \tilde{x}_2(0) \quad \cdots \quad \tilde{x}_n(0)\right\}^T \tag{9.2}$$

$$\left\{\dot{\tilde{x}}(0)\right\}=\left\{\underline{\dot{x}}(0), \overline{\dot{x}}(0)\right\}=\left\{\dot{\tilde{x}}_1(0) \quad \dot{\tilde{x}}_2(0) \quad \cdots \quad \dot{\tilde{x}}_n(0)\right\}^T \tag{9.3}$$

Solution of Eq. (9.1) for free vibration with given interval values of mass and stiffness gives the corresponding interval eigenvalues and eigenvectors. These are denoted respectively by $\tilde{\lambda}_i$ and $\left\{\tilde{A}\right\}_i=\left\{\underline{A}, \overline{A}\right\}_i$, $i=1,\ldots,n$ where $\tilde{\omega}_i^2\left(=\tilde{\lambda}_i\right)$ are the system's interval natural frequency. It may be noted that the free vibration equation will be an interval eigenvalue problem. The interval eigenvalue and vector are obtained then by considering different sets of lower and upper stiffness and mass values. Although there exist different methods to handle interval eigenvalue problems but here the above procedure has been used so that we may handle the inverse of the matrices in crisp form separately as lower and upper value. And that is why now we will replace the '~' from all notations and will consider the case for lower form first and simultaneously for upper form. Hence the modal matrix for lower form $\{\underline{A}\}$ may be written as

$$[\underline{A}]=[\{\underline{A}\}_1 \quad \{\underline{A}\}_2 \quad \cdots \quad \{\underline{A}\}_n] \tag{9.4}$$

Denoting the diagonal matrix made up of the eigenvalues in lower form as $\underline{\lambda}_i$, as $[\underline{\lambda}]_{n\times n}$, a new set of co-ordinates in lower form $\{\underline{y}\}$ related to the co-ordinates $\{\underline{X}\}$ is introduced by the well known transformation

$$\{\underline{X}\}=[\underline{A}]\,\{\underline{y}\} \tag{9.5}$$

If the system (9.1) is subjected to an initial velocity only then substituting Eq. (9.5) in Eq. (9.1) for ambient vibration, the following equation is obtained for the response in lower form as:

$$\{\underline{X}\}=[\underline{A}][\underline{D}][\underline{\omega}]^{-1}\,[\underline{A}]^{-1}\,\{\underline{\dot{x}}(0)\} \tag{9.6}$$

whereas for the horizontal displacement in lower form we have the equation

$$\{\underline{\ddot{y}}\}+[\underline{\lambda}]\{\underline{y}\}=[\underline{P}]^{-1}\,[\underline{A}]^{T}\,\{\underline{F}(t)\} \tag{9.7}$$

where

$$[\underline{P}]=[\underline{A}]^{T}\,[\underline{M}][\underline{A}] \tag{9.8}$$

The final response for this case may be expressed in term of the original co-ordinates $\{\underline{X}\}$ after solving Eq. (9.7) for $\underline{y}$ and then putting in Eq. (9.5). In the similar manner we can compute for the upper form. The training patterns are now trained using Interval Functional Link Neural Network.

9.2 Interval Orthogonal Polynomials

In this section we introduce an interval version of orthogonal polynomials.

9.2.1 Interval Chebyshev Polynomial

For each natural number *m*, the first four degree of interval Chebyshev polynomials may be written as in Patrício et al. [204]

$$T_{0,m}(x)=\left[1-\frac{1}{m},1+\frac{1}{m}\right]$$

$$T_{1,m}(x)=\left[1-\frac{1}{m},1+\frac{1}{m}\right]x$$

$$T_{2,m}(x)=2\left[1-\frac{1}{m},1+\frac{1}{m}\right]x^2-\left[1-\frac{1}{m},1+\frac{1}{m}\right]1$$

$$T_{3,m}(x)=4\left[1-\frac{1}{m},1+\frac{1}{m}\right]x^3-3\left[1-\frac{1}{m},1+\frac{1}{m}\right]x$$

Interval Chebyshev polynomials of higher degree may be obtained by the well known recursive rule

$$T_{j+1,m}(x)=2xT_{j,m}(x)-T_{j-1,m}(x) \tag{9.9}$$

For each $m\in \mathrm{N}$ and $j\in \mathrm{N}$, we can say $T_{j,m}(x)$ as interval Chebyshev Polynomial

9.2.2 Interval Legendre Polynomial

For each natural number *m*, the first four degree of interval Legendre polynomials may be written as in Patrício et al. [204]

$$L_{1,m}(x)=\left[1-\frac{1}{m},1+\frac{1}{m}\right]$$

$$L_{2,m}(x)=\left[1-\frac{1}{m},1+\frac{1}{m}\right]x$$

$$L_{3,m}(x)=\frac{3}{2}\left[1-\frac{1}{m},1+\frac{1}{m}\right]x^2-\frac{1}{2}\left[1-\frac{1}{m},1+\frac{1}{m}\right]$$

$$L_{4,m}(x)=\frac{5}{2}\left[1-\frac{1}{m},1+\frac{1}{m}\right]x^2-\frac{3}{2}\left[1-\frac{1}{m},1+\frac{1}{m}\right]x$$

Interval Legendre polynomials of higher degree may be obtained by the well known recursive rule

$$L_{j+1,m}(x)=\frac{2j+1}{j+1}xL_{j,m}(x)-\frac{j}{j+1}L_{j-1,m}(x) \tag{9.10}$$

For each $m\in \mathrm{N}$ and $j\in \mathrm{N}$, we may say $L_{j,m}(x)$ as interval Legendre Polynomial

9.3 Learning algorithm of Interval Functional Link Neural Network (IFLNN)

In IFLNN, the interval weights are updated in order to minimize the cost function. Here interval error back propagation algorithm has been used for learning and for updating the weights of IFLNN. Inputs $\tilde{O}_i = \tilde{X}_i$ are the interval frequencies and outputs $\tilde{O}_J = \tilde{k}_J$ are the stiffness parameters. As such, linear sum $\tilde{z}_J$ can be calculated as

$$\tilde{Z}_J = \sum_{j=1}^{N} \tilde{w}_j \tilde{U}_j(X) + \tilde{\theta}_j \tag{9.11}$$

$J = i =$ number of input vectors and j is the degree of the polynomial

where $\tilde{w}_j$ are the interval weights, $\tilde{\theta}_j$ are the interval bias and $\tilde{U}_j(X)$ are expanded inputs vector in interval form. These $\tilde{U}_j(X)$ are considered here as Chebyshev and Legendre polynomials in interval form.

The net output is given as

$$\tilde{O}_J = f\left(\tilde{Z}_J\right)$$

Here unipolar sigmoidal function has been used as the activation function and defined as

$$f\left(\tilde{Z}_J\right) = \frac{1}{1 + e^{-\gamma\left(\tilde{Z}_J\right)}}$$

Cost function used for minimization of error is defined as

$$E = \frac{1}{2}\left[\tilde{d}_j - \tilde{O}_j\right]^2 = \frac{1}{2}\tilde{e}_j^2 \tag{9.12}$$

where $\tilde{d}_J$ is the desired output, $\tilde{O}_J$ is the target output and $\tilde{e}_J$ is the error value. The error value is computed to obtain the desired accuracy in interval form. Weights are updated as follows:

$$\tilde{w}_j(New) = \tilde{w}_j(Old) + \Delta\tilde{w}_j \tag{9.13}$$

where change in weights in interval form are calculated as

$$\Delta\tilde{w}_j = \left[-\eta \frac{\partial \tilde{E}}{\partial \tilde{w}_j}\right] = \left[-\eta \left(\tilde{d}_J - \tilde{O}_J\right)\left(1 - \tilde{O}_J{}^2\right) U_i\left(\tilde{X}\right)\right] \tag{9.14}$$

Here η is the learning parameter. We follow the same procedure to update the bias $\tilde{\theta}_j$ .The multi-input and multi-output single layer IFLNN architecture is shown in Figure 9.1, where $\left\{\tilde{U}_j\right\}_{j=1}^{N}$ are set of interval Chebyshev and Legendre orthogonal polynomials.

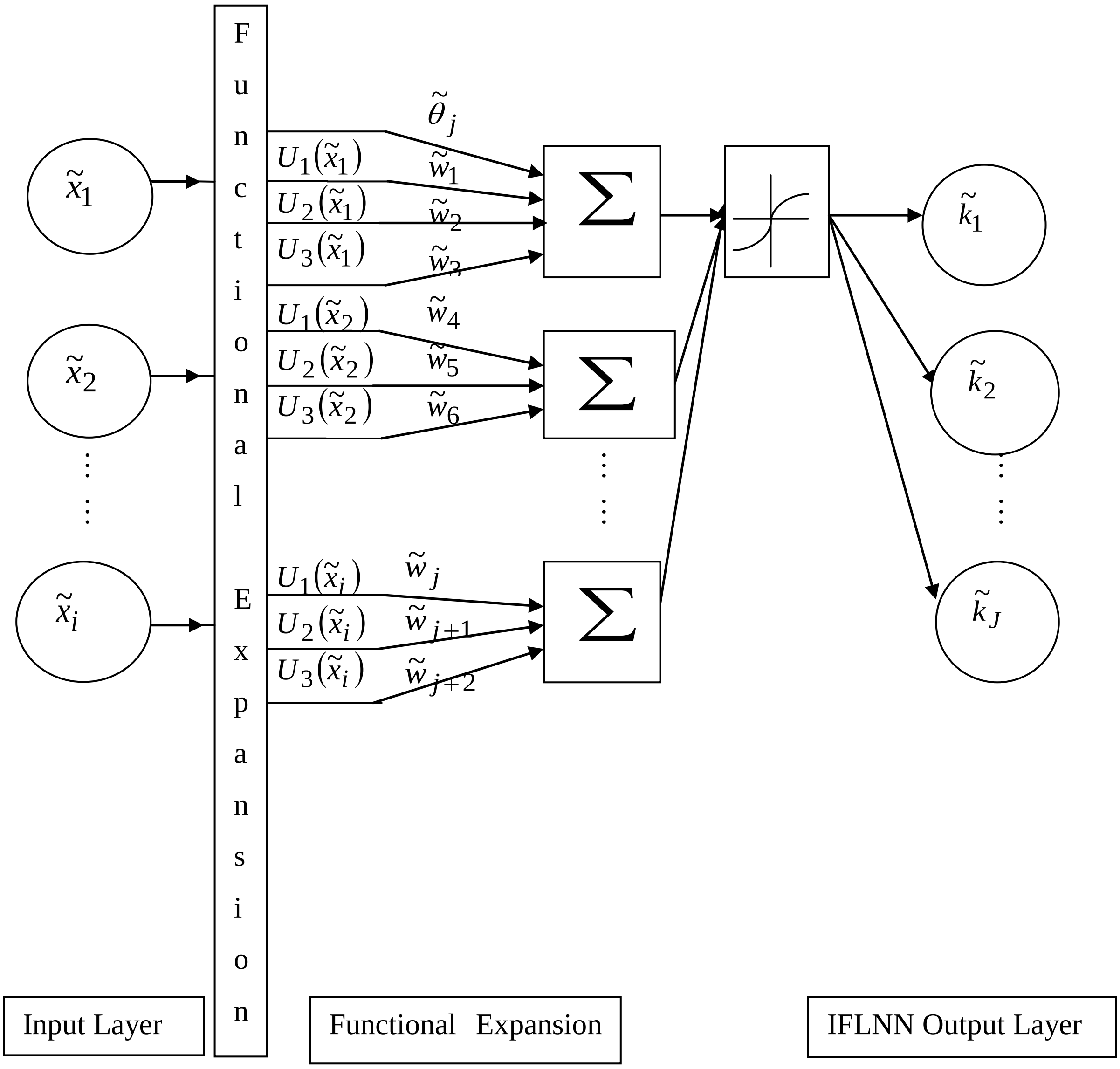

Figure 9.1: Single Layer Multi-Input and Multi-Output Interval Functional Link Neural Network

9.4 Results and Discussion

Although the developed method has been used for different storey shear structure but here only two and five storey shear structure has been reported to understand the methodology. To investigate the present method numerical experiment has been shown for two and five storeys lumped mass structure to identify interval stiffness parameters. One may note for identifying the interval stiffness parameters we need to have interval responses in the input nodes. In practical application due to error in measurements, we may have the response data in interval form. It is worth mentioning that the response may actually be obtained from some experiments. But here the analyses have been shown by numerical simulation only. In this respect one may see that the procedure is mentioned with constant masses but with interval stiffness parameters. To get the set of data of interval responses and interval stiffness parameters, the problem has to be solved first as forward vibration problem. For this the initial design (structural) parameters in interval form are randomized [74] and training sets of initial interval stiffness parameters are generated. For the above sets of initial interval stiffness parameters, the set of corresponding responses in interval form are generated from Eq. (9.6) for ambient vibration and from Eq. (9.5) for other case (after solving Eq. (9.8) for $\underline{y}$ or $\overline{\mathrm{y}}$). In order to get the interval responses for ambient vibration problem, Eq. (9.6) is used and for forced vibration problem Eq. (9.5) and Eq. (9.8) are used. The neural network training is done till a desired accuracy is reached. We will identify the stiffness parameters in interval form using the interval form of the maximum absolute response. The methodology has been discussed by giving the results for following two cases with two functional link networks.

