UTILIZATION OF ARTIFICIAL NEURAL NETWORKS IN BUILDING DAMAGE PREDICTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY
BARIŞ ERKUŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF CIVIL ENGINEERING

SEPTEMBER 1999

Approval of the Graduate School of Natural and Applied Science

Prof. Dr. Tayfur ÖZTÜRK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Mustafa TOKYAY
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of sciences.

Prof. Dr. Polat GÜLKAN
Supervisor

Examining Committee Members

Prof. Dr. Engin KEYDER

Prof. Dr. Polat GÜLKAN

Prof. Dr. Haluk SUCUOĞLU

Assoc. Prof. Dr. Uğur POLAT

Dr. Engin KARAESMEN

ABSTRACT

UTILIZATION OF ARTIFICIAL NEURAL NETWORKS IN BUILDING DAMAGE PREDICTION

ERKUŞ, Barış
M.S., Department of Civil Engineering
Supervisor: Prof. Dr. Polat GÜLKAN

September 1999, 121 Pages

In this study effects of several characteristics of structural systems such as shear wall, column, beam parameters on structural damage of generalised yielding structural systems under earthquake effects are examined. In the first part of the study one frame with shear wall, two beam and column frames and a SDOF system are analysed for nonlinear damage under the EL Centro earthquake ground motion. A modified version of Park and Ang damage model is employed for this purpose. In the second part of the study, artificial neural networks, are utilised for the implementation of damage analysis. Two types of multilayer neural network models are used. In the first model the damage indices are estimated by two multilayer networks trained with two different training sets obtained from the first part of the study. In the second model a new approach is used. In this model the problem of damage classification is implemented. For both models learning rate and period concepts are introduced.

It is found that shear wall, column, damping and structural mass are the most important structural parameters that effect damage. This effect is more evident when PGA of the earthquake used in the analyses is larger than 0.2g. It is

It is found that shear wall, column, damping and structural mass are the most important structural parameters that effect damage. This effect is more evident when PGA of the earthquake used in the analyses is larger than 0.2g. It is observed that the design according to TS-500, EUROCODE-2 and ACI results in the most economical design. Performance of the neural network implementation of the damage indices is showed to be highly dependent on the size of the training set. The new model proposed is observed to be an efficient tool for damage classification problem.

Keywords: damage, damage model, nonlinear dynamic structural analysis, artificial neural networks.

ÖZ

BİNA HASAR TAHMİNLERİNDE YAPAY SİNİR AĞLARININ KULLANILMASI

ERKUŞ, Barış
Yüksek Lisans Tezi, İnşaat Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Polat GÜLKAN

Eylül 1999, 121 Sayfa

Bu çalışmada yapı sistemlerinin perde duvar, kolon, kiriş gibi çeşitli özelliklerinin, deprem etkileri altında meydana gelen hasar üzerindeki etkileri araştırılmıştır. Çalışmanın birinci bölümde bir adet perde duvarlı çerçeve, iki adet kolon ve kirişlerden meydan gelen çerçeve, ve bir adet tek serbestlik dereceli sistem, doğrusal olmayan hasar analizi yapacak şekilde El Centro deprem kaydı kullanılarak hesaplanmıştır. Bu amaçla değiştirilmiş Park ve Ang hasar modeli kullanılmıştır. Çalışmanın ikinci bölümde yapay sinir ağları hasar hesabı için kullanılmıştır. Çalışmalar iki farklı yapay sinir ağı modeli üzerinde yoğunlaşılmıştır. Birinci modelde çok katmanlı bir ağ, birinci bölümde elde edilen iki farklı öğrenme kümesi kullanılarak hasar değeri hesaplamasında kullanılmıştır. İkinci bölümde yeni bir yaklaşım denenmiştir. Bu modelde hasar sınıflandırma problemi irdelenmiştir. Her iki model için öğrenme hızı aralığı ve periyodu konuları tanıtılmıştır.

Perde duvar, kolon, sönüm yüzdesinin ve yapısal kütlenin, yapısal hasarı etkileyen en önemli parametreler olduğu saptanmıştır. Bu etkinin El Centro depreminin en yüksek yer ivmesinin 0.2g değerinden daha büyük olması durumunda arttığı gözlemlenmiştir. TS-500 EUROCODE-2 ve ACI yapı yönetmeliklerine göre yapılan tasarımların en ekonomik dizayn parametrelerini sağladığı saptanmıştır. Yapay sinir ağlarının performasının öğrenme kümesinin büyüklüğünden çok etkilendiği gösterilmiştir. Önerilen yeni modelin hasar sınıflandırma probleminde oldukça başarılı olduğu gözlemlenmiştir.

Anahtar kelimeler: hasar, hasar modeli, lineer olmayan dinemik hesap, yapay sinir ağları.

to my family...

ACKNOWLEDGMENTS

I would like to express my deepest respect to Prof. Dr. Polat Gülkan for his great guidance and help throughout this study, to Dr. Erhan Karaesmen and Dr. Engin Karaesmen for their great help for the attendance to CISM course in Italy and for their great guidance on this study, to Prof. Dr. Uğur Halıcı for her course EE543 on Neural Networks and for guidance on this study.

I wish to express my sincere appreciation to Egemen Computer Co., especially to İsmail Meral Egemen for his great support. All the computations were done by the computer supplied by him without any charge.

I would like to thank to Sinan Akkar for his warnings and help about this study.

I would like to thank to my instructors Prof. Dr. Polat Gülkan, Prof. Dr. Tanvir Wasti, Prof. Dr. Mehmet Utku, Assoc. Prof. Dr. Saleh Abdalla, Prof. Dr. Yalçın Mengi, Prof. Dr. Uğur Ersoy, Prof. Dr. Güney Özcebe, Prof. Dr. Tuğrul Tankut, Prof. Dr. Ahmed Khdeir, Prof. Dr. Ergin Atımtay, Dr. Engin Karaesmen for the courses that I have taken.

I would like to thank to my friends Devrim Türker, Murat Nalçacı, Ömer Erbay, Taner Kolçak, Taner Karagöl, Mehmet Ali Somuncu, Özgür Kurç, Erdem Canbay, Onur Sonuvar, Youssef Hamdan, Evren Yıldız, Bora Uşkay, Eygü Mındıkoğlu, and other friends that I forgot. Thanks to CEIC crew and other student assistants.

I wish to express my gratitude to my parents for their patience and support.

I wish to thank to M. Kemal Atatürk for this beautiful country that he left us to protect. No way other than science and scientific thinking style!

I would like to thank to the bands Sepultra, Metallica, Megadeth, Slayer, In Flames, Over Kill, and the others for their support with their music.

TABLE OF CONTENTS

6 CONCLUSIONS AND RECOMMENDATIONS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

CHAPTER 2

a_g	:Ground acceleration
A_{S1}	:Area of first steel layer of column
A_{S2}	:Area of second steel layer of column
$\mathbf{C}$	:Viscous damping matrix
d	:Depth of column
dE	:Incremental absorbed energy
D	:Damage index
DI	:Damage index
EC	:Initial Young's Modulus of concrete
ECN	=FCxZF
EI_0	:Elastic rotational stiffness at the center of the structural element
EI_a, EI_b	:Tangential rotational stiffness at the ends of the element defined by yield penetration model
E_h	:Dissipated energy
$EPS0$	:Strain at maximum strength of concrete
$EPSH$	:Strain at start of hardening of steel
$EPSU$	:Ultimate strain in compression of concrete
ES	:Modulus of elasticity of steel
ESH	:Modulus of strain hardening
f_0	:Elastic flexibility at the center of the structural element
f_a, f_b	:Flexibility at the ends of the structural element
f'_{aa}, f'_{ab}, f'_{bb}	:Flexibility coefficients
f'_{AA}, f'_{BB}, f'_{AB}	:Elements of element flexibility matrix that relates θ_A, θ_B to M_A, M_B

FC	:Unconfined compressive strength of concrete
FS	:Yield strength of steel
FSU	:Ultimate strength
GA_z	:Shear stiffness of the structural element
[K']	:Element stiffness matrix relates end moments to end rotations at the face of the structural element
k_{aa}, k_{ab}, k_{ba}, k_{bb}	:Elements of matrix K'
$[\mathbf{K_e}]$	Final element stiffness matrix includes shear and axial force-deformation relations
$[\mathbf{K_s}]$	:Element stiffness matrix relates end moments to end rotations at the nodes
$[\mathbf{K_t}]$	:Global stiffness matrix
l	:Element length
L	:Element length
L	:Allocation vector for horizontal ground acceleration
$m_i(x)$, $m_j(x)$	Moment distributions due to virtual moment at the ends of the element
M	:Lumped mass matrix
M_a, M_b	:Moments at the ends of the structural element
M_y	:Yield moment
p_t	:Longitudinal steel ratio as percentage
PGA	:Peak ground acceleration
Q_y	:Yield strength
Q_u	:Ultimate strength
u_a, u_b	:Degrees of freedom that corresponds to the moments X_a, X_b
X_a, X_b	:Forces at the ends of the structural element in the global X direction
v_a, v_b	:Degrees of freedom that corresponds to the moments Y_a, Y_b
$v_i(x)$, $v_j(x)$	:Shear distributions due to the virtual shear at the ends of the element
Y_a, Y_b	:Forces at the ends of the structural element in the global Y direction

ZF	:Parameter defining slope of falling branch of stress-strain curve of unconfined concrete
α_m, α_k	:Parameters for expression of damping matrix
λ	:Energy weighting factor
θ_a, θ_b	:Degrees of freedom that corresponds to the moments M_a, M_b
θ_m	:Maximum rotation
θ_u	:Ultimate rotation capacity
θ_r	:Recoverable rotation during unloading
Δu	:Incremental displacement vector
$\Delta\dot{u}$	:Incremental velocity vector
$\Delta\ddot{u}$	:Incremental acceleration vector
$\Delta\ddot{x}$	:Incremental horizontal ground acceleration
ω_i	:Circular frequency of mode I
ξ	:Critical damping
δ_m	:Maximum deformation
δ_u	:Ultimate deformation
β	:An empirical parameter used in Park and Ang damage model
ρ_w	:Confinement ratio(horizontal steel ratio)

CHAPTER 3

a	:Sigmoid constant
a, b	:Synaptic weights
d_j	:Expected outputs of the second layer
d_E	:Euclidean distance between two vectors
e	:Error of the neuron output
$f(t)$, $f(x)$	:Activation function
L	:Number of layers
L	:Learning set
o	:Expected output of the neuron
o'	:Neuron output

R	:Subspace of X that a relation between the inputs and outputs is exists
T	:Training set
w_{ij}	:Synaptic weight connecting the output of neuron i in the first layer to the input of neuron j in the second layer
x_i	:Inputs for any two layer of a multilayer network
X	:Input space for the physical system
y_i	:Outputs for any two layer of a multilayer network
θ	:Bias
ξ	:Sum of the squared error terms
η	:Learning rate
α	:Momentum constant
δ^k	:Error vector for the kth layer
Σ	:Activation function

CHAPTER 4, 5

BDP	:Beam dimension percent
BSP	:Beam steel percent
CDP	:Column dimension percent
CSP	:Column steel percent
DI	:Damage indices
DR	:Damping ratio
FR1	:Frame 1
FR2	:Frame 2
FR3	:Frame 3
MP	:Mass percent
MP	:Mass proportional damping
NN-1a	:First neural network with 84 training data
NN-1b	:First neural network with 445 training data
NN-2	:Second neural network

PGA	:Peak ground acceleration
PGD	:Peak ground distance
PGV	:Peak ground velocity
RZ	:Rayleigh damping
SDOF	:Single degree of freedom example
SP	:Stiffness proportional damping
SWAR	:Shear wall area ratio
SWSR	:Shear wall steel ratio

CHAPTER 1

INTRODUCTION

1.1 General

Damage detection is an important task that is utilised for post-earthquake assessment of structures, disaster planning and design code implementations. Several attempts have been made to establish a sound prediction of the damage of structures experienced earthquakes including visual and numerical inspection of damage.

Post earthquake assessment of structural system helps the engineers to estimate the vulnerability of the buildings to aftershocks and hence enabling them to decide whether the building should be demolished or not. Disaster planning aims to estimate the cost of a possible earthquake, number of casualties, amount of temporary accommodation needed etc. Finally damage assessment is a very helpful tool in the design phase in the sense that it provides considerations on a wider range of design objectives, adjusting the design criteria according to a performance based format. Other then these practical applications, damage analysis can be an explanatory guide for several researches.

Every method proposed for damage detection has its own problems. Although visual detection of damage may be very simple and quick for small-scale structures, it may be unreliable, subjective and hard to determine for large-scale structures. Problems with visual inspection directed the researchers to study on numerical detection of damage, which is basically based on the estimation of the response history and change in the dynamical characteristics of the earthquake experienced building. More or less the methods followed to obtain a numerical value of damage are same. A mathematical model of the structure and a damage model are constructed first. The structure is exposed to an earthquake loading whose parameters such as peak ground acceleration, dominant period or effective duration are known and observing the changes in different characteristics of the structure will lead to numerical values of element and structure level damages namely local and global damage indices respectively.