Case (i): Ambient vibration: interval response with initial condition in interval form
Case (ii): Forced vibration: interval response with the forcing function in interval form.

9.4.1 Ambient Vibration

Example 1. Two storey shear buildings:

The input layer will have the nodes as $\left\{\tilde{X}_1=\left[\underline{X}_1\ ,\overline{X}_1\right] \text{and } \tilde{X}_2=\left[\underline{X}_2\ ,\overline{X}_2\right]\right\}$ and output layer will have the nodes as $\left\{\tilde{k}_1=\left[\underline{k}_1,\overline{k}_1\right] \text{and } \tilde{k}_2=\left[\underline{k}_2\ ,\overline{k}_2\right]\right\}$ for two storey shear structure. For case (i) one problem have been solved for two storey shear structure. Here the system is subjected to initial condition expressed by the vector (with zero displacement) in interval form as $\left\{\underline{\dot{x}(0)}\ \ \overline{\dot{x}(0)}\right\}=\left\{(8\ ,10)\ \ (-10\ ,-8)\right\}^T$. The masses are kept constant for this problem and are taken as $\underline{m}_1=\overline{m}_1=1$ and $\underline{m}_2=\overline{m}_2=1$. The initial interval stiffness parameter is considered as $\tilde{k}_1=[1000, 2000]$ and $\tilde{k}_2=[1000, 2000]$. 50 sets of data for both responses and structural parameters are generated from these initial interval stiffness parameters. The values of *m* are taken as 2 for interval Chebyshev and interval Legendre Polynomials. Different degrees of Chebyshev polynomials in interval form are considered for training in order to achieve the desired accuracy of 0.001. But, here we have given the results for two degree polynomials. After training 10 trained data among 50 is incorporated for comparison of the desired and interval ChNN values for this problem in Tables 9.1. The CPU time for ChNN is 110.34 secs. The stored converged weights of interval ChNN is used for testing interval LeNN. With the same desired values the testing has been done for interval LeNN. Comparison of results between desired and interval LeNN has been given in Tables 9.2.

Table 9.1: Comparison of desired and interval ChNN value for ambient vibration with interval initial condition for $\underline{k}_1, \overline{k}_1$ and $\underline{k}_2$, $\overline{k}_2$ for two storey building

Data No.	$\underline{k}_1$ (ChNN)	$\underline{k}_1$ (Des)	$\overline{k}_1$ (ChNN)	$\overline{k}_1$ (Des)	$\underline{k}_2$ (ChNN)	$\underline{k}_2$ (Des)	$\overline{k}_2$ (ChNN)	$\overline{k}_2$ (Des)
1	1819.938	1860.4406	1944.1804	1949.1417	1358.0107	1347.8792	1794.3657	1716.9174
2	1355.8861	1311.3061	1989.1416	1934.4051	1452.3857	1446.0266	1995.9252	2009.4916
3	1362.9696	1305.5338	1921.9842	1984.3983	1092.1555	1054.2395	1282.1899	1297.8493
4	1388.3489	1342.9363	1828.0428	1858.9388	1106.5749	1177.1075	1465.9163	1424.5225
5	1450.3335	1477.0682	1751.8282	1785.559	1457.5081	1474.8399	1629.2183	1662.8081
6	1434.6163	1513.3774	1671.9363	1658.1984	1360.7641	1330.829	1763.5848	1773.9571
7	996.2312	1035.2282	1153.025	1177.6025	1811.7288	1828.204	1866.9356	1898.4861
8	1453.1112	1398.5895	1941.8661	1852.2066	1184.0867	1110.2215	1180.4295	1118.1552
9	1119.4539	1133.9313	1513.2829	1569.0325	1100.9639	1188.117	1916.9943	1988.4179
10	1088.3682	1030.889	1886.814	1864.0999	1381.4564	1369.6349	1552.7181	1539.9821

Table 9.2: Comparison of desired and interval LeNN value for ambient vibration with interval initial condition for $\underline{k}_1, \overline{k}_1$ and $\underline{k}_2$, $\overline{k}_2$ for two storey building

Data No.	$\underline{k}_1$ (LeNN)	$\underline{k}_1$ (Des)	$\overline{k}_1$ (LeNN)	$\overline{k}_1$ (Des)	$\underline{k}_2$ (LeNN)	$\underline{k}_2$ (Des)	$\overline{k}_2$ (LeNN)	$\overline{k}_2$ (Des)
1	1817.938	1860.4406	1944.1804	1949.1417	1357.0107	1347.8792	17964.3657	1716.9174
2	1359.8861	1311.3061	1984.1416	1934.4051	1456.3857	1446.0266	1999.9252	2009.4916
3	1365.9696	1305.5338	1924.9842	1984.3983	1094.1555	1054.2395	1287.1899	1297.8493
4	1382.3489	1342.9363	1829.0428	1858.9388	1107.5749	1177.1075	1464.9163	1424.5225
5	1457.3335	1477.0682	1755.8282	1785.559	1454.5081	1474.8399	1622.2183	1662.8081
6	1433.6163	1513.3774	1678.9363	1658.1984	1360.7641	1330.829	1763.5848	1773.9571
7	995.2312	1035.2282	1157.025	1177.6025	1818.7288	1828.204	1868.9356	1898.4861
8	1458.1112	1398.5895	1841.8661	1852.2066	1180.0867	1110.2215	1188.4295	1118.1552
9	1113.4539	1133.9313	1519.2829	1569.0325	1108.9639	1188.117	1918.9943	1988.4179
10	1080.3682	1030.889	1884.814	1864.0999	1389.4564	1369.6349	1559.7181	1539.9821

9.4.2 Force Vibration

Example 1. Five storey shear buildings:

The input nodes are taken as $\{\tilde{x}_1=[\underline{x}_1, \overline{x}_1], \tilde{x}_2=[\underline{x}_2, \overline{x}_2], \tilde{x}3=[\underline{x}_3, \overline{x}_3], \tilde{x}_4=[\underline{x}_4, \overline{x}_4]$ and $\tilde{x}_5=[\underline{x}_5, \overline{x}_5]\}$

and the output nodes are taken as $\{\tilde{k}_1=[\underline{k}_1,\overline{k}_1],\ \tilde{k}_2=[\underline{k}_2\,,\overline{k}_2],\tilde{k}_3=[\underline{k}_3\,,\overline{k}_3],\tilde{k}_4=[\underline{k}_4\,,\overline{k}_4]$ and $\tilde{k}_5=[\underline{k}_5\,,\overline{k}_5]\}$ for five storey shear structure. In case (ii) the forcing function vector in interval form with zero initial condition is defined as $\tilde{F}_1(t)=\{80\sin((1.6\pi\,\mathrm{t})+\pi),100\sin((1.6\pi\,\mathrm{t})+\pi)\}$, $\tilde{F}_2(t)=\{80\sin(1.6\pi\,\mathrm{t}),100\sin(1.6\pi\,\mathrm{t})\}$, $\tilde{F}_3(t)=\{80\sin((3.2\pi\,\mathrm{t})+\pi),100\sin((3.2\pi\,\mathrm{t})+\pi)\}$, $\tilde{F}_4(t)=\{80\sin(3.2\pi\,\mathrm{t}),100\sin(3.2\pi\,\mathrm{t})\}$and $\tilde{F}_5(t)=\{80\sin((3.2\pi\,\mathrm{t})+\pi),100\sin((3.2\pi\,\mathrm{t})+\pi)\}$ The masses are kept constant for this problem and are taken as $\underline{m}_1=\overline{m}_1=\cdots\ \underline{m}_5=\overline{m}_5=1$. The initial interval stiffness parameter is considered as $\tilde{k}_1=[2200,1100],\tilde{k}_2=\cdots=\tilde{k}_5=[2100,1100]$. Here 60 data sets for both responses and structural parameters are generated from these initial interval stiffness parameters. The values of *m* are taken as 2 for interval Chebyshev and interval Legendre Polynomials. Chebyshev polynomials of fifth degree are considered to train interval ChNN so as to get an accuracy of 0.001. Comparison between the desired and ChNN values for 10 data among 60 sets of data have been plotted in Figures 9.2(a)-9.2(e). The CPU time for ChNN is 268.645721 secs. Again testing is done with LeNN from the converged weights of ChNN. Results between desired and intervals LeNN are plotted in Figures 9.3(a)-9.3(e).

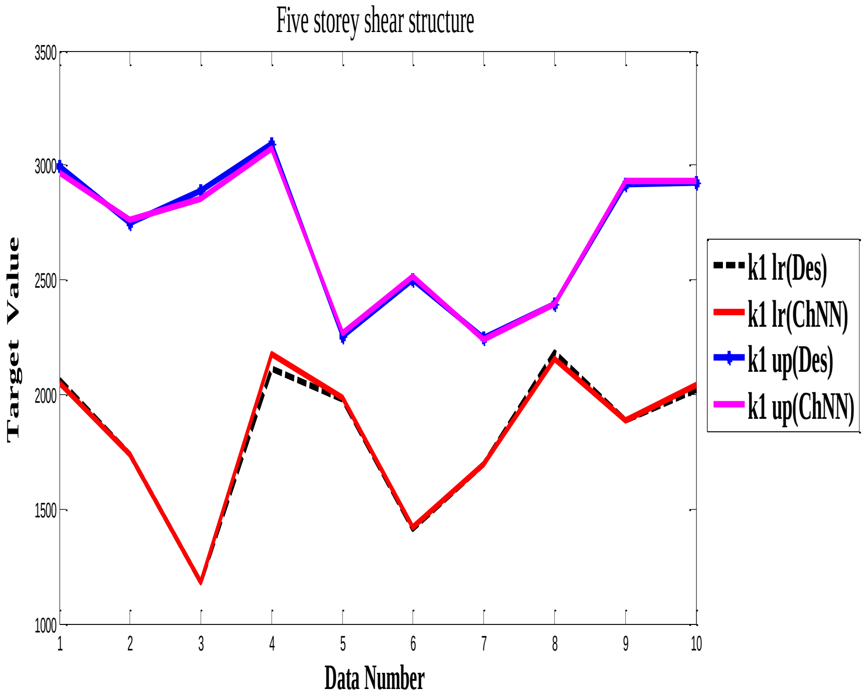

Figure 9.2(a): Comparison of desired and interval ChNN value for forced vibration with interval forcing function for $\underline{k}_1,\overline{k}_1$ for five storey shear building

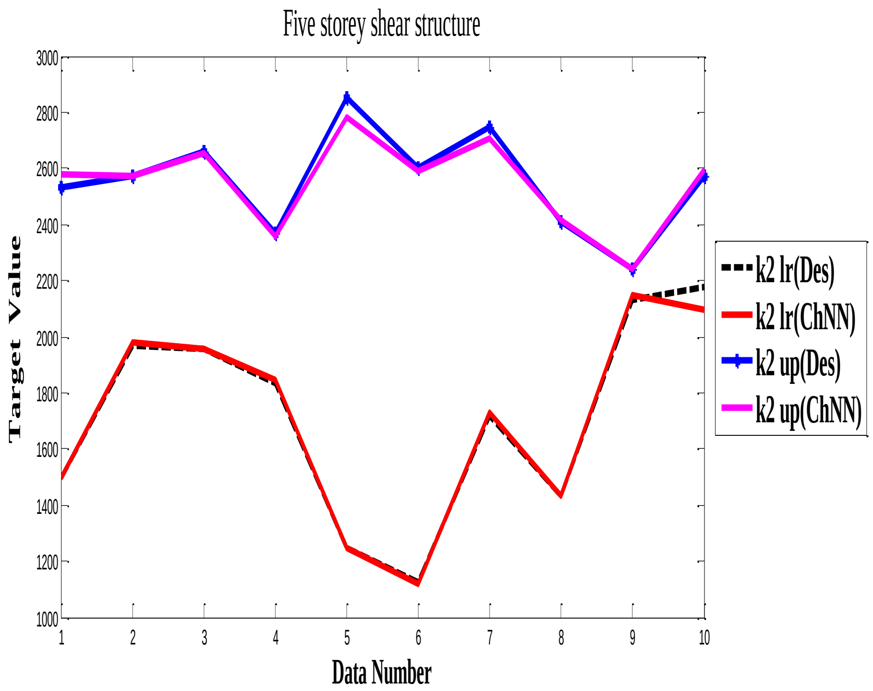

Figure 9.2(b): Comparison of desired and interval ChNN value for forced vibration with interval forcing function for $\underline{k}_2, \overline{k}_2$ for five storey shear building

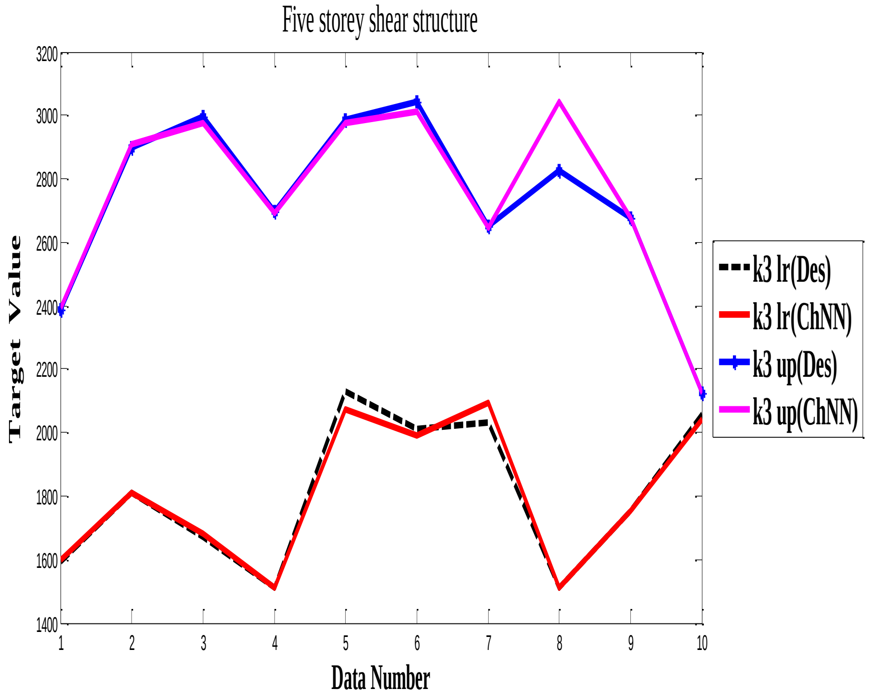

Figure 9.2(c): Comparison of desired and interval ChNN value for forced vibration with interval forcing function for $\underline{k}_3, \overline{k}_3$ for five storey shear building

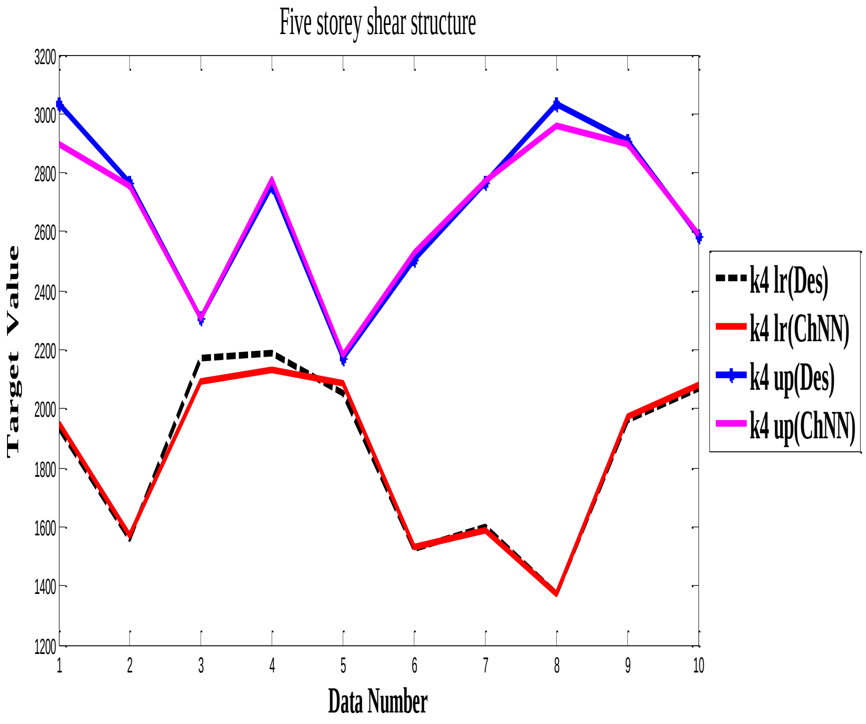

Figure 9.2(d): Comparison of desired and interval ChNN value for forced vibration with interval forcing function for $\underline{k}_4, \overline{k}_4$ for five storey shear building

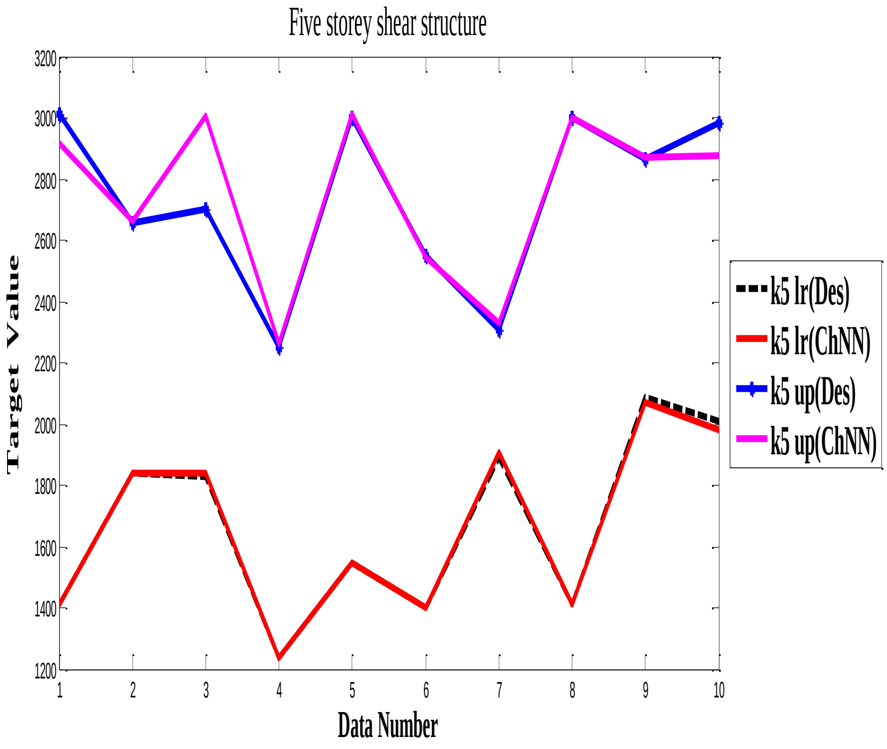

Figure 9.2(e): Comparison of desired and interval ChNN value for forced vibration with interval forcing function for $\underline{k}_5, \overline{k}_5$ for five storey shear building

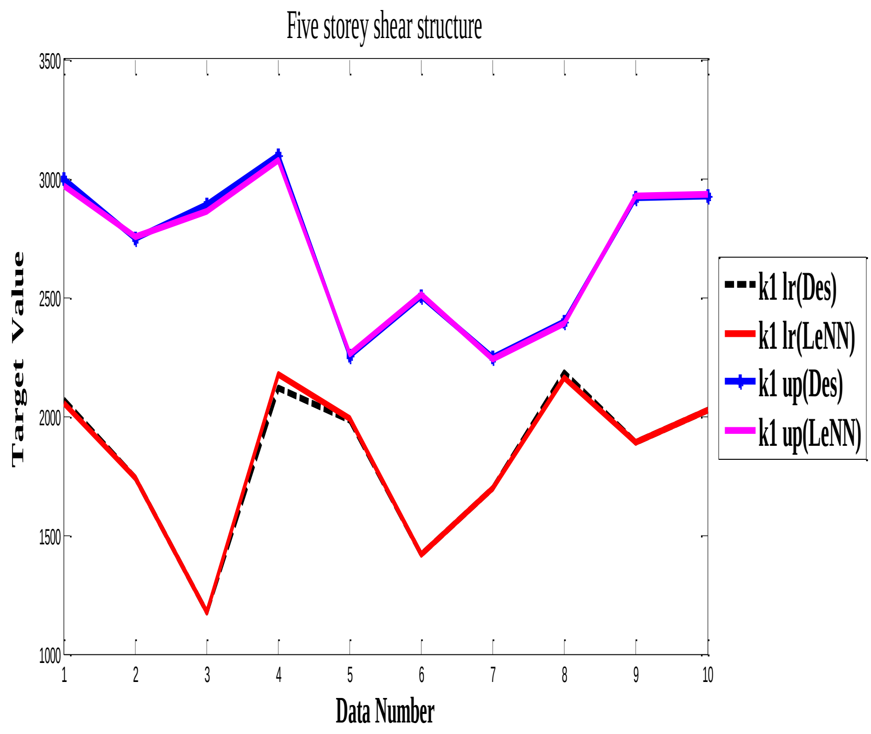

Figure 9.3(a): Comparison of desired and interval LeNN value for forced vibration with interval forcing function for $\underline{k}_1, \overline{k}_1$ for five storey shear building

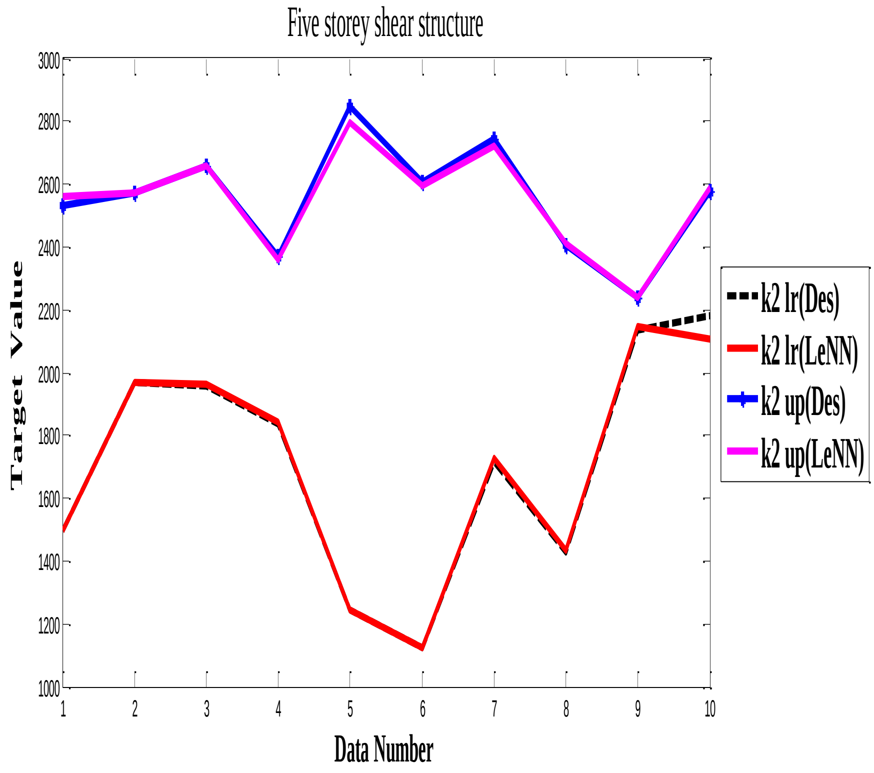

Figure 9.3(b): Comparison of desired and interval LeNN value for forced vibration with interval forcing function for $\underline{k}_2, \overline{k}_2$ for five storey shear building