The measures employed in damage models that the structure owes such as ductility ratio, interstorey drift, accumulated plastic deformation and energy dissipated during the cyclic response, changes in stiffness and ultimate strength, bring important problems concerning the reliability of the result. First of all, these parameters of the structure that is exposed to earthquake should be known which is a challenging job. For example to estimate the energy dissipated during the earthquake will be almost impossible if special devices which will determine the response are not placed on the structure prior to earthquake, which is nearly true for other parameters. This is obviously unpractical and uneconomical way to detect the damage in public sense. Second, the mathematical models that are used to simulate the nonlinear response of structures to eliminate the in-situ estimation of the structural parameters should give the nearly exact response of structure. which is ever challenging problem of the structural engineering. Third the parameters that have no units in damage models should give realistic results. These parameters are proposed after statistical estimations on some experimental data. And finally, the damage classification should be universal which means that it should be applicable for all kind of structures.

One can increase the number of the problems that are faced during the damage assessment. All these problems concerning with the visual or numerical damage analysis lead the researchers to study on more practical, economical, quick but reliable damage assessment techniques. One of these techniques, namely damage assessment using neural networks, aims to predict the damage numerically or just by doing a classification, using a database of previous earthquakes and damage data but without making a nonlinear dynamic analysis. Studies on this concept is very new to structural engineering although artificial neural networks have become a very popular and powerful tool for several engineering areas such as electrical, mechanical and aerospace engineering. Very little studies on neural network implementation of damage analysis have shown that neural networks will work well if the neural network model is designed very well.

What basically neural networks do is to classify similar data or to make an interpolation or an extrapolation in an *n*-dimensional space or after a process called "training" which is done using a pre-estimated input-output set, namely training set, where the input will be the earthquake and structural parameters and the output will be the damage. The performance of the neural networks should be checked by comparing the results it gives for the earthquakes and structures that it never "meet" before with the actual results.

Although utilising these methods is a very simple task, and reliable results can be obtained for some kind of problems, one should always keep in mind that this will not mean that the same neural networks model will give reliable results for different types of systems or problems. Moreover since these methods are very new to structural engineering society, they need deeper research and stronger understanding of neural networks concepts.

1.2 Object and Scope

The main object of the current study is to investigate the effects of different parameters on damage of reinforced concrete structures and observe the performance of the neural network implementation of damage analysis. For this purpose software called IDARC for the two dimensional, inelastic damage analysis of reinforced concrete frames and software for the neural network analysis are used. In the first part, three reinforced concrete frames and a single degree of freedom system are analysed with IDARC to observe the performance of the software. The effects of the peak ground acceleration, column, beam and shear wall parameters, damping and mass distribution on the damage are observed. The damage model used in the software is modified version of Park and Ang model. In the second part of the study two neural network implementations are studied. In the first model the numerical estimation of damage is studied, while in the second model the classification of damage is tested.

CHAPTER 2

IDARC2D: A COMPUTER PROGRAM FOR THE INELASTIC DAMAGE ANALYSIS OF BUILDINGS

2.1 General

Developments in the computer technology have guided the research in structural engineering from an experimental based understanding of member and structure behaviour to an experimental validation of computer based models of members and structures. Several nonlinear analysis software is produced to predict the exact behaviour of the structures. Kaanan and Powell introduced the most well known software DRAIN-2D in 1973 to perform a nonlinear time-history analysis. The other software, like SARFC, ANSR, IDARC are reported to be used in the research fields [1, 9]. These software did not become popular in the engineering market since the reliability of the software are not verified well and they are not aimed to be used by end-users but only researchers.

In this study it was decided to use IDARC since it enables modelling of shear walls and damage analysis in addition to other analysis options. In this chapter a review of the element modelling and theory of structural and damage analysis will be given. Additional information can be found in the technical report of IDARC [9].

2.2 Element Models

In IDARC, each type of element has its own stiffness formulation. These macro formulations are then combined to obtain the stiffness matrix of the whole structure. Element stiffness matrices are updated according to the model called spread plasticity model during the nonlinear time history analyse. The general formulation of the stiffness matrices of the elements that relates end moments and the end rotations at the face of structural element is given by Equation 2.1.

$$[\mathbf{K}'] = \begin{bmatrix} k_{aa} & k_{ab} \\ k_{ba} & k_{bb} \end{bmatrix} \tag{2.1}$$

Where,

$$k_{aa} = \frac{12EI_0EI_aEI_b}{D_{et}L}(f'_{bb}GA_zL^2 + 12EI_0EI_aEI_b) \tag{2.2}$$

$$k_{ab} = k_{ba} = \frac{-12EI_0EI_aEI_b}{D_{et}L}(f'_{ab}GA_zL^2 + 12EI_0EI_aEI_b) \tag{2.3}$$

$$k_{bb} = \frac{12EI_0EI_aEI_b}{D_{et}L}(f'_{aa}GA_zL^2 + 12EI_0EI_aEI_b) \tag{2.4}$$

and EI_0 is elastic rotational stiffness at the center of the element; EI_a and EI_b are the tangential rotational stiffnesses at the ends of the element; GA_z is the shear stiffness of the element; L is the length of the element; f_{aa}, f_{ab}, f_{bb} are the flexibility coefficients. Derivations and formulations of f_{aa}, f_{ab}, f_{bb} D_{et} are given in the technical report [9]. How this element matrix will be used to obtain the structural stiffness matrix will be explained briefly later in this chapter.

In this study only the column, beam, shear wall and edge column elements are used. In IDARC, these elements are modelled considering flexural, shear and axial deformations. Hysteretic flexural and shear deformations are modelled by three-parameter Park model while the default model of axial deformations is linear-elastic spring in this software. A rigid zone can be defined for the column and beam elements to increase the stiffness at the joints. To employ the spread plasticity model when finding **K'**, moment curvature relations of the cross-sections at the ends of the members should be obtained which is done by obtaining the parameters of a simplified trilinear relation whose graph is given in Figure 2.1. These parameters can be given as inputs by the user or can be estimated by the software using formulations defined for every type of element. This simplified version is used for other shear-drift relations also.

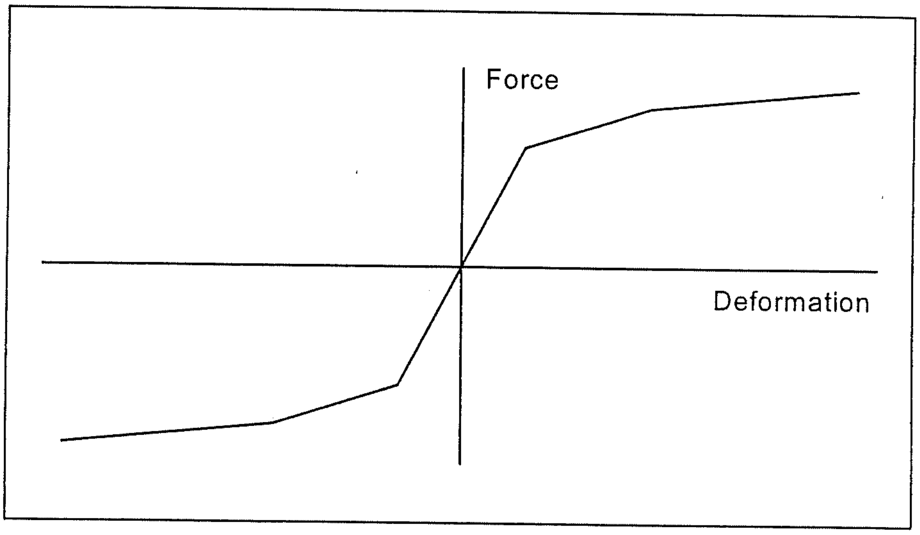

Figure 2.1 Simplified trilinear force-deformation relation used in IDARC

The degrees of freedom for the column, beam and shear wall macro elements are given in Figure 2.2, Figure 2.3, and Figure 2.4, respectively. Edge columns are modelled as axial springs. Hence they have only axial deformations.

The stiffness matrix of the element is transformed into another stiffness matrix $\mathbf{K}_s$, which relates the moments and rotations at the nodes. Considering the force equilibrium of all the forces perpendicular to the axis of the element, the final stiffness matrix $\mathbf{K}_e$ is obtained for shear and moments and the corresponding deformations by some algebra.

2.3 Hysteretic Rules

The hysteretic behaviour of the elements is modelled by three-parameter Park model in IDARC [9]. This model integrates three parameters namely, strength degradation, stiffness degradation and pinching into the hysteretic behaviour (Figures 2.5, 2.6, 2.7). The element stiffness matrices are updated according this hysteretic behaviour during the nonlinear analysis.

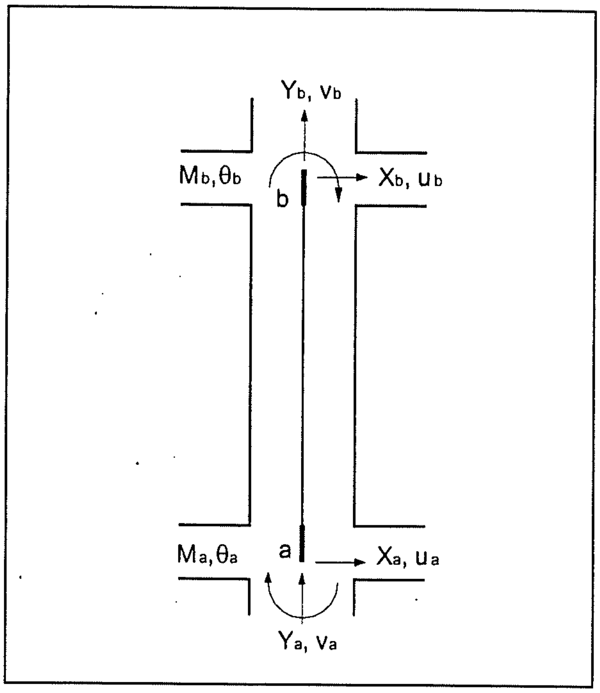

Figure 2.2 Degrees of freedom of a column element used in IDARC

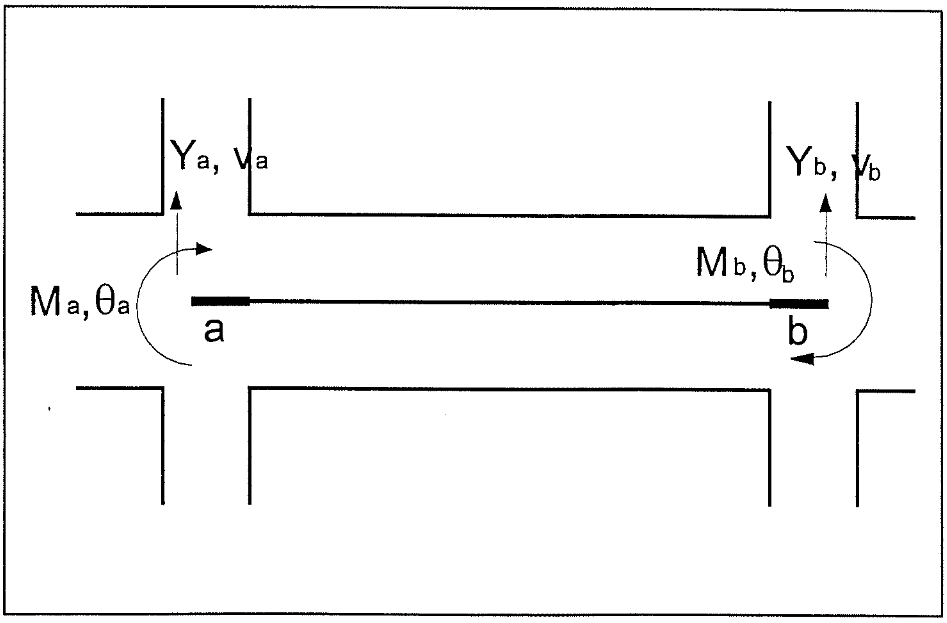

Figure 2.3 Degrees of freedom of a beam element used in IDARC

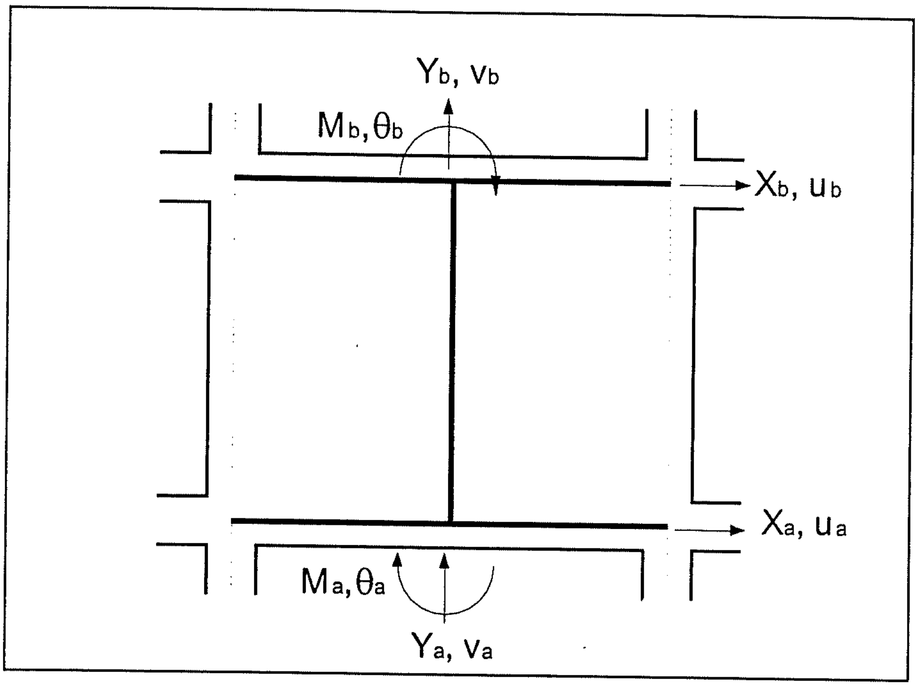

Figure 2.4 Degrees of freedom of a shear wall element used in IDARC

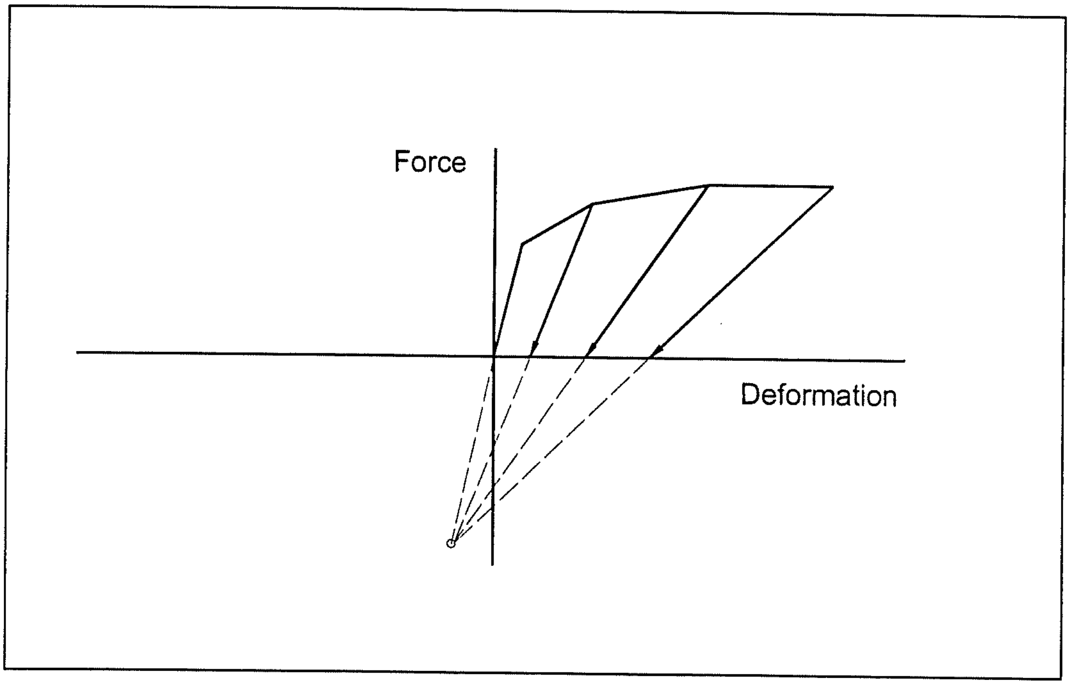

Figure 2.5 Stiffness degradation modelling in IDARC

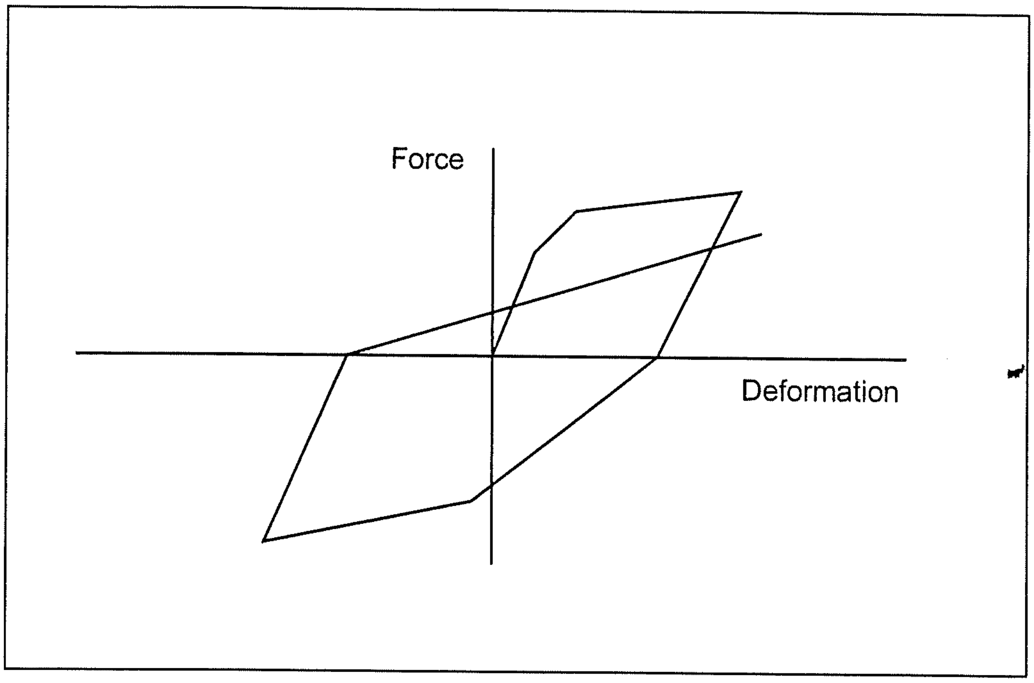

Figure 2.6 Strength degradation modelling used in IDARC

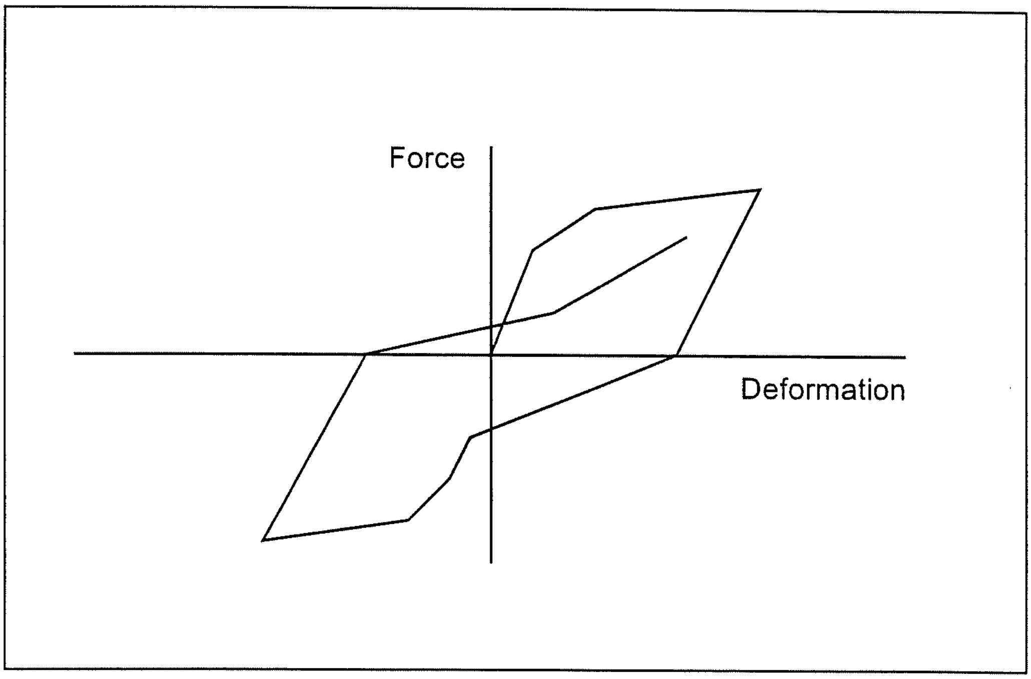

Figure 2.7 Pinching modelling used in IDARC

2.4 Analysis

There are four options for analysis in IDARC. In this study Nonlinear Dynamic Analysis is used. In this type of analysis first of all the element stiffness matrices and trilinear moment curvature relations are obtained using initial axial loads on the columns supplied by the user. Second the global stiffness matrix is obtained. After calculation of all these matrices the following equation is solved using a combination of Newmark-Beta integration method and pseudo-force method.

$$[\mathbf{M}]\{\Delta\ddot{\mathbf{u}}\}+[\mathbf{C}]\{\Delta\dot{\mathbf{u}}\}+[\mathbf{K}_t]\{\Delta\mathbf{u}\}=-[\mathbf{M}]\{\mathbf{L}\}\Delta\ddot{x}_g \tag{2.5}$$

Where **M** is the lumped mass matrix of the structure; **C** is the viscous damping matrix of the structure; **K**$_t$ is the tangent stiffness matrix; $\Delta\mathbf{u}, \Delta\dot{\mathbf{u}}$ and $\Delta\ddot{\mathbf{u}}$ are the incremental vectors of displacement, velocity and acceleration in the structure respectively; **L** is the allocation vector for the horizontal ground accelerations; $\Delta\ddot{x}$ is the incremental horizontal ground acceleration. The solution is carried out incrementally considering the changes in the element stiffness matrix.

2.5 Spread Plasticity Model

This model is employed in IDARC to merge the different flexibility characteristics of the various sections throughout the element into the element stiffness matrix. This will integrate the effects of inelastic deformations that occur near the joints into the matrix, which will result in a different curvature distribution than that an elastic behaviour will cause.

Assumed distribution of the element flexibility is shown in the Figure 2.8. The geometry of this diagram is determined through a model called yield penetration model in IDARC.

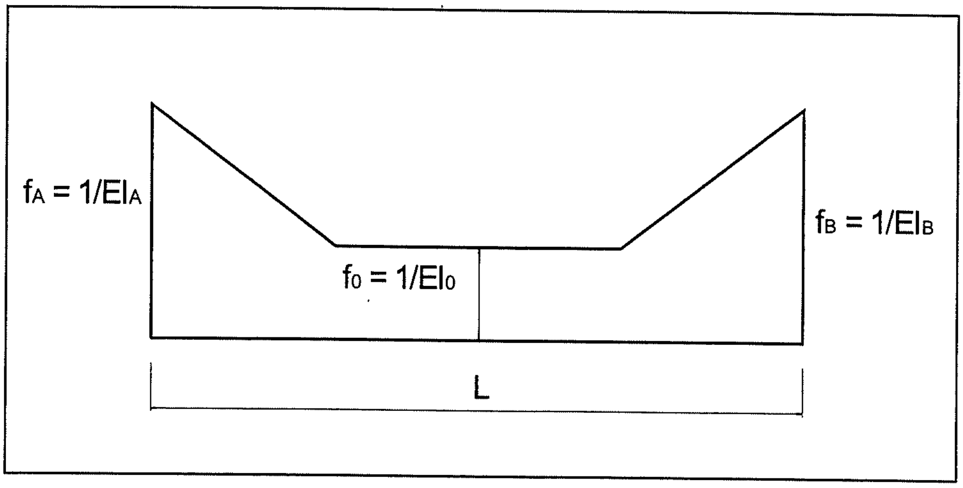

Figure 2.8 Spread plasticity model used in IDARC

The flexibility matrix including shear distortions, relating moments and rotations at the ends of the element is given as follows:

$$\begin{Bmatrix} \theta_A \\ \theta_B \end{Bmatrix} = \begin{bmatrix} f_{AA} & f_{AB} \\ f_{BA} & f_{BB} \end{bmatrix} \begin{Bmatrix} M_A \\ M_B \end{Bmatrix} \tag{2.6}$$

The flexibility coefficients are obtained from

$$f_{ij} = \int_0^L \frac{m_i(x) m_j(x)}{EI(x)} dx + \int_0^L \frac{v_i(x) v_j(x)}{GA_z} dx \tag{2.7}$$

where $m_i(x)$ and $m_j(x)$ are the moment distributions due to the virtual moment at the ends of the element, while $v_i(x)$ and $v_j(x)$ are the corresponding shear distributions. These integrations are evaluated to find expressions for f_{AA}, f_{AB}, f_{BA}, f_{BB} and f'_{AA}, f'_{AB}, f'_{BA}, f'_{BB} and D_{et}. Expressions for this parameters can be found in the technical report.

2.6 Damage Analysis

In this study a modified version of Park and Damage model which is directly supplied by IDARC is used. Hence a detailed discussion on this model will be given here.