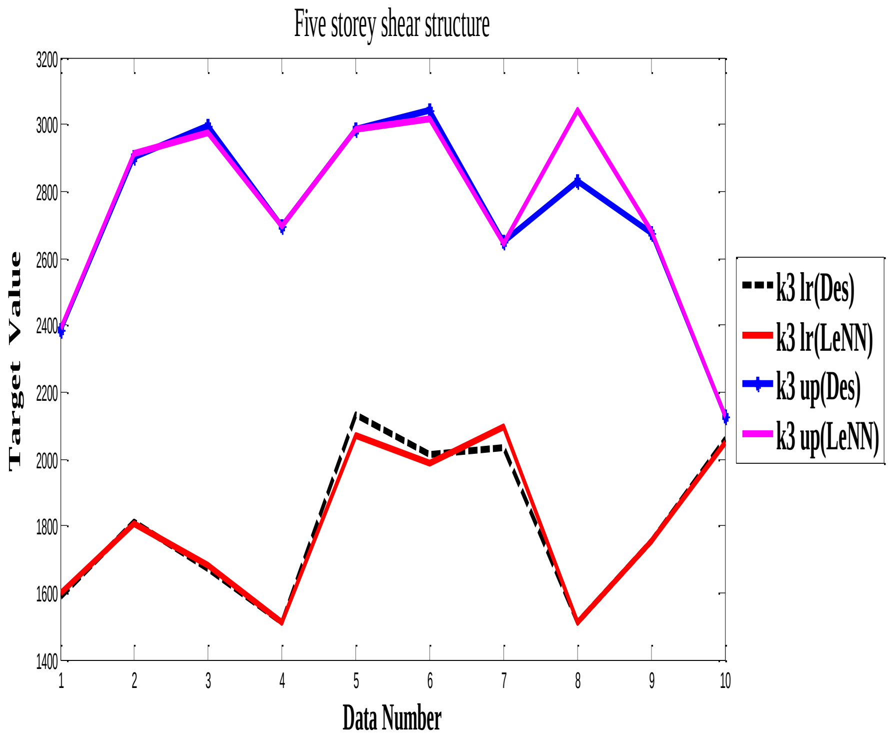

Figure 9.3(c): Comparison of desired and interval LeNN value for forced vibration with interval forcing function for $\underline{k}_3, \overline{k}_3$ for five storey shear building

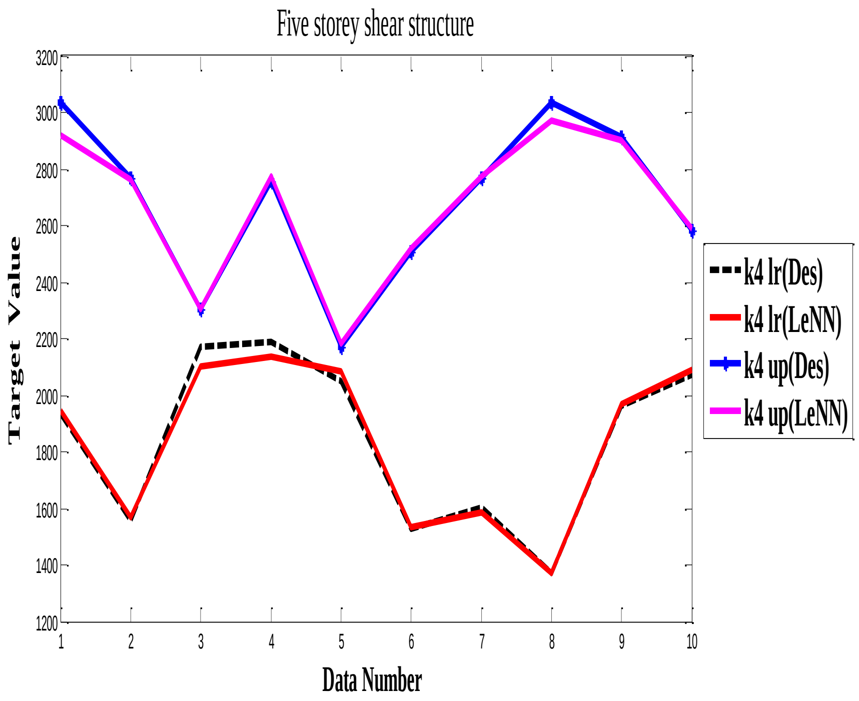

Figure 9.3(d): Comparison of desired and interval LeNN value for forced vibration with interval forcing function for $\underline{k}_4, \overline{k}_4$ for five storey shear building

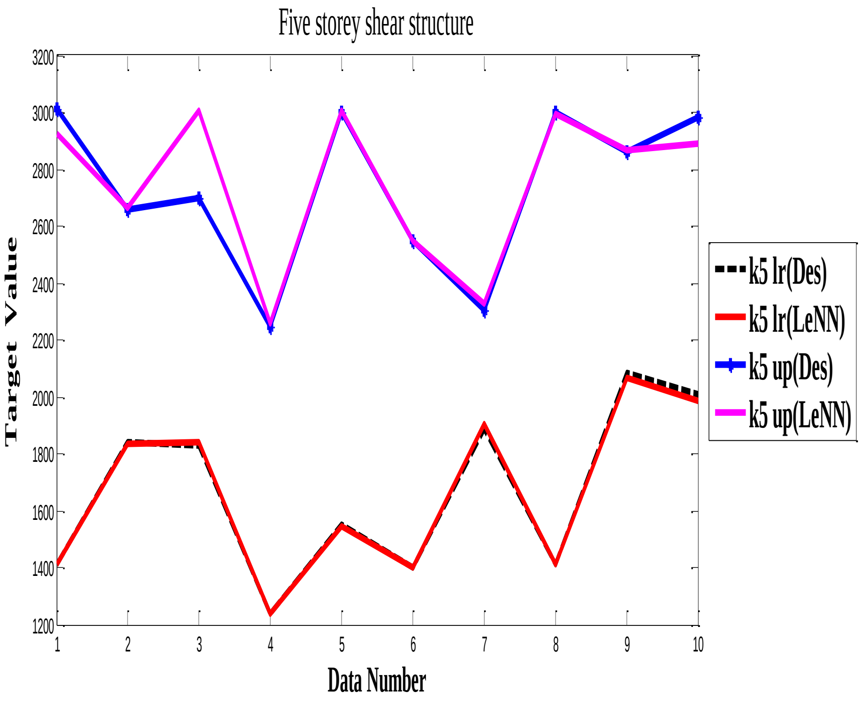

Figure 9.3(e): Comparison of desired and interval LeNN value for forced vibration with interval forcing function for $\underline{k}_5, \overline{k}_5$ for five storey shear building

9.5 Conclusion

A novel method is presented here to estimate interval stiffness parameters of multistorey shear building using single layer multi-input and multi-output Interval Functional Link neural network (IFLNN). Training with ChNN is done and the converged weights are stored. The converged weights of interval ChNN is used for testing LeNN. The models need to have the knowledge of the initial design parameters namely stiffness and mass of the said problem and using these initial design parameters, responses for a structure in interval form may be computed by numerical simulation. These interval responses are used as inputs to train the IFLNN model. The converged IFLNN model will have the capability to estimate the present interval stiffness parameter values for each floor. It may be seen that the present IFLNN model are easy to implement with low computational complexity and are also more efficient than INN. Example problems of two and five storey shear buildings have been analyzed for free and forced vibration case to show the efficacy and usefulness of the IFLNN model. Testing has been done for shear buildings which validate the novelty of the present methods.

Chapter 10

Conclusions and Future Directions

10.1 Overall Conclusion

This chapter includes overall conclusions of the present study and suggestions for future work. It is already mentioned that interval and fuzzy neural network model have been developed to solve uncertain (interval and fuzzy) coupled ordinary differential equations with respect to uncertain (interval and fuzzy) system identification problems for multistorey shear buildings. Novel techniques for estimation of structural parameters have been developed using single layer with multi-input and multi-output functional link neural network viz. ChNN, LeNN, HeNN. INN, FNN and FLNN have also been used for response prediction of multistorey shear buildings subject to earthquake ground motion. Finally, single layer functional link neural network has been extended to handle uncertain (interval) structural problem for identification of structural parameters.

In the following paragraphs conclusions are drawn with respect to the various proposed methods and the application problems mentioned in the previous chapters.

❖ Procedures have been developed to identify uncertain (interval and fuzzy) structural parameters (stiffness etc.) from frequency data for multistorey shear buildings using interval and fuzzy neural network models (Chapter 3). Here, initial design parameters namely stiffness, mass and frequency parameters of the said problem are known. It is assumed that after certain period of time only the stiffness is changed and not the mass i.e. mass remains the same. As such, equipments are available to get the present values of the frequencies and using these one may get the present parameter values by the developed mathematical model such as INN and FNN. If sensors are placed to capture the frequency of the floors in interval and fuzzified (uncertain) form then those may be fed into the proposed new INN and FNN models to get the present stiffness parameters. Although to train the new INN and FNN models, set of data are generated here by solving the governing coupled (uncertain) ordinary differential equations numerically beforehand. As such converged INN and FNN models give the present stiffness parameter values in interval/fuzzified form for each floor. Thus one may predict the health of the uncertain structure. These methods are applied to solve different multi-storey shear buildings to show the efficacy of the proposed methods.

- Next a forward problem for each time step is solved in (Chapter 4) for a given input to the system, rather than solving the inverse vibration problem. Thus, the solution vector is generated. Again, the initial design parameters viz. stiffness, mass and so the responses of the said problem are known in uncertain form. The initial values of the uncertain physical parameters of the system are used to obtain the interval and fuzzy responses. The uncertain (interval and fuzzy) responses and the corresponding stiffness parameters are used as the input/output in the neural net. Next, the interval neural network and fuzzy neural network model are trained by the proposed interval and fuzzy error back propagation training algorithm. After training of the model, physical parameters may be identified if new maximum response data is supplied as input to the net which are also in interval and fuzzified form. The procedures have been demonstrated for multi-storey shear buildings and the structural parameters are identified using the response of the structure subject to initial condition and horizontal displacement which are also in interval and fuzzified form. Hence the present method will also be helpful for predicting the health of the uncertain structure.

- Methods have been proposed in (Chapter 5) by creating a trained black box in terms of INN and FNN containing the characteristics of the multi-storey structure and of the earthquake motion. INN and FNN model have been used to compute the structural response of multi-storey shear structures subject to ground motion data (interval and fuzzy) of Indian earthquakes occurred at Chamoli and Uttarkashi. The interval and fuzzy neural network are first trained for a real earthquake data viz. the ground acceleration as input and the numerically generated responses of different floors of multi-storey buildings as output. It may be noted that till date no model exists to handle positive and negative data in the INN and FNN. As such, here the bipolar data in [-1, 1] are converted first to unipolar form that is to [0, 1] by means of a novel transformation for the first time to handle the above training patterns in normalized form. Once the training is done, again the unipolar data are converted back to its bipolar form by using the inverse transformation. The trained INN and FNN architectures are then used to simulate and test the structural response of different floors for various intensity earthquake data. As mentioned that although the simulation is done with numerically generated response data for particular earthquake (experimental) data but the idea may also be used for actual experimental data of the building response. So, by using the input and output as the ground motion and the floor response, one can train the models. Accordingly then the storey response may be predicted for future earthquakes using the trained model. It is found that the predicted responses given by INN and FNN models are good for practical purposes.

❖ Powerful soft computing techniques viz. single layer interval and fuzzy neural network models have been used for identification of uncertain structural parameters (Chapter 6). If the available information and/or data are uncertain or non-probabilistic in nature, then mathematical model needs to be developed accordingly. As such, INN and FNN have been developed which can handle uncertain (interval and fuzzified) data. Direct method for system identification of uncertain multi-storey shear structures from their dynamic responses have been proposed in interval and fuzzified form. To identify the physical parameters in interval and fuzzified form, the governing equations of motion are used systematically in a series (cluster) of INNs and FNNs. The equations of motion is first solved by the classical method to get responses of the consecutive stories and then the equations of motion are modified based on relative responses of consecutive stories in such a way that the new set of equations can be implemented in a cluster of INNs and FNNs. Here, single-layer INNs and FNNs have been used for training each cluster of the INN and FNN such that the converged weights give the uncertain structural parameters. Various example problems have been solved and related results are reported to show the reliability and powerfulness of the method. It is worth mentioning that the cluster of FNN may directly estimate the structural parameters in fuzzified form. The fuzzy estimates are certainly useful for design engineers by knowing the lower, centre and upper bounds of the structural parameters.

❖ Solving system identification problems by classical or traditional neural network are sometimes time consuming. Hence a novel method has been presented in (Chapter 7) which can handle system identification problems with less computational time limit. A single layer functional link neural network with multi-input and multi-output with feed forward neural network model and principle of error back propagation has been used to identify structural parameters. In the Functional Link neural network model, the hidden layer is excluded by enhancing the input patterns with the help of orthogonal polynomials such as Chebyshev, Legendre and Hermite. Training with multilayer neural network (MNN) is also done and it is noted that ChNN, LeNN and HNN takes less computation time and gives good result as compared to MNN. The models need to have the knowledge of the initial design parameters namely stiffness and mass of the said problem and using these initial design parameters, frequencies for a structure may be computed by numerical simulation. These frequencies are used as inputs to train the FLNN model. Present structural parameter values of the shear structure may be identified by using the proposed FLNN. The converged FLNN model will have the capability to estimate the present stiffness parameter values for each floor. It may be seen that the present FLNN model are easy to implement with low computational complexity and are also more efficient than MNN. Example problems of three, six and eight storey shear buildings have been analyzed to show the efficacy and usefulness of the present FLNN model. Testing is also been done for three and six storey shear buildings with the stored converged weights of FLNN which validate the novelty of the present methods.

- Further, we have proposed Functional Link Neural Network (FLNN) for structural response prediction of tall buildings due to seismic loads (Chapter 8). The ground acceleration data has been taken as input and structural responses of different floors of multi-storey shear buildings have been considered as output. Here, functional expansion block in FLNN has been used along with efficient Chebyshev and Legendre polynomials. ChNN is trained with random weights for different order of polynomials and it is seen that for higher order of polynomial we get better accuracy. LeNN is trained for different order of polynomials by two ways, one way is by using the stored converged weights of ChNN and the other way is by considering the random weights. After the training is complete it is found that LeNN with stored converged weights gives better accuracy than random weights. Training is done with Chamoli earthquake data but for testing different intensities of Uttarkashi earthquake data have been used. With the stored weights (of training), testing is done to show the powerfulness of the present model. In testing it is found that FLNN can predict structural response for different storey subject to different earthquake loads. It is worth mentioning that FLNN gives better accuracy in all the cases and also found to be computationally more efficient than MNN as it takes less computation time.

- Finally, a new FLNN viz. interval functional link neural network (IFLNN) has been developed and then uncertain structural parameters of multistorey shear buildings have been identified in (Chapter 9). The parameters are identified here using response of the structure for both ambient and forced vibration. In the functional link, the Chebyshev and Legendre orthogonal polynomials are taken as intervals.

Although exhaustive investigations have been done by developing different ANN models for structural system identification problems, but we do not claim that the proposed methods are most efficient. As such, there may be some gaps and few limitations on the proposed models and we may identify various scopes of improvement. Accordingly, these limitations and scopes may open a new vista for future research and are discussed in the following section.

10.2 Scope for Further Research

- System identification methods to handle higher order nonlinear coupled differential equations using INN and FNN models.

- New Single layer FLNN using other orthogonal polynomials to solve system identification problems.

- Single layer FLNN model has been developed for linear structures which may be extended to nonlinear cases.

- Single layer FLNN can be extended to fuzzy functional link neural network for solving uncertain system identification problems.

- Estimation of uncertain structural parameters by cluster/series of interval and fuzzy neural network for SI problems with nonlinear coupled differential equations.

- Development of other new techniques to predict structural response subject to seismic loads.

- One dimensional shear structure is considered here. The analysis may be extended for two or three dimensional problems but then the modelling to be done accordingly for the INN, FNN and FLNN.

- INN and FNN may also be refined with uncertain weights etc.

- Other structural problems may also be solved to validate the models.

- Experimental data may be used (if available) for the system identification problems.

References

[1] Bekey, G. A., 1970. “System identification–an introduction and a survey”. *Simulation,* **15**(4), October, pp. 151-166.

[2] Hart, G., and Yao, J., 1977. “System identification in structural dynamics”. *Journal of Engineering Mechanics Division,* **103**(6), pp. 1089-1104.

[3] Ibanez, P., 1979. “Review of analytical and experimental techniques for improving structural dynamics models”. *Welding Research Council Bulletin*, Bulletin no. **249**, pp. 1–193.

[4] Sinha, N. K., and Kuszta, B., 1983. *Modelling and Identification of Dynamic Systems*. Van Nostrand Reinhold Co., New York.

[5] Datta, A. K., Shrikhande, M., and Paul, D. K., 1998. “System identification of buildings: a review”. Proceedings of eleventh symposium on earthquake engineering. Roorkee, India: University of Roorkee.

[6] Kerschen, G., Worden, K., Vakakis, A. F., and Golinval, J. C., 2007. “Nonlinear system identification in structural dynamics: current status and future directions”. 25th International Modal Analysis Conference, Orlando, Floride, U.S.A. pp. 1-26.

[7] Fan, W., and Qiao, P., 2011. “Vibration-based damage identification methods: a review and comparative study”. *Structural Health Monitoring*, **10**(1), pp. 83-111.

[8] Sirca Jr, G. F., and Adeli, H., 2012. “System Identification in structural engineering”. *Scientia Iranica*, **19**(6), December, pp. 1355-1364.

[9] Masri, S. F., Bekey, G. A., Sassi, H., and Caughey, T. K. 1982. “Non Parametric identification of a class of nonlinear multidegree dynamic systems.” *Earthquake Engineering and Structural Dynamics*, **10**(1), January, pp. 1–30.

[10] Natke, H. G., 1982. *Identification of Vibrating Structures*, Springer, Berlin.

[11] Zhao, Q., Sawada, T., Hirao, K., and Nariyuki, Y., 1995. “Localized identification of MDOF structures in the frequency domain”. *Earthquake Engineering and Structural Dynamics*, **24**(3), March, pp. 325-338.

[12] Loh, C. H., and Tou, I. C., 1995. “A system identification approach to the detection of changes in both linear and non-linear structural parameters”. *Earthquake Engineering and Structural Dynamics*, **24**(1), January, pp. 85–97.

[13] Yuan, P., Wu, Z., and Ma, X., 1998. “Estimated mass and stiffness matrices of shear building from modal test data”. *Earthquake Engineering and Structural Dynamics*, **27**(5), May, pp. 415–421.

[14] Udwadia, F. E., and Proskurowski, W., 1998. “A memory matrix-based identification methodology for structural and mechanical systems”. *Earthquake Engineering and Structural Dynamics*, **27**(12), December, pp. 1465-1481.

[15] Sanayei, M., McClain, J. A. S., Wadia-Fascetti, S., and Santini, E. M., 1999. “Parameter estimation incorporating modal data and boundary conditions”. *Journal of Structural Engineering*, **125**(9), September, pp. 1048-1055.

[16] Quek, S. T., Wang, W., and Koh, C. G., 1999. “System identification of linear MDOF structures under ambient excitation”. *Earthquake Engineering and Structural Dynamics*, **28**(1), January, pp. 61–77.

[17] Huang, C. S., 2001. “Structural identification from ambient vibration measurement using the multivariate AR model”. *Journal of Sound and Vibration*, **241**(3), March, pp. 337–359.

[18] Koh, C. G., Chen, Y. F., and Liaw, C. Y., 2003. “A hybrid computational strategy for identification of structural parameters”. *Computers & Structures*, **81**(2), February, pp. 107–117.

[19] Takewaki, I., and Nakamura, M., 2005. “Stiffness-damping simultaneous identification under limited observation”. *Journal of Engineering Mechanics*, **131**(10), October, pp. 1027-1035.

[20] Lei, Y., Su, Y., and Shen, W., 2013. “A probabilistic damage identification approach for structures under unknown excitation and with measurement uncertainties”. *Journal of Applied Mathematics*, **2013**, May, pp. 1-7.

[21] Friswell, M. I., Inman, D. J., and Pilkey, D. F., 1998. “The direct updating of damping and stiffness matrices”. *AIAA Journal*, **36**(3), pp. 491–493.

[22] Fassois, S. D., and Sakellariou, J. S., 2007. “Time-series methods for fault detection and identification in vibrating structures”. *Philosophical Transactions of the Royal Society A*, **365**(1851), December, pp. 411–448.

[23] Alvin, K. F., Robertson, A.N., Reich, G.W., and Park, K.C., 2003. “Structural system identification: from reality to models”. *Computers & Structures*, **81**(12), May, pp. 1149–1176.

[24] Yu, E., Taciroglu, E., and Wallace, J. W., 2007. “Parameter identification of framed structures using an improved finite element model-updating method-Part I: Formulation and verification”. *Earthquake Engineering and Structural Dynamics*, **36**(5), pp. 619-639.

[25] Bhat, R. B., 1997. “Nature of stationarity of the natural frequencies at the natural modes in the Rayleigh–Ritz method”. *Journal of Sound and Vibration*, **203**(2), June, pp. 251-263.

[26] Lu, Y., and Tu, Z., 2004. “A two-level neural network approach for dynamic FE model updating including damping”. *Journal of Sound and Vibration*, **275**(3–5), August, pp. 931–952.

[27] Perry, M. J., Koh, C. G., and Choo, Y. S., 2006. “Modified genetic algorithm strategy for structural identification”. *Computers and Structures*, **84**(8-9), March, pp. 529–540.

[28] Brownjohn, J. M. W., 2003. “Ambient vibration studies for system identification of tall buildings”. *Earthquake Engineering and Structural Dynamics*, **32**(1), January, pp. 71–95.

[29] Mahmoudabadi, M., Ghafory-Ashtiany, M., and Hosseini, M., 2007. “Identification of modal parameters of non-classically damped linear structures under multicomponent earthquake loading”. *Earthquake Engineering and Structural Dynamics*, **36**(6), pp. 765–782.

[30] Tang, H., Xue, S., and Fan, C., 2008. “Differential evolution strategy for structural system identification”. *Computers and Structures*, **86**(21-22), November, pp. 2004–2012.

[31] Nandakumar, P., and Shankar, K., 2013. “Identification of structural parameters using consistent mass transfer matrix”. *Inverse Problems in Science and Engineering*, **22**(3), April, pp.436–457.

[32] Billmaier, M., and Bucher, C., 2013. “System identification based on selective sensitivity analysis: A case-study”. *Journal of Sound and Vibration*, **332**(11), May, pp. 2627–2642.

[33] Muthukumaran, P., Bhat, R. B., and Stiharu, I., 1998. “Localization of structural vibrations and acoustic radiation through boundary conditioning”. *Transactions-Canadian Society for Mechanical Engineering*, **22**(4B), pp. 519-532.

[34] Lagaros, N. D., Papadrakakis, M., and Kokossalakis, G., 2002. "Structural optimization using evolutionary algorithms". *Computers and Structures*, **80**(7-8), March, pp. 571-589.

[35] Yang, J. N., Lei, Y., Pan, S. and Huang, N., 2003. "System identification of linear structures based on Hilbert-Huang spectral analysis, Part 1: Normal modes". *Earthquake Engineering and Structural Dynamics*, **32**(9), July, pp. 1443–1467.

[36] Chakraverty, S., 2004. "Modelling for identification of stiffness parameters of multistorey structure from dynamic data." *Journal of Scientific and Industrial Research* **63**(2), February, pp. 142–148.

[37] Nicoud, Y. R., Raphael, B., and Smith, I. F., 2005. "System identification through model composition and stochastic search". *Journal of Computing in Civil Engineering*, **19**(3), July, pp. 239-247.

[38] Chakraverty, S., 2005. "Identification of structural parameters of multistorey shear buildings from modal data". *Earthquake Engineering and Structural Dynamics*, **34**(6), May, pp. 543–554.

[39] Yoshitomi, S., and Takewaki, I., 2009. "Noise-bias compensation in physical-parameter system identification under microtremor input". *Engineering Structures*, **31**(2), February, pp. 580–590.

[40] Lei, Y., and Jiang, Y. Q. 2011. "A two-stage Kalman estimation approach for the identification of nonlinear structural parameters". *Procedia Engineering*, In the Proceedings of the Twelfth East Asia-Pacific Conference on Structural Engineering and Construction — EASEC12 **14**, pp. 3088-3094.

[41] Dinh, H. M., Nagayama, T., and Fujino, Y., 2012. "Structural parameter identification by use of additional known masses and its experimental application". *Structural Control and Health Monitoring*, **19**(3), April, pp. 436-450.

[42] Beskhyroun, S., Wotherspoon, L., and Ma, Q. T., 2013. "System identification of a 13-story reinforced concrete building through ambient and forced vibration". In the proceedings of 4th ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering, Greece, June, pp. 1–11.

[43] Jianping, H., Peijuan, Z., and Hongtao, W., 2014. "Structural modal parameter identification and damage diagnosis based on Hilbert-Huang transform". *Earthquake Engineering and Engineering Vibration*, **13**(1), March, pp. 101-111.

[44] Wang, F., Ling, X., Xu, X., and Zhang, F., 2014. “Structural stiffness identification based on the extended Kalman filter research”. *Abstract and Applied Analysis,* **2014**, pp. 1-8

[45] Cho, S., Park, J. W., and Sim, S. H., 2015. “Decentralized system identification using stochastic subspace identification for wireless sensor networks”. *Sensors,* **15**(4), April, pp. 8131-8145.

[46] Angeles, J. M., and Alvarez-Icaza, L., 2010. “Identification of seismically excited buildings with two orthogonal horizontal components”. *Journal of Vibration and Control,* **17**(6), October, pp. 881-901.

[47] Niu, L., 2012. “Monitoring of a frame structure model for damage identification using artificial neural networks”. Proceedings of 2nd International Conference on Electronic & Mechanical Engineering and Information Technology (EMEIT-2012), pp. 0438–0441.

[48] Masri, S. F., Smyth, A., Chassiakos, A., Caughey, T. K., and Hunter, N., 2000. “Application of neural networks for detection of changes in nonlinear system”. *Journal of Engineering Mechanics,* **126**(7), July, pp. 666–676.