2.6.1 Park and Ang Damage Model

This model was introduced by Park and Ang [6] in 1984. They proposed that damage could be expressed as a linear combination of excessive deformation that can be called as normalised deformation and damage contributed by repeated cyclic loading effect that can be called as energy absorption. This is represented by a damage index

$$D = \frac{\delta_M}{\delta_u} + \frac{\beta}{Q_y \delta u} \int dE \qquad (2.8)$$

where δ_M is the maximum deformation under earthquake; δ_u is the ultimate deformation under monotonic loading; Q_y is the calculated yield strength (if the maximum strength, Q_u, is smaller than Q_y, Q_y is replaced by Q_u.); dE incremental absorbed energy; β is a non negative parameter that is obtained from statistical analysis of past experiments. One can easily observe that if maximum deformation is equal to ultimate deformation, which is the collapse case, damage index will be greater than one because the energy term will be almost β. The yielding and ultimate parameters are determined by IDARC as described in above parts. The other earthquake loading dependent parameters are calculated during the execution of the program. Although the formulation of β for reinforced concrete sections is given by Park and Ang [6] as in Equation 2.9.

$$\beta = \left(-0.447 + 0.073 \frac{l}{d} + 0.24 n_o + 0.314 p_t \right) \times 0.7^{\rho_w} \qquad (2.9)$$

Where l/d is shear span ratio (should be taken as 1.7 if it is smaller than 1.7); n_o normalised axial stress (should be taken as 0.2 if it is smaller than 0.2); p_t is longitudinal steel ratio as percentage (should be taken as 0.75% if it is smaller than 0.75%); and ρ_w is confinement ratio, Park et al. [6] suggested a value 0.1. In IDARC the second choice is used.

This model is used for local damage indices. Then the element damage indices are combined to obtain the global damage indices.

The classification of the damage using the above model is given by Park et al. [6] as in Table 2.1 and by Stone and Taylor [7] as in Table 2.2.

Table 2.1 Classification of damage according to Park et al. [6]

Damage Index	**Degree of Damage**	**Physical Appearance**
$D < 0.1$	SLIGHT	Sporadic Occurrence of Cracking.
$0.1 \leq D < 0.25$	MINOR	Minor Cracks Throughout Building. Partial Crashing of Concrete in Columns.
$0.25 \leq D < 0.4$	MODERATE	Extensive Large Cracks. Spalling of Concrete in Weaker Elements.
$0.4 \leq D < 1.0$	SEVERE	Extensive Crushing of Concrete. Disclosure of Buckled Reinforcements.
$D \geq 1.0$	COLLAPSE	Total or Partial Collapse of Building.

Table 2.2 Classification of damage according to Stone and Taylor [7]

Damage Index	**Degree of Damage**	**Physical Appearance**
$D < 0.11$	SLIGHT	No Damage or Localised Minor Cracking
$0.11 \leq D < 0.4$	REPAIRABLE	Extensive Spalling but Inherent Stiffness Remains.
$0.4 \leq D < 0.77$	IRREPAIRABLE	Still Standing but Failure Imminent
$D \geq 0.77$	COLLAPSE	Total or Partial Collapse of Building.

2.6.2 Modified Park and Ang Model in IDARC

A slight modification was introduced to the original model by Kunnath et al. [3] in the third version of IDARC since the relation between element, story or top story deformations with the local plastic rotations is difficult to establish due to the inelastic response is confined by plastic regions near the ends of some members.

The modified version is given by Equation 2.10.

$$D = \frac{\theta_m - \theta_r}{\theta_u - \theta_r} + \frac{\beta}{M_y \theta_u} E_h \tag{2.10}$$

Where θ_m is the maximum rotation attained during the loading history; θ_u is the ultimate rotation capacity; θ_r is the recoverable rotation when unloading; M_y is the yield moment; and E_h is the dissipated energy in the section. The element damage index is then selected as the largest damage index of the end sections.

Story level damage indices should be calculated to obtain an overall structural damage indices.

$$D_{storey} = \sum (\lambda_i)_{component} (D_i)_{component} \tag{2.11}$$

$$(\lambda_i)_{component} = \left(\frac{E_i}{\sum E_i} \right)_{component} \tag{2.12}$$

$$D_{overall} = \sum (\lambda_i)_{storey} (D_i)_{storey} \tag{2.13}$$

$$(\lambda_i)_{storey} = \left(\frac{E_i}{\sum E_i} \right)_{storey} \tag{2.14}$$

Where λ_i are the energy weighting factors E_i are the total energy absorbed by the component of storey " i ".

2.7 Program Input

The structural parameters are supplied to the program by a text file. These parameters include the shape of the structure, concrete and reinforcements properties, parameters that define the hysteretic behaviour of cross-sections, column, beam, shear wall and edge column parameters, dynamic analysis parameters, earthquake ground motion parameters etc.

2.7.1 Concrete Parameters

User should enter the parameters that define the stress-strain relation of concrete. But in the technical report, it is not clear whether the unconfined or the confined parameters of the concrete should be entered as input to the program. In Section 3.2.2.2 the model that is used is explained. According to this definition the stress-strain diagram for unconfined concrete is given by Figure 2.9.

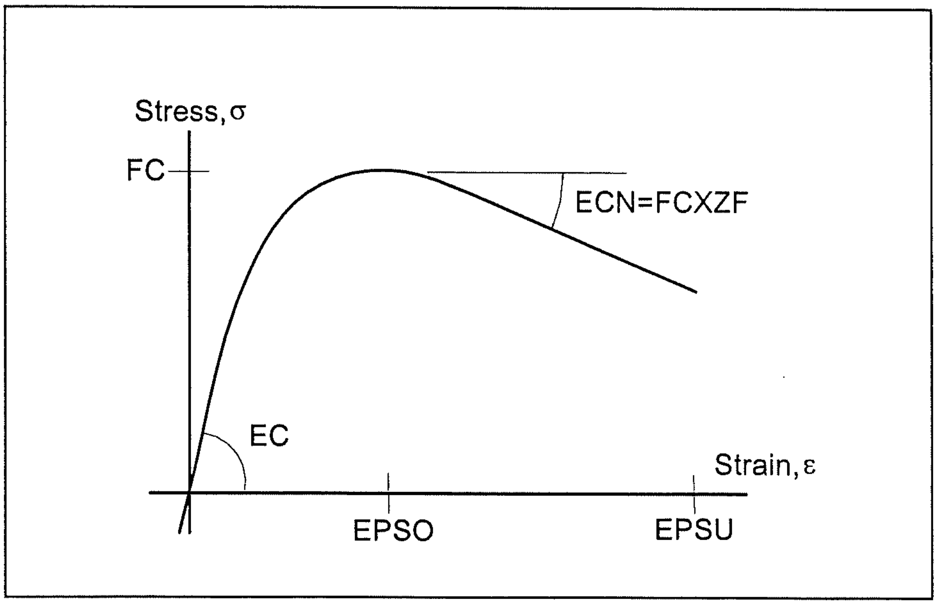

Figure 2.9 Stress-strain curve for unconfined concrete used in IDARC

Since the confinement ratio does not significantly affect the maximum compressive stress, it is considered to influence only the ultimate strain, which is defined by the ZF factor.

The parameters for the input are explained in the technical report. In this part it is stated that "EPSU and ZF are derived from equation (3.12) and depends on section data". This is done if EPSU and ZF are given as zero by the user. This means that EPSU may be dependent on the horizontal reinforcement properties although it seems to be a parameter for unconfined concrete in the figures Fig.3.8 and Fig. A.3 of the report. The case studies supplied by the report are examined to understand the value of EPSU. In some case studies EPSU is given directly while the in the others it is calculated by IDARC. Table 2.3 gives the values for EPSU. Since the envelope properties of the sections are given by the user inputs, EPSU is not given directly.

Table 2.3 EPSU values for some case studies

CASE STUDY	EPSU	DEFAULT(D) OR USER SPECIFIED(U)
1	0.03	D
3	0.012	U
6-NS	0.004	U

As can be seen, EPSU is taken in a range of 0.004-0.03. But the well known value of EPSU is 0.003.

2.7.2 Reinforcement Properties

Reinforcement properties are defined by entering the parameters of the stress-strain curve shown in Figure 2.10.

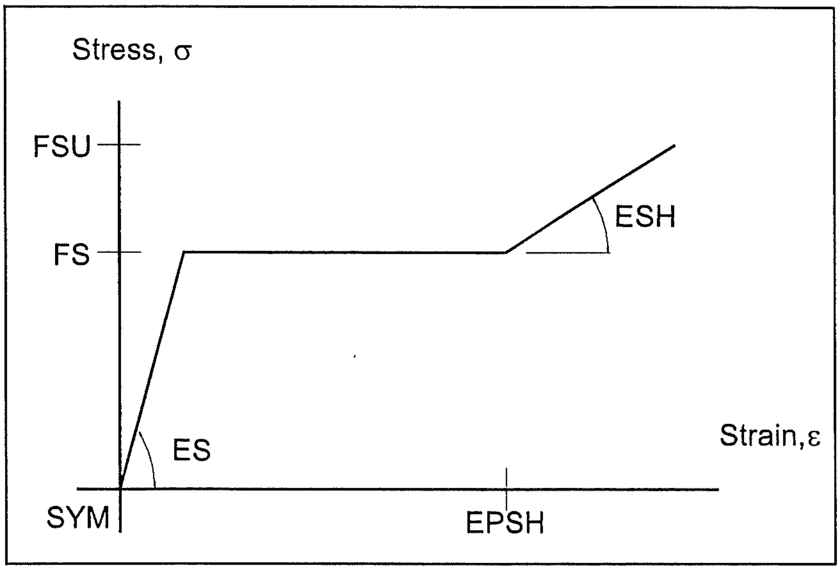

Figure 2.10 Stress-strain curve for steel used in IDARC

2.7.3 Column Parameters

IDARC permits to define only two types of steel layers as shown in Figure 2.11. The other column parameters such as dimensions, hoop bar parameters and the effectiveness of the hoop bar for different arrangements of stirrups are also given. All of these parameters are specified for the top and bottom cross-sections.

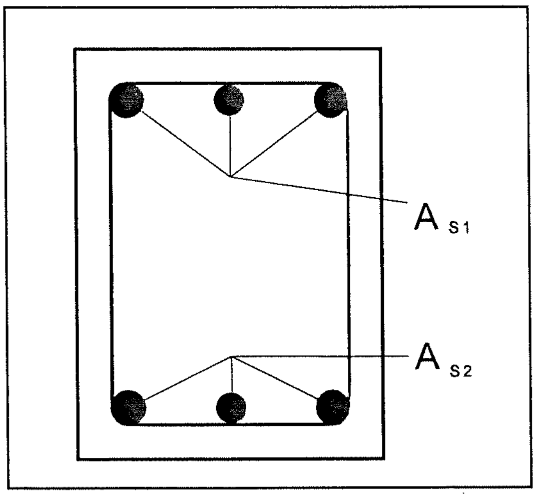

Figure 2.11 Vertical steel definition of column element in IDARC

2.7.4 Beam Parameters

The sectional properties of the first and second node of a beam element are entered. Hence if the steel areas are not the same throughout the element one can divide this beam element into several beam elements.

2.7.5 Shear Wall and Edge Column Parameters

Other than beam and column elements, steel parameters are entered as steel ratios in shear wall elements. Shear wall cross section can be divided into sub-regions to define the variations in steel ratios.

Edge columns are used to model the columns on the right and left side of the shear wall. As in shear walls, steel parameters are supplied as steel ratios for the edge column elements.

2.7.6 Mass

Masses are defined as nodal masses through out each storey. Instead of entering the mass values the user should enter the weight correspondence of these masses.

2.7.7 Dynamic Analysis

The ground acceleration data is given in a separate file and the parameters such as the time interval and number of data that will be considered are entered in the input file. This ground acceleration can be scaled to achieve specified peak ground acceleration. The ground acceleration between two data is assumed to be varying linearly (Figure 2.12)

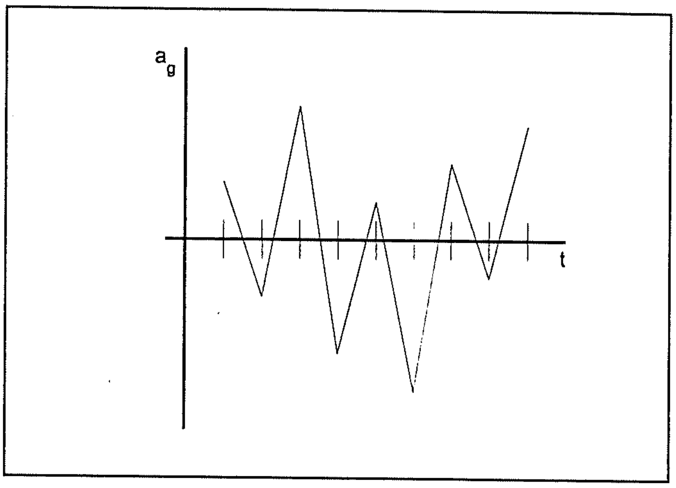

Figure 2.12 Linear change of ground acceleration

In the report it is noted that nonlinear analysis is sensitive to the choice of time step for response analysis. A value of 0.005 sec is suggested for typical buildings. It is suggested that a smaller value should be used if large amount of changes is expected in the stiffness of the elements. Also it is noted that improper choice for this value may yield numerical instabilities which will give large damage values (D>>3).

There are three options for the type of damping: mass proportional, stiffness proportional and Rayleigh damping. Damping matrix is expressed as

$$[\mathbf{C}]=\alpha_M[\mathbf{M}] + \alpha_K[\mathbf{K}]\,. \qquad (2.15)$$

The expressions for α_M, α_K are given in Table 2. ω_i is the circular frequency of mode *i* and ξ is the critical damping which is same for all modes.