[49] Kao, C.Y., and Hung, S. L., 2003. “Detection of structural damage via free vibration responses generated by approximating artificial neural networks”. *Computers & Structures,* **81**(28–29), November, pp. 2631–2644.

[50] Furukawa, T., Ito, M., and Inoue, Y., 2005. “System identification of the base isolated structure by prediction error method using recorded seismic response data under Hyogoken-Nanbu earthquake”. In the proceedings of 12th World Conference on Earthquake Engineering, 30 January-4 February, Auckland, New Zealand, **2**, pp. 1-7.

[51] Lin, C. C., Hong, L. L., Ueng, J. M., Wu, K. C., and Wang, C. E., 2005. “Parametric identification of asymmetric buildings from earthquake response records”. *Smart Materials and Structures,* **14**(4), July, pp. 850-861.

[52] Pillai, P., and Krishnapillai, S., 2007. “A hybrid neural network strategy for identification of structural parameters”. *Structure and Infrastructure Engineering,* **6**(3), April, pp. 379–391.

[53] Wang, X. J. and Cui, J., 2008. “A two-step method for structural parameter identification with unknown ground motion”. In the proceedings of 14th World Conference on Earthquake Engineering, 12-17, October, Beijing, China, pp. 1-8.

[54] Hegde, G., and Sinha, R., 2008. “Parameter identification of torsionally coupled shear buildings from earthquake response records”. *Earthquake Engineering and Structural Dynamics,* **37**(11), September, pp. 1313–1331

[55] Hong, A. L., Betti, R., and Lin, C. C., 2009. “Identification of dynamic models of a building structure using multiple earthquake records”. *Structural Control and Health Monitoring,* **16**(2), March, pp. 178–199.

[56] Zhang, K., Li, H., Duan, Z., and Law, S. S., 2011. “A probabilistic damage identification approach for structures with uncertainties under unknown input”. *Mechanical Systems and Signal Processing,* **25**(4), May, pp. 1126–1145.

[57] Xu, B., Song, G., and Masri, S.F., 2012. “Damage detection for a frame structure model using vibration displacement measurement”. *Structural Health Monitoring,* **11**(3), January, pp. 281–292.

[58] Xu, C., Chase, J. G., and Rodgers, G. W., 2014. “Physical parameter identification of nonlinear base isolated buildings using seismic response data”. *Computers & Structures,* **145**, December, pp. 47-57.

[59] Ebrahimian, M., and Todorovska, M. I., 2015. “Structural system identification of buildings by a wave method based on a Nonuniform Timoshenko beam model”. *Journal of Engineering Mechanics,* **141**(8), August, pp. 04015022-1-04015022-11.

[60] Zhou, C., Chase, J. G., Rodgers, G. W., and Tomlinson, H., 2015. “Physical parameter identification of structural systems with hysteretic pinching”. *Computer-Aided Civil and Infrastructure Engineering,* **30**(4), April, pp. 247-262.

[61] Guo, Y., Kwon, D. K., and Kareem, A., 2016. “Near-real-time hybrid system identification framework for civil structures with application to Burj Khalifa”. *Journal of Structural Engineering,* **142**(2), February, pp. 04015132-1-04015132-15.

[62] Derras, B., and Bekkouche, A., 2011. “Use of the artificial neural network for peak ground acceleration estimation”. *Lebanese Science Journal,* **12**(2), pp. 101-114.

[63] Lagaros, N. D., and Papadrakakis, M., 2012. “Neural network based prediction schemes of the non-linear seismic response of 3D buildings”. *Advances in Engineering Software,* **44**(1), February, pp. 92-115.

[64] Zamani, A. S., Al-Arifi, N. S., and Khan, S., 2012. “Response prediction of earthquake motion using artificial neural networks”. *IJAR-CSIT Research Paper,* **1**(2), pp. 50–57.

[65] Robles, G. C. M. A., and Hernandez-Bercerril, R. A., 2012. "Seismic alert system based on artificial neural networks". *International Journal of Environmental, Chemical, Ecological, Geological and Geophysical Engineering*, **6**(6), pp. 342–347.

[66] Hong, H., Liu, T., and Lee, C. S., 2012. "Observations on the application of artificial neural network to predicting ground motion measures". *Earthquake Science,* **25**(2), April, pp. 161–175

[67] Ramhormozian, S., Omenzetter, P., and Orense, R. 2013. "Artificial neural networks approach to predict principal ground motion parameters for quick post-earthquake damage assessment of 526 bridges". In: NZSEE Conference, pp. 1–9.

[68] Kosmatopoulos, E. B., Smyth, A. W., Masri, S. F., and Chassiakos, A. G., 2000. "Adaptive neural identification of nonlinear structural system". ASCE Engineering Mechanics Conference, Austin, TX, May 2000.

[69] Lagaros, N. D., and Papadrakakis, M., 2004. "Learning improvement of neural networks used in structural optimization". *Advances in Engineering Software,* **35**(1), January, pp. 9-25

[70] Yun, C. B., and Bahng, E. Y., 2000. "Substructural identification using neural networks". *Computers & Structures*, **77**(1), June, pp. 41–52.

[71] Wu, Z. S., Xu, B., and Yokoyama, K., 2002. "Decentralized parametric damage based on neural networks". *Computers Aided Civil and Infrastructure Engineering*, **17**(3), May, pp. 175–184.

[72] Xu, B., Wu, Z., Chen, G., and Yokoyama, K., 2004. "Direct identification of structural parameters from dynamic responses with neural networks". *Engineering Applications of Artificial Intelligence*, **17**(8), December, pp. 931–943.

[73] Chen, C. H., 2005. "Structural identification from field measurement data using a neural network". *Smart Materials and Structures*, **14**(3), May, pp. S104–S115.

[74] Chakraverty, S., 2007. "Identification of structural parameters of two-storey shear buildings by the iterative training of neural networks". *Journal Architectural Science Review,* **50**(4), July, pp. 380–384.

[75] Pizano, D. G., 2011. "Comparison of frequency response and neural network techniques for system identification of an actively controlled structure". *Dyna,* **78**(170), April, pp. 79–89.

[76] Khanmirza, E., Khaji, N., and Majd, V. J., 2011. “Model updating of multistorey shear buildings for simultaneous identification of mass, stiffness and damping matrices using two different soft computing methods”. *Expert Systems with Applications*, **38**(5), May, pp. 5320–5329.

[77] Facchini, L., Betti, M., and Biagini, P., 2014. “Neural network based modal identification of structural systems through output-only measurement”. *Computers & Structures*, **138**(1), July, pp. 183-194.

[78] Khanmirza, E., Khaji, N., and Khanmirza, E., 2014. “Identification of linear and non-linear physical parameters of multistorey shear buildings using artificial neural network”. *Inverse Problems in Science and Engineering*, **23**(4), July, pp. 670-687.

[79] Wu, X., Ghaboussi, J., and Garett, J. H., 1992. “Use of neural networks in detection of structural damage”. *Computer & Structures*. **42**(4), February, pp. 649-659.

[80] Conte, J. P., and Durrani, A. J., 1994. “Seismic response modeling of multi-story buildings using neural networks”. *Journal of Intelligent material Systems and Structures*, **5**(3), May, pp. 392-402.

[81] Pandey, P. C., and Barai, S. V., 1995. “Multilayer perceptron in damage detection of bridge structures”. *Computers & Structures.* **54**(4), February, pp. 597-608.

[82] Zhao, J., Ivan, J. N., and DeWolf, J. T., 1998. “Structural damage detection using artificial neural networks”. *Journal of Infrastructure System*, **4**(3), September, pp. 93-101.

[83] Xu, B., Wu, Z. S., and Yokoyama, K., 2002. “A localized identification method with neural networks and its application to structural health monitoring”. *Journal of Structural Engineering*, 48A, pp. 419–427.

[84] Huang, C. S., Hung, S. L., Wen, C. M., and Tu, T. T., 2003. “A neural network approach for structural identification and diagnosis of a building from seismic response data”. *Journal of Earthquake Engineering and Structural Dynamics*, **32**(2), February, pp. 187–206.

[85] Mathur, V. K., Chakraverty, S., and Gupta, P., 2004. “Response prediction of typical rural house subject to earthquake motions using artificial neural networks, *Journal of Indian Building Congress*, **11**, pp. 99-105.

[86] Chakraverty, S., Marwala, T., and Gupta, P. 2006. "Response prediction of structural system subject to earthquake motions using artificial neural network". *Asian Journal of Civil Engineering*, **7**(3), pp. 301-308.

[87] Bakhary, N., Hao, H., and Deeks, A. J., 2009. "Structure damage detection using neural with multi-stage substructuring". *Advances in Structural Engineering*, **13**(1), pp. 1–16.

[88] Zhang, S., Wang, H., and Wang, W., 2010. "Damage detection in structures using artificial neural networks". International Conference on Artificial Intelligence and Computational Intelligence, Sanya, October, **1**, pp. 207–210.

[89] Chakraverty, S., Gupta, P., and Sharma, S., 2010. "Neural network-based simulation for response identification of two-storey shear building subject to earthquake motion". *Neural Computing and Applications*, **19**(3), April, pp. 367-75.

[90] Oliva, C. S. H., and Pichardo, F. A. N., 2010. "Seismic hazard assessment: An artificial neural network estimation". American Geophysical Union, Fall Meeting.

[91] Kerh, T., Huang, C., and Gunaratnam, D., 2011. "Neural network approach for analyzing seismic data to identify potentially hazardous bridges". Mathematical Problems in Engineering, **2011**, pp. 1–15.

[92] Xie, J., Qiu, J. F., Li, W., and Wang, J. W., 2011. "The Application of Neural Network Model in Earthquake Prediction in East China". *Chapter in Advances in Computer Science, Intelligent System and Environment*. **106**, pp. 79-84.

[93] Shi, A., and Yu, X. H., 2012. "Structural damage detection using artificial neural networks and wavelet transform". IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proceedings, pp. 7–11.

[94] Aghamohammadi, H., Mesgari, M. S., Mansourian, A., and Molaei, D., 2013. "Seismic human loss estimation for an earthquake disaster using neural network". *International Journal of Environmental Science and Technology*, **10**(5), September, pp. 931–939.

[95] Suryanita, R., and Adnan, A., 2013. "Application of neural networks in bridge health prediction based on acceleration and displacement data domain". In Proceedings of the international multi conference of engineers and computer scientists, Kowloon, Hong Kong, 13–15 March, **1**, pp. 42–47.

[96] Reyes, J., Esteban, A. M., and Alvarez, F. M., 2013. "Neural networks to predict earthquakes in Chile". *Applied Soft Computing*, **13**(2), February, pp. 1314-1328.

[97] Niksarlioglu, S., and Kulahci, F., 2013. "An artificial neural network model for earthquake prediction and relations between environmental parameters and earthquakes". *International Journal of Environmental, Chemical, Ecological, Geological and Geophysical Engineering*, **7**(2), pp. 87-90.

[98] Alvarez, F. M., Reyes, J., Esteban, A. M., and Escudero, C. R., 2013. "Determining the best set of seismicity indicators to predict earthquakes two case studies: Chile and the Iberian Peninsula". *Knowledge-Based Systems*. **50**, September, pp. 198-210.

[99] Hakim, S. J. S., Razak, H. A., Ravanfar, S. A., 2014. "Structural damage identification using Artificial Neural Networks (ANNs) and Adaptive Neuro Fuzzy Interface System (ANFIS)". Proceedings of the 9th International Conference on Structural Dynamics. EURODYN 2014, Porto, Portugal, 30 June-2 July.

[100] Sriram, A., Shahryar, R., and Bourennani, F., 2014. "Artificial neural networks for earthquake anomaly detection". *Journal of Advanced Computational Intelligence and Intelligent Informatics,* **18**(5), September, pp. 701-713.

[101] Ishibuchi, H., and Tanaka, H., 1991. "An extension of the BP-algorithm to interval input vectors". Proceedings of IJCNN, Singapore, **2**, pp. 1594-1599.

[102] Ishibuchi, H., Tanaka, H., and Okada, H., 1993. "An architecture of neural network with interval weights and its application to fuzzy regression analysis". Fuzzy Sets and Systems, **57**(1), July, pp. 27-39.

[103] Kwon, K., Ishibuchi, H., and Tanaka, H., 1993. "Nonlinear mapping of interval vectors by neural networks". *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93), Nagoya, Japan,* **1**, pp. 758-761.

[104] Beheshti, M., Berrached, A., de Korvin, A., Hu, C., and Sirisaengtaksin, O., 1998. "On Interval Weighted Three-Layer Neural Networks". Proceedings of 31st annual simulation symposium, Boston, MA, April 5–9, pp. 188–194.

[105] Garczarczyk, Z. A., 2000. "Interval Neural Networks". IEEE International Symposium on circuits and systems, Geneva, Switzerland, May, 28–31, pp. III567–III570.

[106] Escarcina, R. E. P., Bedregal, B. R. C., and Lyra, A., 2005. "Interval Computing in Neural Networks: One Layer Interval Neural Networks". In Proceedings of

Intelligent Information Technology, 7th International Conference on Information Technology, CIT, Hyderabad, India, pp. 68-75.

[107] Chetwynd, D., Worden, K., and Manson, G., 2006. “An Application of Interval Valued Neural Networks to a Regression Problem”. *Proceedings of the Royal Society A*, **462** (2074), October, pp. 3097–3114.

[108] Wang, X., Yang, H. and Qiu, Z., 2010. “Interval Analysis Method for Damage Identification of Structures”, *AIAA Journal*, **48**(6), pp. 1108-1116.

[109] Zhang, M. Q., Beer, M., and Koh, C. G., 2012. “Interval analysis for system identification of linear MDOF structures in the presence of modeling errors”. *Journal of Engineering Mechanics*, **138**(11), pp. 1326–1338.

[110] Okada, H., Matsuse, T., Wada, T., and Yamashita, A., 2012. “Interval GA for Evolving Neural Networks with Interval Weights and Biases”. SICE annual conference, Akita, August, 20–23, pp. 1542–1545.

[111] Sodhi, S. S., and Chandra, P., 2013. “Interval Based Weight Initialization Method for Sigmoidal Feedforward Artificial Neural Networks”. 2nd AASRI Conference on Computational Intelligence and Bioinformatics, AASRI Procedia, **6**, pp. 19-25.

[112] Lu, J., Xue, S., Zhang, X., and Han, Y. 2015. “A Neural Network Based Interval Pattern Matcher”. *Information*. **6**(3), pp. 388-398.

[113] Chakraverty, S., and Sahoo, D. M., 2014. “Interval response data based system identification of multi storey shear building using interval neural network modelling”. *Computer Assisted Methods in Engineering and Science*, **21**(2), pp. 123-140.

[114] Chakraverty, S., and Behera, D., 2014. “Parameter Identification of Multistorey Frame Structure from Uncertain Dynamic Data”. *Strojniški vestnik - Journal of Mechanical Engineering*, **60**(5), June, pp. 331-338.

[115] Sahoo, D. M., Das, A., and Chakraverty, S., 2014. “Interval data-based system identification of multistorey shear buildings by artificial neural network modelling”. *Architectural Science Review*, **58**(3), February, pp. 244-254.

[116] Chakraverty, S., and Sahoo, D. M., 2016. “Structural parameters identification of uncertain multi-storey shear buildings using fuzzy neural network modelling”. *Inverse Problem in Science and Engineering*, **24**(3), April, pp. 434-452.

[117] Buckley, J. J., and Hayashi, Y., 1992. “Fuzzy neural nets and applications”. *Fuzzy Sets and Systems Artificial Intelligence*, **3**, pp. 11-14.

[118] Hayashi, Y., Buckley, J. J., and Czogala, E., 1993. “Fuzzy neural network with fuzzy signals and weights”. *International Journal of Intelligent Systems*, **8**(4), pp. 527-537.

[119] Buckley, J. J., and Hayashi, Y., 1994. “Fuzzy neural networks a survey”. *Fuzzy Sets and Systems*, **66**(1), August, pp. 1-13.

[120] Buckley, J. J., and Hayashi, Y., 1995. “Neural nets for fuzzy systems”. *Fuzzy Sets and Systems,* **71**(3), May, pp. 265-276.

[121] Ishibuchi, H., Fujioka, R., and Tanaka, H., 1992. “An architecture of neural networks for input vectors of fuzzy numbers”. In: IEEE international conference on fuzzy systems, 8-12 March, pp. 1293–1300.

[122] Ishibuchi, H., Tanaka, H., and Okada, H., 1993. “Fuzzy neural networks with fuzzy weights and fuzzy biases”. In: IEEE international conference on neural networks, 28 March-1 April, **3**, pp. 1650–1655.

[123] Ishibuchi, H., Morioka, K., and Tanaka, H., 1994. “A fuzzy neural network with trapezoid fuzzy weights”. In: IEEE world congress on computational intelligence proceedings of the third IEEE conference on fuzzy systems, 26-29 June, **1**, pp. 228–233.

[124] Ishibuchi, H., Kwon, K., and Tanaka, H., 1995. “A learning algorithm of fuzzy neural networks with triangular fuzzy weights”. *Fuzzy Sets and Systems*, **71**(3), pp. 277–293.

[125] Ishibuchi, H., Morioka, K., and Turksen, I. B., 1995. “Learning by fuzzified neural networks”. *International Journal of Approximate Reasoning*, 13(4), pp. 327–358.

[126] Som, T., and Mukherjee, R. N., 1989. “Some fixed point theorems for fuzzy mappings”. *Fuzzy sets and systems*, **33**(2), November, pp. 213–219.

[127] Pal, S. K., and Mitra, S., 1992. “Multilayer perceptron, fuzzy sets, and classification”. *IEEE Transactions on Neural Networks*, **3**(5), September, pp. 683-697.

[128] Muthukumaran, P., Demirli, K., Stiharu, I., and Bhat, R. B., 2000. “Boundary conditioning for structural tuning using fuzzy logic approach”. *Computers & Structures*, **74**(5), February, pp. 547-557.

[129] Mitra, S., and Hayashi, Y., 2002. “Neuro-fuzzy rule generation: survey in soft computing framework”. *IEEE transactions on neural networks*, **11**(3), August, pp. 748-768.

[130] Mitra, S., and Pal, S. K., 2005. "Fuzzy sets in pattern recognition and machine intelligence". *Fuzzy Sets and Systems*, Invited Position paper commemorating 40th Anniversary of Fuzzy Sets, **156**, pp. 381-386.

[131] Chakraborty, A., Tiwari, S. P., Chattopadhyay, A., and Chatterjee, K., 2014. "Duality in fuzzy multi objective linear programming problem with multi constraint". *International Journal of Mathematics in Operational Research,* **6**(3), pp. 297-315.

[132] Qiu, D., Dong, R., Lu, C., and Mu, C., 2016. "On the stability of solutions of fuzzy differential equations in the quotient space of fuzzy numbers". *Journal of Intelligent and Fuzzy Systems*, **31**(1), June, pp. 45-54.

[133] Rani, D., and Gulati, T. R., 2016. "Application of intuitionistic fuzzy optimization technique in transportation models, OPSEARCH, **53**(4), December, pp. 761–777.

[134] Rani, D., and Gulati, T. R., 2016. "Time optimization in totally uncertain transportation problems". *International Journal of Fuzzy Systems,* April, pp. 1-12,

[135] Lu, P. C., 1998. "The application of fuzzy neural network techniques in constructing an adaptive car-following indicator". AI EDAM, **12**(3), June, pp. 231–241.

[136] Li, Z., Kecman, V., and Ichikawa, A., 2002. "Fuzzified neural network based on fuzzy number operations". *Fuzzy Sets and Systems*, **130**(3), September, pp.291-304.

[137] Yu, W., and Li, X., 2004. "Fuzzy identification using fuzzy neural networks with stable learning algorithms". *IEEE Transactions on Fuzzy System*, **12**(3), June, pp. 411–420.

[138] Er, M. J., Liu, F., and Li, M. B., 2010. "Self-constructing fuzzy neural networks with extended Kalman filter". *International Journal of Fuzzy Systems*, **12**(1), March, pp. 66–72.

[139] Pankaj, D. S., and Wilscy, M., 2011. "Face recognition using fuzzy neural network classifier advances in parallel distributed computing". *Communications in Computer and Information Science*, **203**, pp. 53–62.

[140] Wang, N., 2011. “A generalized ellipsoidal basis function based online self-constructing fuzzy neural network”. *Neural Processing Letters,* **34**(1), August, pp.13–37.

[141] Umoh, U. A., Nwachukwu, E. O., Obot, O. U., and Umoh, A. A., 2011. “Fuzzy-neural network model for effective control of profitability in a paper recycling plant”. *American Journal of Scientific and Industrial Research,* **2**(4), August, pp. 552–558.

[142] Vijay kumar, A., Aruna, Reddy, M. V., 2011. “A fuzzy neural network for speech recognition”. *ARPN Journal of Systems and Software*, **1**(9), December, pp. 284–290.

[143] Sharma, R., and Sinha, A. K., 2012. “A production planning model using fuzzy neural network: a case study of an automobile industry”. *International Journal of Computer Applications,* **40**(4), February, pp. 19–22.

[144] Swaroop, R., 2012. “Load forecasting for power system planning using fuzzy-neural networks”. In: Proceedings of the world congress on engineering and computer science, San Francisco, USA, 24–26 October, **1**, pp. 1–5.

[145] Zhang, H., and Dai, X., 2012. “The application of fuzzy neural network in medicine—a survey”. In: International conference on biology and biomedical sciences, **9**, pp. 111–116.

[146] Malek, H., Ebadzadeh, M. M., and Rahmati, M., 2012. “Three new fuzzy neural networks learning Algorithms based on clustering, training error and genetic algorithm”. *Applied Intelligence,* **37**(2), September, pp. 280–289.

[147] Leu, V. Q., Mwasilu, F., Choi, H. H., Lee, J., and Jung, J. W., 2014. “Robust fuzzy neural network sliding mode control scheme for IPMSM drives”. *International Journal of Electronics*, **101**(7), June, pp. 919–938.

[148] Zahedi, G., Saba, S., Elkamel, A., and Bahadori, A., 2014. “Ozone pollution prediction around industrial areas using fuzzy neural network approach”. *Clean Soil, Air Water*, **42**(7), July, pp. 871–879.

[149] Wang, C. H., and Hung, K. N., 2013. “Intelligent adaptive law for missile guidance using fuzzy neural networks”. *International Journal of Fuzzy System,* **15**(2), June, pp. 182–191.

[150] Sahoo, D. M., and Chakraverty, S., 2013. "Fuzzified data based neural network modeling for health assessment of multistorey shear buildings". *Advances in Artificial Neural System*, **2013**, pp. 1–12.

[151] Chakraverty, S., and Sahoo, D. M., 2016. "Fuzzy neural network based response of uncertain system subject to earthquake motions". *Computational Methods in Earthquake Engineering*, **44**, pp. 363-385.

[152] Pao, Y. H., 1989. *Adaptive Pattern Recognition and Neural Networks*. Addison-Weseley Longman Publishing Co., U.S.A.

[153] Patra, J. C., Pal, R. N., Chatterji, B. N., and Panda, G., 1999. "Identification of nonlinear dynamic systems using functional link artificial neural networks". *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics*. **29**(2), April, pp. 254-262.

[154] Patra, J. C., and Kot, A. C., 2002. "Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks". *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics*, **32**(4), August, pp. 505-511.

[155] Panda, G., and Das, D. P., 2003. "Functional link artificial neural network for active control of nonlinear noise processes". International Workshop on Acoustic Echo and Noise Control (IWAENC2003), Kyoto, Japan, September, pp. 163-166.

[156] Mackenzie, M. R., and Tieu, A. K., 2003. "Hermite neural network correlation and application". *IEEE Transactions on Signal Processing*, **51**(12), December, pp. 3210-3219.

[157] Mackenzie, M., and Tieu, K., 2004. "Gaussian filters and filter synthesis using a Hermite/Laguerre neural network". *IEEE Transactions on Neural Networks* **15**(1), January, pp. 206-214.

[158] Purwar, S., Kar, I. N., and Jha, A. N., 2005. "On-line system identification of complex system using Chebyshev neural networks". *Applied Soft Computing*, **7**(1), January, pp. 364-372.

[159] Ma, L., and Khorasani, K., 2005. "Constructive feedforward neural networks using Hermite polynomial activation functions". *IEEE Transactions on Neural Networks,* **16**(4), July, pp. 821-833.

[160] Misra, B. B., and Dehuri, S., 2007. "Functional link artificial neural network for classification task in data mining". *Journal of Computer Science*, **3**(12), pp. 948-955.

[161] Patra, J. C., Chin, W. C., Meher, P. K., and Chakraborty, G., 2008. "Legendre FLANN based nonlinear channel equalization in wireless communication system". IEEE International Conference on Systems, Man and Cybernetics (SMC 2008), pp. 1826-1831.

[162] Patra, J. C., Meher, P. K., and Chakraborty, G., 2009. "Nonlinear channel equalization for wireless communication systems using Legendre neural networks". *Signal Processing* **89**(11), November, pp. 2251-2262.