Table 2.4 Parameters for damping matrices

Damping	α_M	α_K
Mass proportional	$2\xi\omega_i$	0
Stiffness proportional	0	$2\xi\omega_i$
Rayleigh	$(2\xi\omega_i\omega_j)/(\omega_i + \omega_j)$	$(2\xi)/(\omega_i + \omega_j)$

CHAPTER 3

ARTIFICIAL NEURAL NETWORKS AND APPLICATIONS TO DAMAGE ANALYSIS

3.1 General

A neural network can be defined as a simplified mathematical model of human nervous system, which has the ability of modifying itself according to past experiences to give reasonable answers for the problems that it never faced before. A more technical definition is given by Haykin [2]:

"A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:
1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store knowledge."

As can be understood from the above definitions, first a model of human nervous system unit, namely neuron, is formed, then using this primary unit a network of units that are interconnected are established. This network is then trained by special algorithms, called learning algorithms, using a pre-calculated or pre-observed set of well-analysed data. After these processes, the network is expected to give reasonable results for the problems that are not included in the set that is used to train the network.

In this chapter, a basis for the neural networks will be given first. Then well known neural network architectures will be explained. Back-propagation algorithm, which is the most popular algorithm in structural engineering research field, will be explained in detail. Matrix representation of this algorithm will be given. And finally another network architecture will be introduced.

3.2 Basic Unit: Neuron

Consider the equation defined by,

$$ax + by + \theta = 0 \tag{3.1}$$

The line and the regions defined by this equation will be as in Figure 3.1

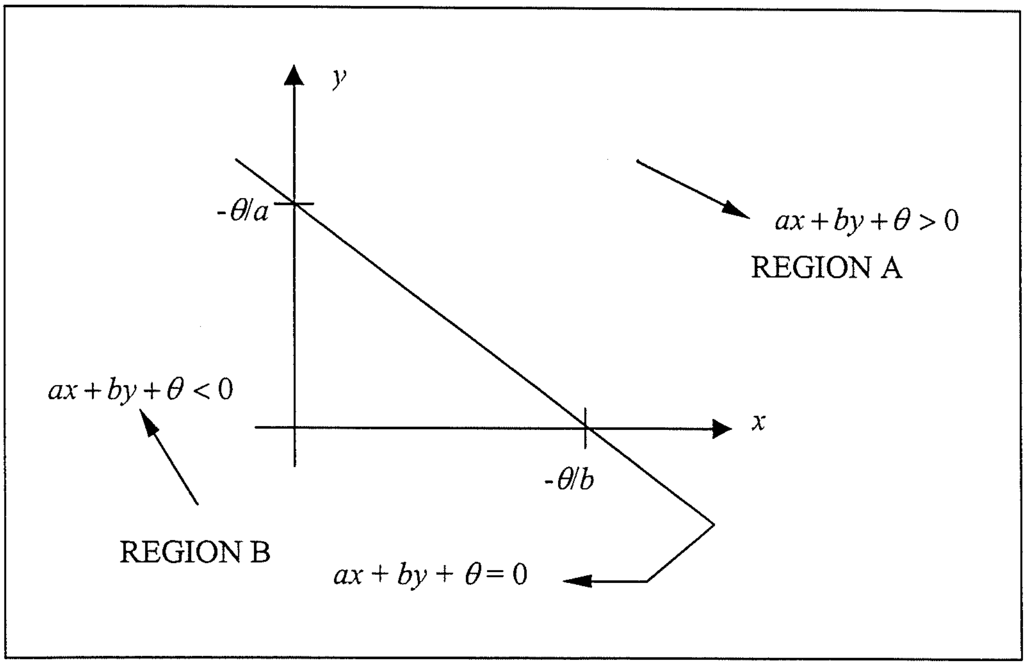

Figure 3.1 Line defined by $ax + by + \theta = 0$

Now consider that we have some (x,y) pairs at hand, assuming that none of them coincides with the line defined by $ax + by + \theta = 0$, we want to determine the region that these pairs belong to. A function defined by the Figure 3.2 can readily do this job.

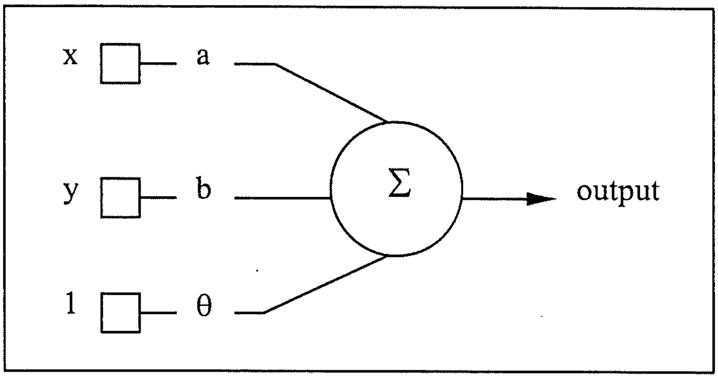

Figure 3.2 A simple model for the $ax + by + \theta = 0$ equation

In Figure 3.2 a, b, θ are known values of the equation. The function represented by Σ is defined as:

$$\sum = \begin{cases} +1, & t \leq 0 \\ -1, & t > 0 \end{cases} \tag{3.2}$$

The model works as follows:

1. Multiply the inputs with the corresponding weights: ax, by, 1θ,
2. Sum all the values obtained in Step 1: $ax + by + \theta$
3. Evaluate the function $f(t)$: $\mathrm{f}(ax + by + \theta)$
4. If he result turns out to be +1 this pair is in region A, otherwise it is in B.

Now consider the reverse problem: If the pairs and the regions they belong to are given, will it be possible to find the parameters that define the equation $ax + by + \theta = 0$ namely a, b, θ? Consider Figure 3.3 for the answer of this problem.

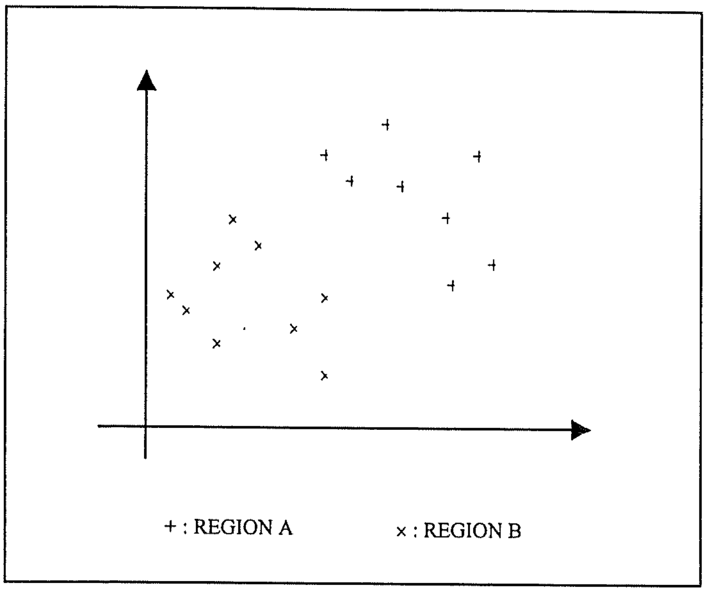

Figure 3.3 Reverse problem of region separation

The question is simple to understand: Can we separate these two types of data by drawing a linear line? It is easy to see that the answer for this problem is not unique. One can draw several lines those separates two types of data with a ruler as shown in Figure 3.4. Every line will have different values of a, b, θ.

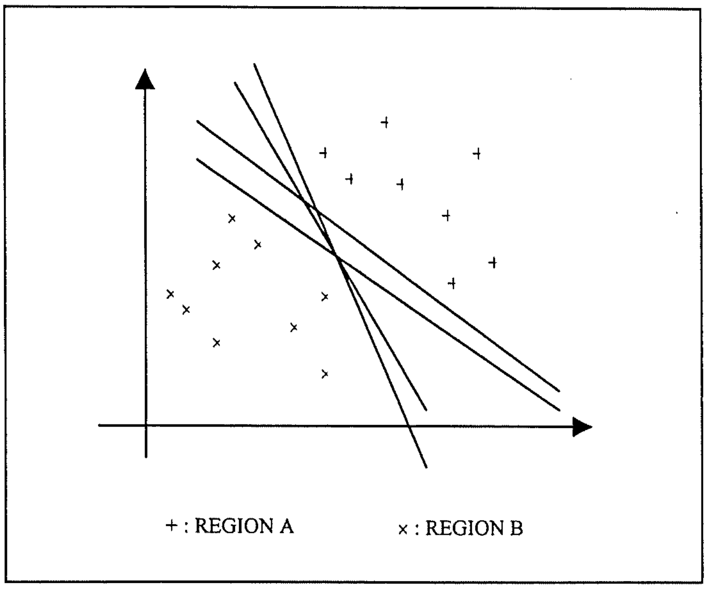

Figure 3.4 Some of the solutions for the separation problem

This problem is one of the basic problems which artificial neural networks deal with. The mathematical model defined by Figure 3.2 is the basic unit of artificial neural networks, and is called neuron. The parameters *x, y* are the inputs for this neuron. Actually the number of inputs need not to be two for a neuron which is dependent on the type of the problem. The parameters *a, b* are called synaptic weights. θ is called bias, which will be explained later.

In this model inputs are multiplied with the corresponding weights to be collected in a pool. Every input will effect the value of the pool as much as its weight. If its weight is small, its effect on the pool will be small and vice versa. This is why the parameters a, b are called as weights.

Let us examine the function $f(t)$ shown in figure 3.5a. In neural networks, the function that takes inputs multiplied with weights as parameter and fires the output of the neuron is called activation function. In our specific example the activation function is a threshold function, which fires 1 when the input exceeds zero. In neural networks several activation functions are used for different types of problems. A list of activation functions is presented in Figure 3.6.

At first glance the problem introduced above may seem very simple. After giving a brief explanation how neural networks deal with this simple problem, which is the general approach of neural networks, more difficult problems that can not be solved with a ruler will be implemented. To clarify the problem, let us give some numerical values as shown in Table 3.1(See also Figure 3.6).

Table 3.1 Numerical data for the separation problem

x	y	Region
1	1	B
1	2	B
2	1	B
3	2	A
3	3	A
4	2	A
4	3	A

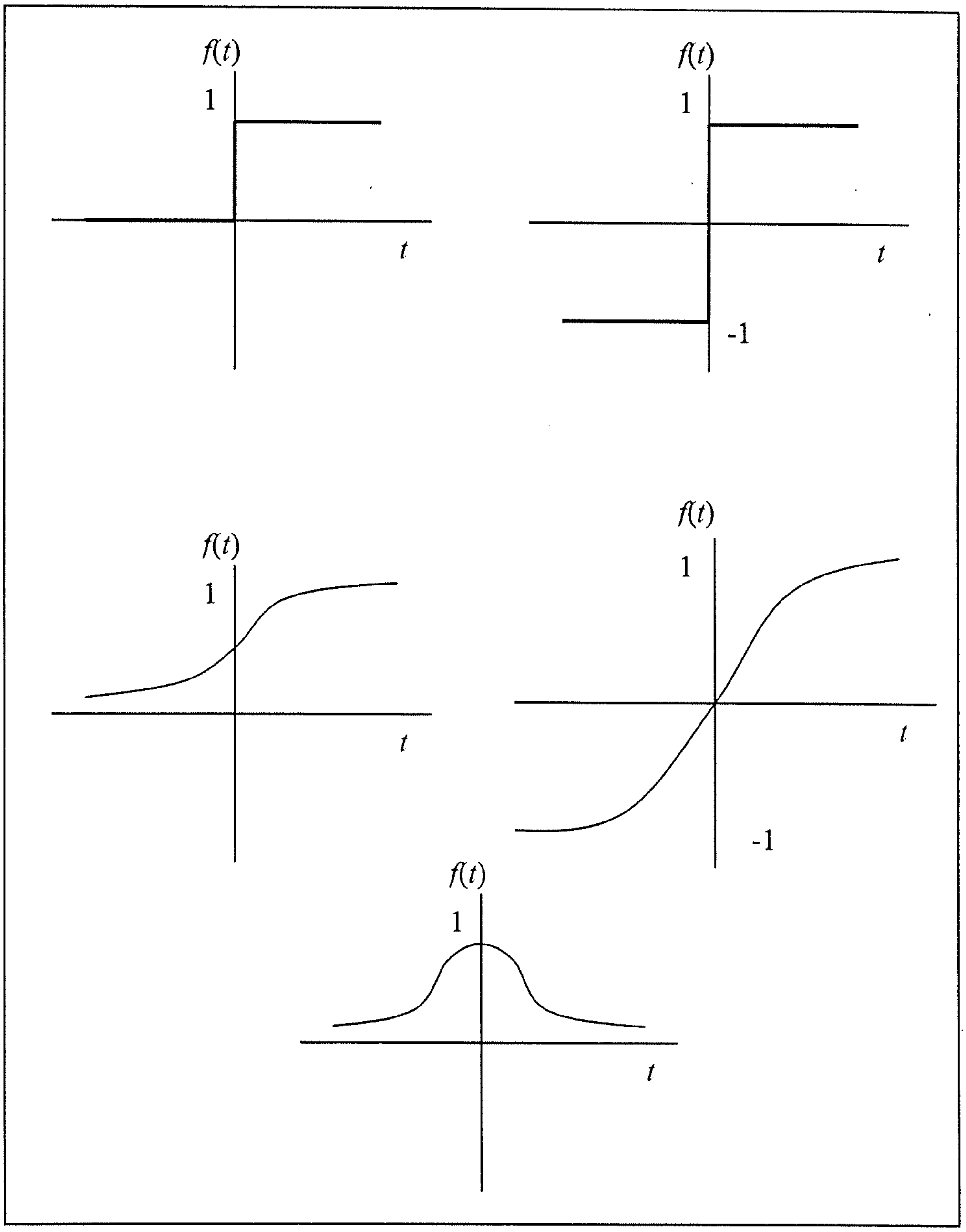

Figure 3.5 Activation functions a) bipolar output threshold function, b)binary output threshold function, c) sigmoid function, d) tangential sigmoid function, e) Gaussian function

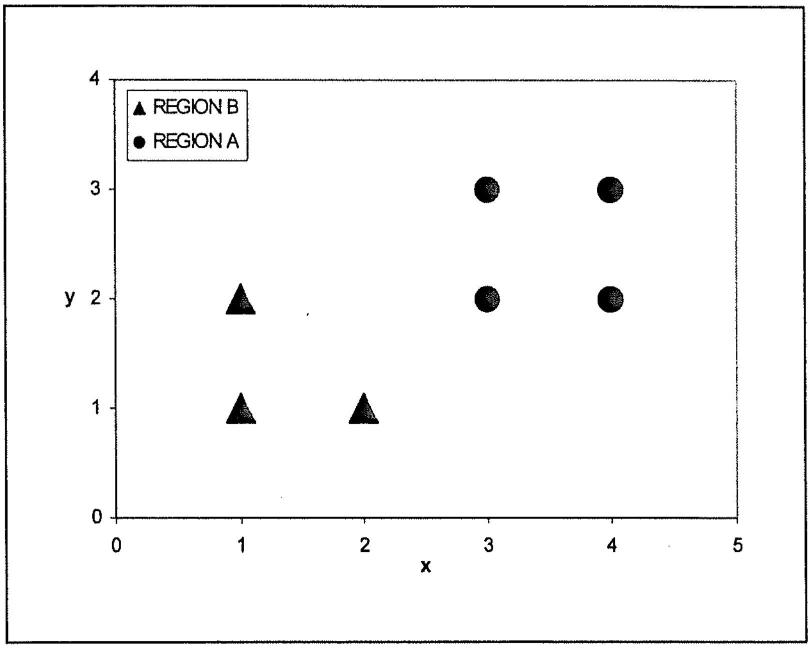

Figure 3.6 Graphical representation of the numerical problem

Now we have the input pairs and their regions at hand and a neuron which we are expecting to implement this problem. The unknowns are *a, b*, and *θ*. The procedure will be as follows: First we will select random values, which will be in the range of (–1,1), for *a, b*, and *θ*. We will use the binary output activation function. If the input results a value of 1, we will say that it belongs to one of the regions, say Region A, and if it comes out to be 0, we will say that it belongs to Region B. Hence the input output set for our simple neuron shown by Figure 3.2 will be as given in Table 3.2:

Table 3.2 Input-output set for the separation problem

x	Y	Output
1	1	0
1	2	0
2	1	0
3	2	1
3	3	1
4	2	1
4	3	1

Of course for the random values of a, b, and θ, our simple neuron will not give the expected outputs. Hence our initial random weights should be updated so that the neuron will give the desired outputs. The algorithm that updates the weights is based on a gradient-descent error minimisation process. Let us call the expected output of the neuron as o, and the neuron output as o'. Hence the error of the neuron will be:

$$e = (o - o') \tag{3.3}$$

But the neuron output $o' = ax + by + \theta$, hence the error will be a function of a, b, and θ for the known values of x, y and o. Hence we are searching for the values of a, b, and θ that minimise the error e. The minimum error can be reached by taking the derivative of the error and using the derivative to reach a local or a global minimum of error (Figure 3.7). Every weight will be updated using this derivative.

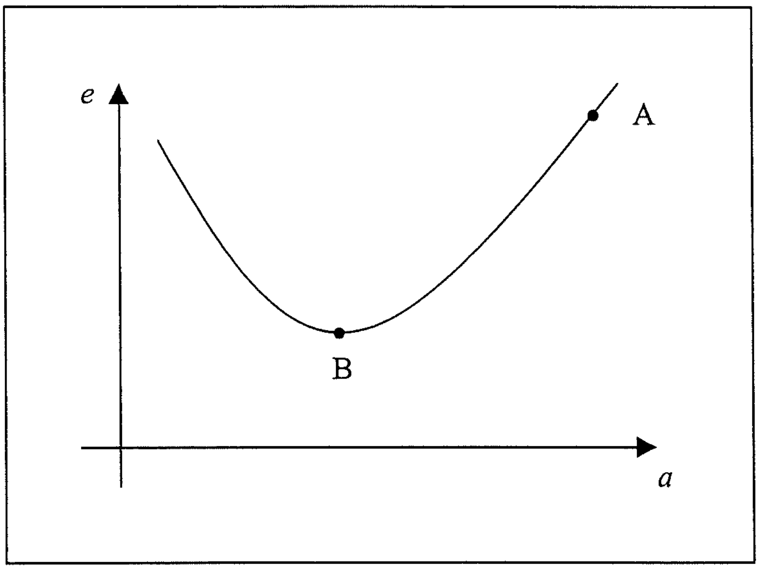

Figure 3.7 The graph of error function of the single neuron

In Figure 3.7, point A represents the error due to the initial random values of weights. The algorithm will start at this point and will reach a neighbourhood of point B, which is a minimum. During this algorithm all of inputs will be introduced one by one and the corresponding error and derivative will be estimated. As can be seen from the graph even at the end of the algorithm there will retain an error. Hence the algorithm guarantees reaching a minimum but does not guarantees that the error will be zero. This discussion brings the problem of local and global minimum, which will be explained, later in this chapter.

Our algorithm will find a set of weights namely a, b and θ, which separates these two regions. The solution of this problem will not be unique. Hence the algorithm will yield a different set of weights for different sets of initial random weights.

Now consider a much more complex problem shown in Figure 3.8. As can be seen separating these two types of data by drawing a linear line is impossible. For this kind of problem, which is not linearly separable, several interconnected units are used. Algorithm for this kind of classification problem will be given in the following sections.

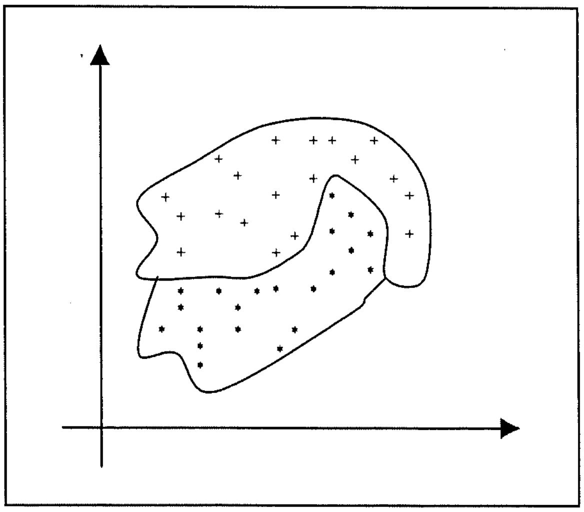

Figure 3.8 A more complex classification problem

Figure 3.9 shows the more general form of a unit. As can be seen the number of inputs is not two and the input for bias can be –1. These units are the building units of multilayer networks.

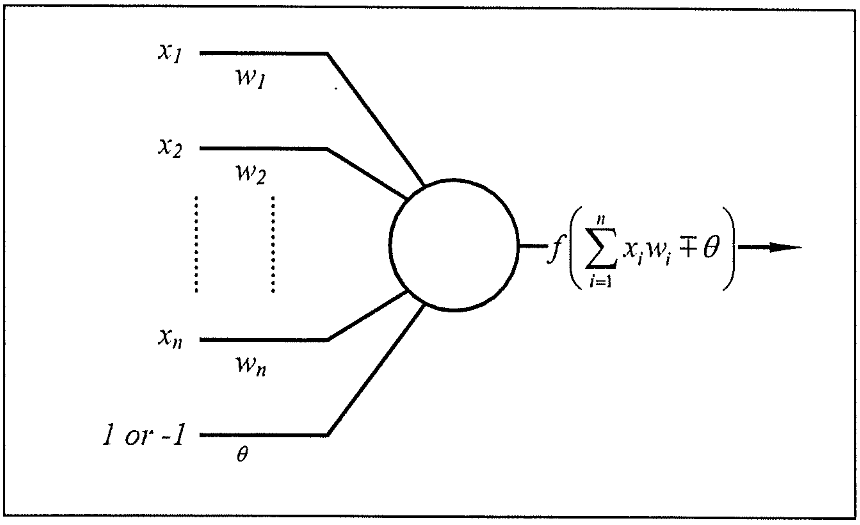

Figure 3.9 General form of a neuron

3.3 Multilayer Layer Neural Networks

Multilayer neural networks implement more complex problems that can not be solved with a single unit. These networks are formed by layers of units as shown in Figure 3.10. The first layer is called the input layer, which includes only the inputs. In the previous examples the number of inputs was two, but in this case number of inputs is *n*. Layer that gives the output is called the output layer. Different than a simple neuron, the number of outputs should not need to be one. Let us call the number of the outputs as *m*. Layers that lie between the input and the output layer are called the hidden layers. The number of the units in these layers is determined according to the type of the problem. The layers are connected by the weights. As in the previous example, the inputs are introduced to the units of the first layer. The outputs of this layer will be inputs for the second layer and the same computations will be applied for the third layer. This process is called as forward propagation of the input data.

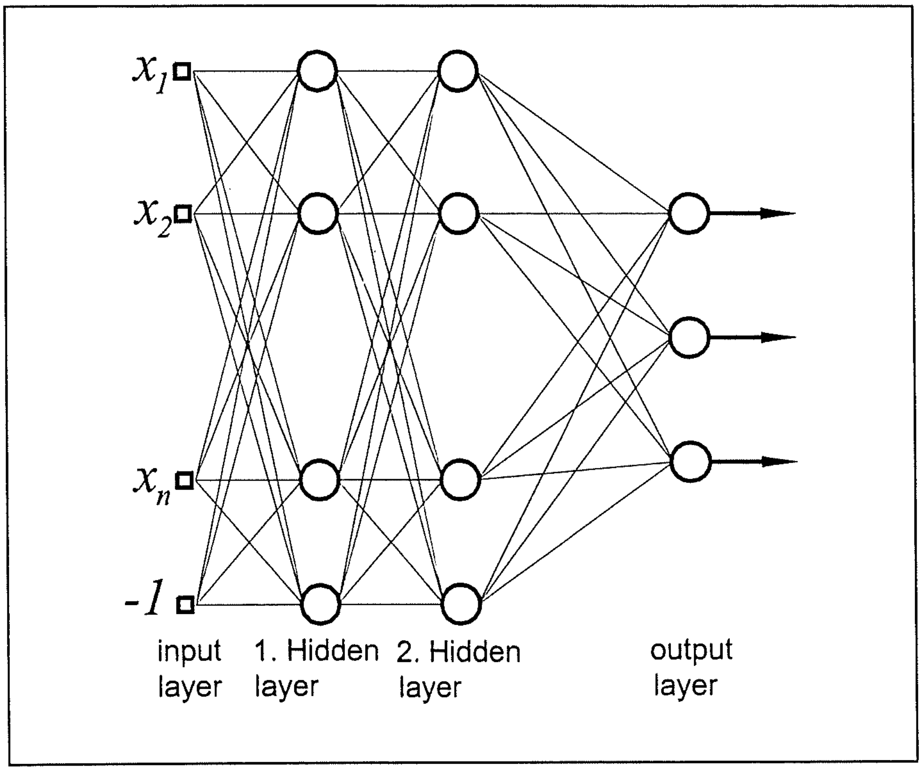

Figure 3.10 Multilayer network with two hidden layers

The complex problem explained in the previous part can be implemented easily by a multilayer network with one hidden layer. The input layer will have three elements that corresponds to (x, y) pairs of the training set and –1 or 1 for the bias input. The last element of the hidden layers should be –1 or 1 also. Output layer will have two elements. Actually it can have one element for our specific problem as done in the previous part, but to give a different architecture different than the one-neuron example; output layer with two elements will be used. The sigmoid activation function will be employed for all units. The procedure will be as follows: First, random weights will be generated. Second an input pair (x,y) and its region will be selected randomly. We are expecting to obtain an output of (1,0) for the pairs that belong to Region A, and (0,1) for Region B. Inputs will be introduced to the network and two outputs will be estimated. But the output of the network will not give the correct result at the first forward propagation since it used a random set of weights. Hence if the output of the network is (s,t), the error will be $(1-s,-t)$ if the introduced pair belongs to Region A and $(-s,1-t)$ if it belongs to region B. Then the weights are updated by an algorithm called as back-propagation in which the derivative of the error function is used. This algorithm will be explained in the next part. This process is repeated for the other input pairs until a permissible error is reached.