[163] Dehuri, S., and Cho, S. B., 2010. "A comprehensive survey on functional link neural networks and an adaptive PSO-BP learning for CFLNN". *Neural Computing and Applications*, **19**(2), March, pp. 187-205

[164] Xiuchun, X., Xiohua, J., and Yunong, Z., 2009. "An algorithm for designing Chebyshev neural network". ISECS International Colloquium on Computing, Communication, Control and Management. pp. 206-209.

[165] Mishra, S. K., Panda, G., Meher, S., and Sahoo, A. K. 2009. "Exponential functional link artificial neural networks for denoising of image corrupted by Gaussian noise". ICACC '09. International Conference on Advanced Computer Control, Singapore, pp. 355-359.

[166] Mishra, S. K., Panda, G., and Meher, S., 2010. "Chebyshev functional link artificial neural networks for denoising of image corrupted by salt and pepper noise". *ACEEE International Journal on Signal and Image Processing*, **1**(1), pp. 413-417.

[167] Patra, J. C., and Bornand, C., 2010. "Nonlinear dynamic system identification using Legendre neural network". The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, pp. 1-7.

[168] Shaik, F. A., Purwar, S., and Pratap, B., 2011. "Real-time implementation of Chebyshev neural network observer for twin rotor control system". *Expert Systems with Applications*, **38**(10), September, pp. 13043-13049.

[169] Dehuri, S., 2011. "A novel learning scheme for Chebyshev functional link neural networks". *Advances in Artificial Neural Systems*, **2011**, pp. 1-10.

[170] Patra, J. C., 2011. "Chebyshev neural network-based model for dual-junction solar cells". *IEEE Transactions on Energy Conversion*, **26**(1), March, pp. 132-139.

[171] Nanda, S. K., and Tripathy, D. P., 2011. "Application of Legendre neural network for air quality prediction". The 5thPSU-UNS International Conference on

Engineering and Technology (ICET-2011), Phuket, May 2-3, Prince of Songkla University, Thailand.

[172] Nanda, S. K., and Tripathy, D. P., 2011. "Application of functional link artificial neural network for prediction of machinery noise in opencast mines". *Advances in Fuzzy Systems,* **2011**, pp. 1-11

[173] Jiang, L. L., 2012. "Chebyshev functional link neural network-based modeling and experimental verification for photovoltaic arrays". WCCI 2012 IEEE World Congress on Computational Intelligence, 10-15 June, Brisbane, Australia, 1-9.

[174] Hassim, Y. M. M., and Ghazali, R., 2012. "Using artificial bee colony to improve functional link neural network training". *Applied Mechanics and Materials,* **263-266**, pp. 2102-2108.

[175] Ali, H. H., and Haweel, M. T., 2012. "Legendre neural networks with multi input multi output system equations". Computer Engineering & Systems (ICCES), 2012 Seventh International Conference on Computer Engineering & Systems, Cairo, pp. 92-97.

[176] Li, W., and Deng, C., 2012. "Fast and robust method for dynamic gesture recognition using Hermite neural network". *Journal of Computers,* **7**(5), May, pp. 1176-1183.

[177] Yuan, X. H., Feng, Q. Y., Zeng, T. J., and Ma, H. B., 2013. "Digital Predistortion for RF power amplifiers based on enhanced Orthonormal Hermite polynomial basis neural network". Progress in Electromagnetics Research Symposium Proceedings, (PIERS), Taipei, March 25-28, pp. 1130-1133.

[178] Parija, S., Sahu, P. K., and Nanda, S. K., 2013. "A functional link artificial neural network for location management in cellular network". International Conference on Information Communication and Embedded Systems (ICICES), Chennai, pp. 1160-1164.

[179] Mishra, M. K., and Dash, R., 2014. "A comparative study of Chebyshev functional link artificial neural network, multi-layer perceptron and decision tree for credit card fraud detection". 13th International Conference on Information Technology. pp. 228-233.

[180] Sharma, V., and Purwar, S., 2014. "Nonlinear controllers for a light-weighted all-electric vehicle using Chebyshev neural network". *International Journal of Vehicular Technology*, **2014**, pp. 1-14.

[181] Mall, S., and Chakraverty, S., 2015. "Chebyshev neural network based model for solving Lane–Emden type equations". *Applied Mathematics and Computation,* **247**, November, pp. 100-114.

[182] Mall, S., and Chakraverty S., 2015. "Numerical solution of nonlinear singular initial value problems of Emden–Fowler type using Chebyshev neural network method". *Neurocomputing,* **149**(B), February, pp. 975-982.

[183] Manu, K., Agarwal, E., and Vashisht, M., 2015. "Discrete-Time Chebyshev Neural Observer for Twin Rotor MIMO System". *International Journal of Engineering Research & Technology,* **4**(8), August, pp. 303-307.

[184] Goyal, V., Deolia, V. K., and Sharma, T. N., 2015. "Robust Sliding Mode Control for Nonlinear Discrete-Time Delayed Systems Based on Neural Network". *Intelligent Control and Automation,* **6**(1), February, pp. 75-83.

[185] Chiang, H. K., and Chu, C. T., 2015. "Reference Model with an Adaptive Hermite Fuzzy Neural Network Controller for tracking a Synchronous Reluctance Motor". *Electric Power Components and Systems,* **43**(7), April, pp. 770-780.

[186] Zurada, J. M., 1994. *Introduction to Artificial Neural Systems*. West Publishing Corporation, U.S.A.

[187] Laurene F., 1994. *Fundamentals of Neural Networks- Architectures, Algorithms, and Applications*. Prentice Hall, U.S.A.

[188] Sivanandam, S. N., Sumathi, S., and Deepa, S. N., 2006. *Introduction to Neural Networks using Matlab 6.0*. McGraw Hill Education Private Ltd, India.

[189] Jaulin, L., Kieffer, M., Didrit, O., and Walter, E., 2001. *Applied Interval Analysis*. Springer, London.

[190] Zimmermann H. J., 1991. *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers, Boston, MA.

[191] Ross, T. J., 2004. *Fuzzy Logic with Engineering Applications*. John Wiley & Sons, New York.

[192] Lee, K. H., 2005. *First Course on Fuzzy Theory and Applications*. Springer, Berlin, Heidelberg.

[193] Moore, R.E., 2009. *Introduction to Interval Analysis*. SIAM, Philadelphia.

[194] Haghighi, A. M., Lian, J., and Mishev, D. P., 2012. *Advance Mathematics for Engineers with Applications in Stochastic Processes*. Nova Science Publishers, New York.

[195] Chakraverty, S., 2014. *Mathematics of Uncertainty Modeling in the Analysis of Engineering and Science Problems*. IGI Global, U.S.A.

[196] Chakraverty, S., Tapaswini, S., and Behera, D., 2016. *Fuzzy Arbitrary Order System: Fuzzy Fractional Differential Equations and Applications*. Willey, New Jersey, U.S.A.

[197] Chakraverty, S., Tapaswini, S., and Behera, D., 2016. *Fuzzy Differential Equations and Applications for Engineers and Scientists*. CRC Press, Boca Ratson, U.S.A.

[198] Rumelhart, D. E., Hinton, G. E., Williams, R. J., 1986. "Learning Representations by Back-Propagating Errors". *Nature*, **323**(9), October, pp. 533-536.

[199] Chopra, A. K., 1981. *Dynamics of Structures: A Primer.* Prentice Hall, U.S.A.

[200] Chakraverty, S., 2009. *Vibration of plates*. Taylor and Francis Group, CRC Press, Boca Ratson, U.S.A.

[201] Newmark, N. M., and Rosenblueth, E., 1971. *Fundamentals of Earthquake Engineering*, Prentice-Hall, Pearson Education, U.S.A.

[202] Gerald, C. F., 2004. *Applied Numerical Analysis*. Pearson Education, India.

[203] Bhat, R. B., and Chakraverty, S., 2011. *Numerical Analysis in Engineering*. Narosa Publication, India.

[204] Patrício, F., Ferreira, J. A., and Oliveira, F., 2003. "On the interval Legendre polynomials". *Journal of Computational and Applied Mathematics*, **154**, September, pp. 215-227.

Dissemination

Journals Articles (Published/Accepted)

1. Sahoo Deepti Moyi, Chakraverty S. (2017) Functional Link neural network learning for response prediction of tall shear buildings with respect to earthquake data, **IEEE Transactions on Systems, Man and Cybernetics: Systems**, 47(10), (**IEEE**);

2. Sahoo Deepti Moyi, Chakraverty S. (2017) Functional Link Neural Network approach to solve Structural System Identification Problems, **Neural Computing and Applications**, 28, 1-12 (**Springer**);

3. Chakraverty S., Sahoo Deepti Moyi (2017) Novel transformation based response prediction of shear building using Interval neural network, **Journal of Earth System Science**, 126(3), 32-38 (**Springer**);

4. Chakraverty S., Sahoo Deepti Moyi (2016) Structural parameters identification of uncertain Multi-storey shear buildings using fuzzy neural network modelling, **Inverse Problem in Science and Engineering**, 24(3), 1-19 (**Taylor and Francis**);

5. Chakraverty S., Sahoo Deepti Moyi (2016) Interval artificial neural network based response of uncertain system subject to earthquake motions, **International Journal of Optimization in Civil Engineering**, 6(3), 365-384 (**Iran University of Science & Technology**);

6. Chakraverty S., Sahoo Deepti Moyi (2015) Fuzzy neural network-based system identification of multi-storey shear buildings, **Neural Computing and Applications**, 27(2), 1-16 (**Springer**);

7. Chakraverty S., Sahoo Deepti Moyi (2014) Interval Response Data Based System Identification of Multi Story Shear Buildings using Interval Neural Network Modeling, **Computer Assisted Methods in Engineering and Science**, 21(2), 123-140, (**Polish Academy of Sciences)**;

8. Sahoo Deepti Moyi, Das Abhishek, Chakraverty S. (2014) Interval Data Based System Identification of Multi Storey Shear Buildings by Artificial Neural Network Modelling, **Architectural Science Review**, 58(3), 244-254 (**Taylor and Francis**);

9. Sahoo Deepti Moyi, Chakraverty S. (2013) Fuzzified Data Based Neural Network Modeling for Health Assessment of Multi Storey Shear Buildings, **Advances in Artificial Neural Systems**, 2013,1-12 (**Hindawi**).

Book Chapter

1. Chakraverty S., Sahoo Deepti Moyi (2016) Fuzzy Neural Network Based Response of Uncertain System Subject to Earthquake Motions, Editor: Manolis Papadrakakis, Vagelis Plevris, Nikos D. Lagaros, Springer International Publishing, pp 363-385, , DOI: 10.1007/978-3-319-47798-5_13 (**Springer**).

Conference Presentations

1. Sahoo Deepti Moyi, Das A., Chakraverty S., Interval Data Based System Identification of Multi Storey Shear Buildings by Artificial Neural Network Modeling, National Conference on ‘Computational and Applied Mathematics in Science and Engineering (CAMSE-2012)’, VNIT Nagpur, Maharastra, December 21-22 (2012);

2. Sahoo Deepti Moyi, Chakraverty S., Health Assessment of Two Storey Shear Building by Neural Network Modeling using Fuzzified Data, ‘40th Annual Conference of Orissa Mathematical Society and National Conference on Fourier analysis and Differential equations (OMSNCFD-2012)’, School of Mathematical Sciences, Sambalpur University, Odisha, December 29-30 (2012);

3. Chakraverty S., Sahoo Deepti Moyi, Interval Response Data Based System Identification of Two Storey Shear Building Using Interval Neural Network Modelling, ‘Fuzzy and Interval Based Uncertainty Modeling (FIUM-2013)’, National Institute of Technology Rourkela, Odisha, August 17-19 (2013);

4. Chakraverty S., Sahoo Deepti Moyi, Interval Response Prediction of Structural System Subject to Earthquake Motions Using Interval Artificial Neural Network, ‘National Seminar on Recent Advances in Nonlinear Analysis and Optimization (ANAO-2014)’, Pt. Ravishankar Shukla University, Raipur, February 15-17 (2014);

5. Chakraverty S., Sahoo Deepti Moyi, Identification of Uncertain Mass and Stiffness Matrices of Multi-Storey Shear Buildings Using Fuzzy Neural Network Modelling, ‘Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO - 2015)’ (**Springer**), Institute for Development and Research in Banking Technology, Hyderabad, December 17-19 (2015).

Journals (Communicated)

1. Sahoo Deepti Moyi, Chakraverty S. (2017) Estimation of structural parameters using Chebyshev Neural Network Model, Neural Processing Letters.

2. Chakraverty S., Sahoo Deepti Moyi (2017) Uncertain Structural Parameter Identification by Intelligent Neural Training, Fuzzy Information and Engineering.

Journal (To be communicated)

1. Chakraverty S., Sahoo Deepti Moyi (2017) Uncertain Structural Parameter Estimation by Interval Functional Link Neural Network.