Multilayer networks and back-propagation training has been proved to be a good classifier for more than two patterns. The ability of implementing pattern classification problems of these networks comes from their massively parallel-distributed structure and the nonlinear behaviour of the units and hence of the whole system.

A multilayer network can be visualised as a nonlinear mapping device that maps n-dimensional input space to m-dimensional output space. Hence it can be used not only for classification problems but also for n-dimensional interpolation and extrapolation problems. For example let us consider a nonlinear function, $f(x)$, with $(x, f(x))$ pairs given in Table3.3 (see also Figure 3.11).

Table 3 Input-output set for the interpolation problem

x	$f(x)$
0	8
1	–1
2	–22
3	–49
4	–52
5	23

Several methods can be utilised to make a nonlinear interpolation of this problem. A multilayer neural network will do the same job. For example it will implement the above data as in Figure 3.12.

The shape of the interpolation is highly dependent on the parameters of the neural network such as the number of hidden layers, number of neurons, activation function used etc. These parameters affect the performance of the system for all kind of neural networks. The best model should be determined after testing of several different models.

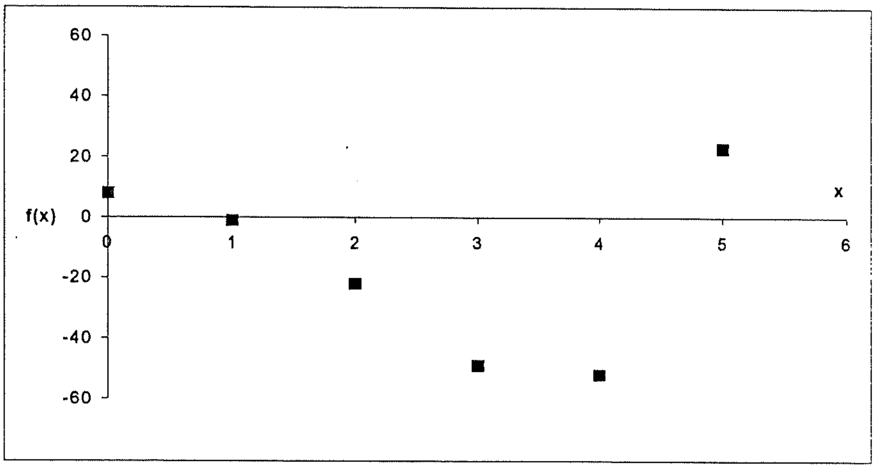

Figure 3.11 Graph of input-output set of the interpolation problem

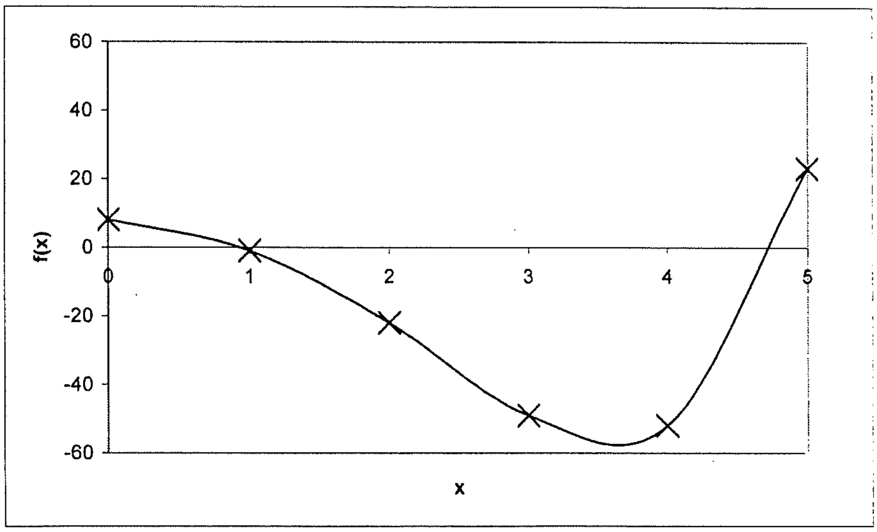

Figure 3.12 Interpolation done by a multilayer neural network

3.3.1 Back-propagation Algorithm

In this part the notation and representation of basic back-propagation algorithm will be given first. Second the matrix representation will be introduced. Finally some remarks will be given for the performance of the network.

3.3.1.1 Notation

Since the algorithm applies to all layers of the multilayer networks, the notation will be shown between two layers only.

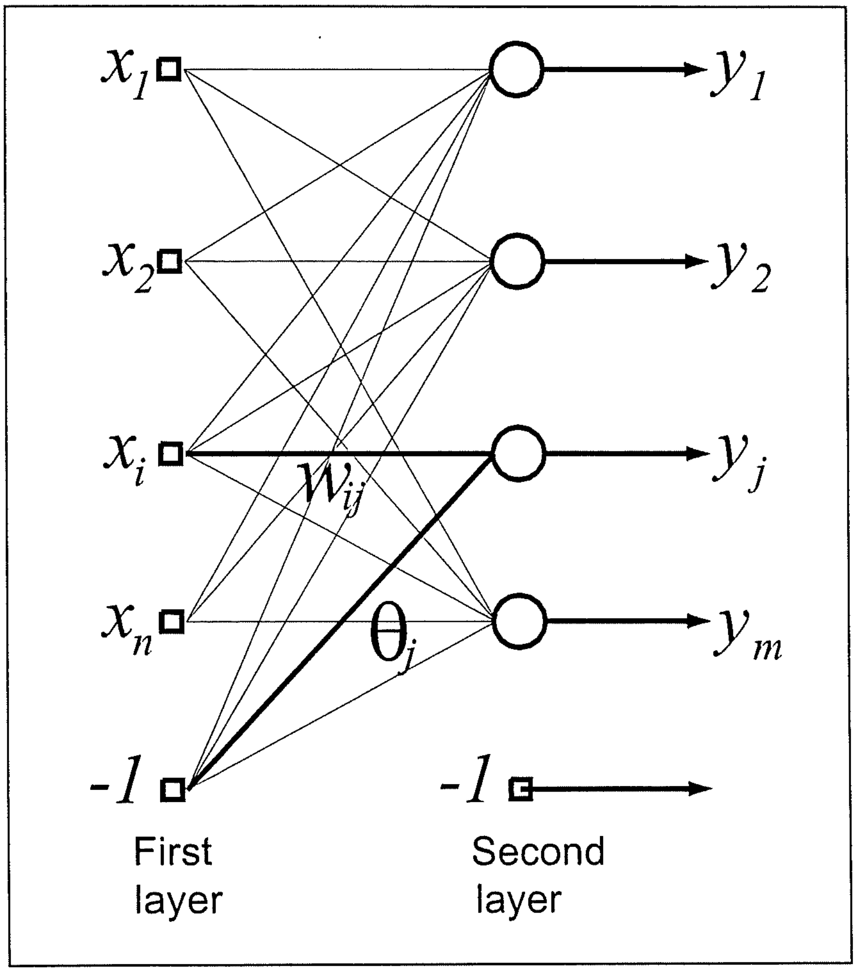

Figure 3.13 Notation for any two layers of the multilayer networks

1. The first layer can be the input layer or an output of the previous hidden layer.
2. The second layer can be the output layer or an input for the next layer.
3. $A(t)$ denotes the value of A at iteration t
4. L is the number of the layers including the input and output layer.
5. Weight $w_{ij}(t)$ is the synaptic weight connecting the output of the neuron i in the first layer to the input of neuron j in the second layer.

6. $\Delta w_{ij}(t)$ is the weight correction applied to weight $w_{ij}(t)$.

7. The expected output of the j'th term of the output layer is d_j.

8. The i'th element of the input vector that is randomly selected from the training set is denoted by $x_i(t)$.

9. The neural network output is given as $y_i(t)$.

10. The sum of the squared errors is denoted by $\mathcal{E}(t)$.

11. Learning rate is denoted by η.

12. Momentum term is given as α.

13. δ_i^k is the error vector for the k'th layer

3.3.1.2 Algorithm

Since the partial derivative of the error function with respect to weights will be computed, continuous differentiable activation functions should be used. Hence back-propagation algorithm can not be applied to networks with binary and bipolar output threshold activation functions. Actually for these type of networks several training algorithms have been developed such as the perceptron learning rule. Algorithm will be given for the sigmoid activation function. One can easily extend the algorithm for other differentiable functions keeping in mind that different type of activation functions are used for different type of problems. The sigmoid function is given by,

$$f(x) = \frac{1}{1+e^{-ax}} \tag{3.4}$$

where a is a constant. The graph of this function is given for a = 1,2,3 in Figure 3.13. As can be seen when a tends to infinity, sigmoid function approaches to behave as a binary output threshold activation function. Derivative of sigmoid function will be

$$\frac{\partial f}{\partial x} = \frac{ae^{-ax}}{1+e^{-ax}} = f(x)[1-f(x)] \tag{3.5}$$

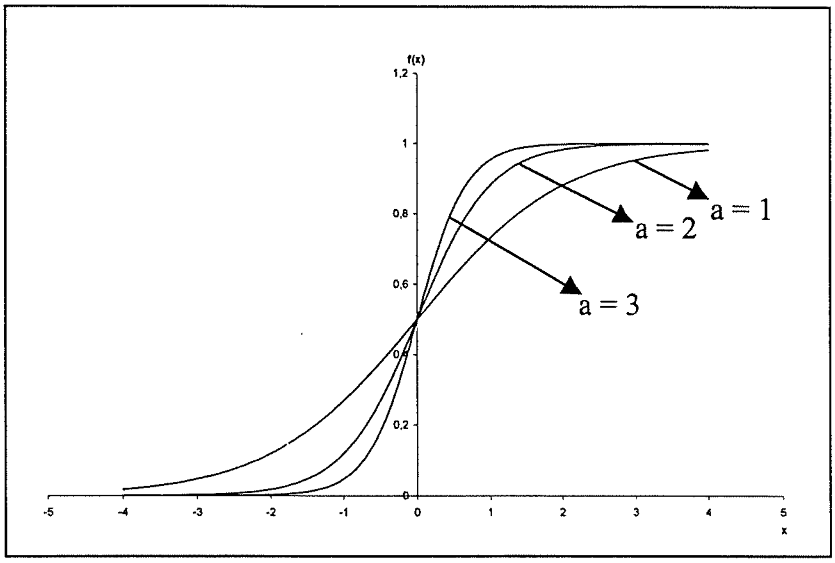

Figure 3.14 Sigmoid activation function

One iteration of the algorithm has basically two phases: forward propagation and back-propagation.

Forward propagation:

1. Choose an input-output vector pair from the training set. This pair can be selected randomly or in a specific range.
2. Generate random values for the weights. The range for the random weights can change from [–0.1,0.1] to [–1.0,1.0]. It is better to determine the range after some trials.
3. Introduce the input vector x_i to the network through the input layer and obtain the output of the network, y_j. This output will be different from the expected output d_j. Hence the error function will be

$$\mathcal{E}(t) = \tfrac{1}{2} \Sigma(d_j - y_j)^2 \qquad (3.6)$$

Back-propagation:

1. Calculate the error vector of the output layer:

$$\delta_j^L(t) = y_j(t)\,[\, 1 - y_i(t)\,]\,[\, d_j(t) - y_j(t)\,] \qquad (3.7)$$

2. Calculate the weight correction for the weights between the (L–1)th layer and the output layer(According to Figure 3.13 these are first and second layer respectively).

$$\Delta w_{ij}(t) = \eta\ \delta_j^L(t)\ x_i(t) \tag{3.8}$$

3. Update the weights between the (L–1)th layer and the output layer

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \tag{3.9}$$

4. Calculate the error vector of the (L–1)th layer:

$$\delta_j^{L-1}(t) = x_i(t)\ [\ 1 - x_i(t)\ ]\ w_{ij}\ \delta_j^L(t) \tag{3.10}$$

here the weights, w_{ij} are the weights between the (L–1)th layer and the output layer.

5. Update the weights between (L–2)th layer and (L–1)th layer as done in step 2 and step 3 until the first and second layer correspond to input and second layer of the network.

Forward propagation and back-propagation are done until a permissible error is reached. Learning rate in step 2 is an important parameter that effects the performance of the network. The learning can gain speed with a term called momentum term. In this case the weight update will be as follows:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) + \alpha\ \Delta w_{ij}(t-1) \tag{3.11}$$

Discussion on learning rate and momentum term will be given later. Now the matrix form the algorithm will be given to visualise the procedure better. An L1–L2–L3–L4 network will be used. Here L1 is the input vector size including the bias input, –1 and L4 is the output vector size. L2 and L3 are the number of neurons in the second and third layer, respectively including bias inputs. The architecture will be in Figure 3.14

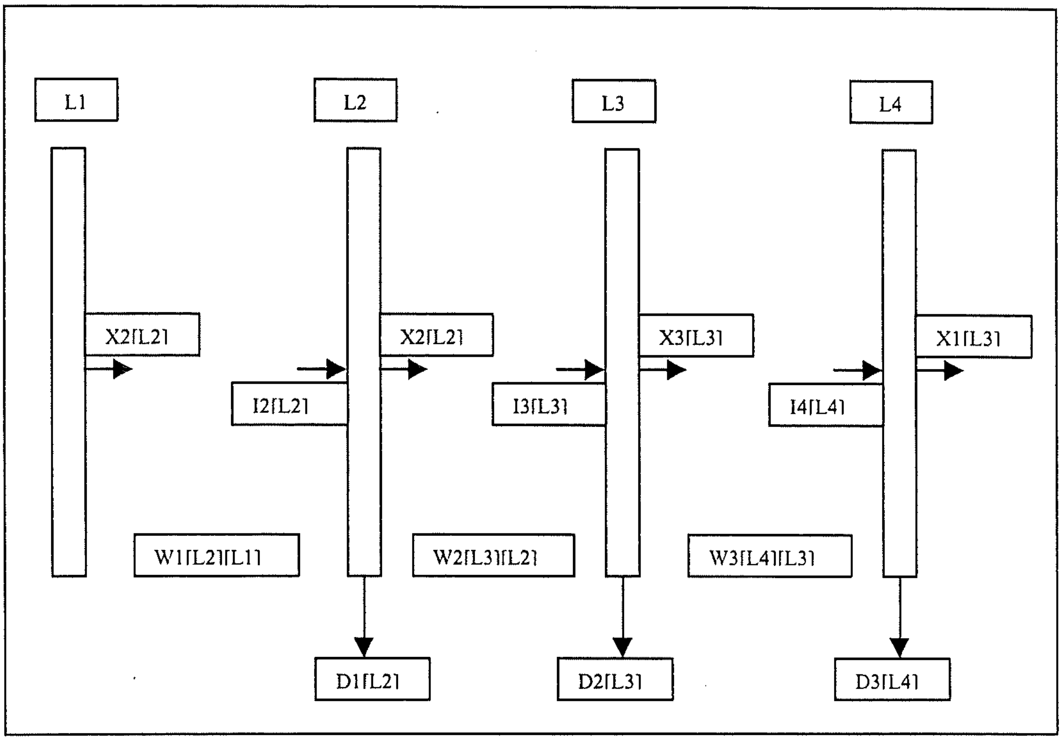

Figure 3.15 Notation for multilayer networks for matrix representation

In Figure 3.15 values in brackets show the dimension of the vectors and matrices.

Forward propagation will proceed as follows:

$$\begin{aligned} &\underline{I2} = \underline{W1}\text{ x }\underline{X1}, \qquad \underline{X2} = f(\underline{I2}) \\ &\underline{I3} = \underline{W2}\text{ x }\underline{X2}, \qquad \underline{X3} = f(\underline{I3}) \\ &\underline{I4} = \underline{W3}\text{ x }\underline{X3}, \qquad \underline{X4} = f(\underline{I4}) \end{aligned} \tag{3.12}$$

Back-propagation will be

$$[D3]_{L4\times 1} = [X4]_{L4\times 1} \bullet [1 - X4]_{L4\times 1} \bullet [D - X4]_{L4\times 1} \tag{3.13}$$

$$[dW3]_{L4\times L3} = [D3]_{L4\times 1} \times [X3]_{1\times L3} \tag{3.14}$$

$$[D2]_{L3\times 1} = [X3]_{L3\times 1} \bullet [1 - X3]_{L3\times 1} \bullet \left[[W3]^{T}_{L3\times L4} \times [D3]_{L4\times 1} \right] \tag{3.15}$$

$$[dW2]_{L3\times L2} = [D2]_{L3\times 1} \times [X2]_{1\times L2} \tag{3.16}$$

$$[D1]_{L2\times 1} = [X2]_{L2\times 1} \bullet [1 - X2]_{L2\times 1} \bullet \left[[W2]^{T}_{L2\times L3} \times [D2]_{L3\times 1} \right] \tag{3.17}$$

$$[dW1]_{L2\times L1} = [D1]_{L2\times 1} \times [X1]_{1\times L1} \tag{3.18}$$

In this notation • (dot) means the dot product of the two vectors, and x means the matrix multiplication. It should be noted that the last elements of the layers other than the output layer are not neurons. They have the value -1.

3.3.1.3 Performance of the Algorithm

Back-propagation algorithm is a gradient descent algorithm. It starts from a point on the error surface and it goes downwards on this surface. When it reaches a minimum, it will stick to this minimum. Hence if the permissible error is smaller then this minimum, algorithm will not converge. This should be checked throughout the iterations.

The minimum value of the error that the algorithm reaches may not be the global minimum. This is an expected problem since the algorithm is a gradient descent algorithm. To eliminate this problem the learning rate should be changed adaptively. The general approach for this is as follows:

1. Every weight should have its own learning rate parameter.
2. When the derivative of the error function has the same sign for several iterations, the learning rate should be increased.

3. When the derivative of the error function do not follow a regular sign change, the learning rate should be decreased.

Small learning rate means small steps on the error surface, and vice versa. Other methods such as stochastic algorithms with simulated annealing, Boltzman machine are used to find the global minimum, which are more powerful but more sophisticated.

The momentum term behaves somewhat like an adaptive learning rate since it accounts for the previous weight update. But the momentum constant should be selected after some experimental runs as in learning rate case. The speed of the back-propagation can be increased by simple modifications. Some algorithms obtained after these modifications are delta-bar-delta rule, Rprop, Qprop, QRprop etc. Fuzzy control of learning rate is another method to accelerate the algorithm.

It should be noted that performance of the algorithm is just a parameter for the performance of the network. Performance of the network depends on several other parameters.

3.3.2 Performance of the Multilayer Networks

It is expected from the network to give reasonable results for the problems that it has never faced before, i.e. the problems that do not exist in the training set. For example, the interpolation of the training set given in Table 3.1 may be as in Figure 3.16. As can be seen easily the nonlinearity given by the network is very high. The reason of this nonlinearity is related with the number of hidden neurons. It is observed from the past experience that the nonlinearity increases with the number of hidden neurons.

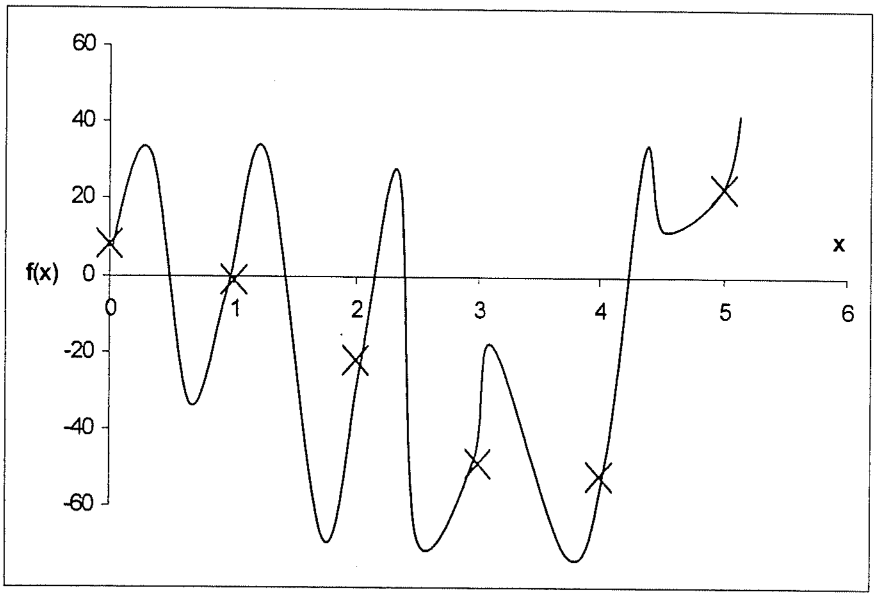

Figure 3.16 Another nonlinear approximation done by a network

The extrapolation is another measure for the performance of the network. It is expected to have reasonable output for the input that is not in the range of training set.

The above problems are called generalisation problems. Figure 3.17 gives a mathematical representation for the generalisation problem. In this figure X represents the input space for the physical system. R is a subspace in which the input and the output variables obey a set of relations. From this set one can select two subspaces namely learning (training) and testing set abbreviated as L and R, respectively. We use the training set to train the network hence the testing set is not introduced to the network. The performance of the network will be measured by testing the network with the testing set.

The most important parameters that effect the generalisation of the networks are first the size and the efficiency of the training set, second the architecture of the network, third the learning algorithm used, and finally the complexity of the physical problem.

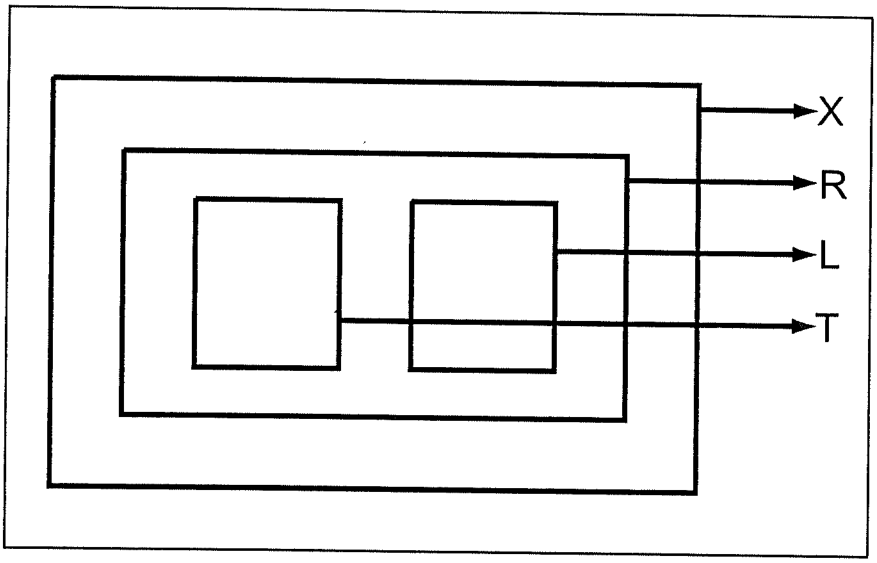

Figure 3.17 Subspaces of the problem input space

The training set should include as different ranges of input subspace, R as possible. But if the learning set is fixed other parameters determine the performance of the network. It is advised that all of these parameters should be tested to obtain better performance. This is why the caution that not every network will give good performance for every kind of problem is very important.

3.4 Unsupervised Learning and Kohonen Network

In back-propagation algorithm we used a predefined input-output set namely training set to train the network. The algorithm compares the actual output and the network output to update the weights. Hence the training occurs in control of a teacher: output. The network output is forced to be the actual output. This type of learning is called supervised learning. In unsupervised learning the characteristics of the input data are determined without using such a teacher. In this type of learning the network output is not forced to be the actual output. The network organises itself according the characteristics of the input data. In Kohonen network, which is one of the unsupervised algorithms, self-organisation is observed on the two-dimensional output layer called Kohonen layer.

A simple Kohonen network consist of an input layer and a two-dimensional output layer as shown in Figure 3.18

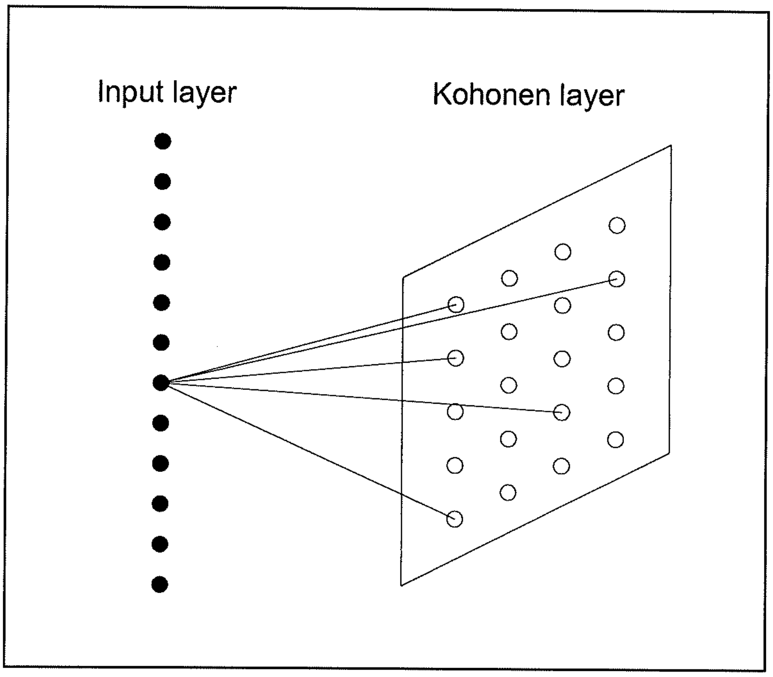

Figure 3.18 Kohonen network

The principal of the Kohonen network is to determine a winning neuron on the output layer when an input pattern is introduced to the network. The criterion to determine the winning neuron may be Euclidean distance between the input vector and the weight vector coming to that neuron. Euclidean distance between any two vectors ***x*** and ***y*** is defined as

$$d_E = \left[\sum (x_i - y_i)^2\right]^{1/2} \qquad (3.19)$$

The winning neuron in the output layer will be the one with minimum Euclidean distance. The physical meaning of this criterion can be visualised by the dot product of the input and weight vector. If the Euclidean distance is smaller, the dot product of the two vectors will be larger as shown in Figure 3.19.

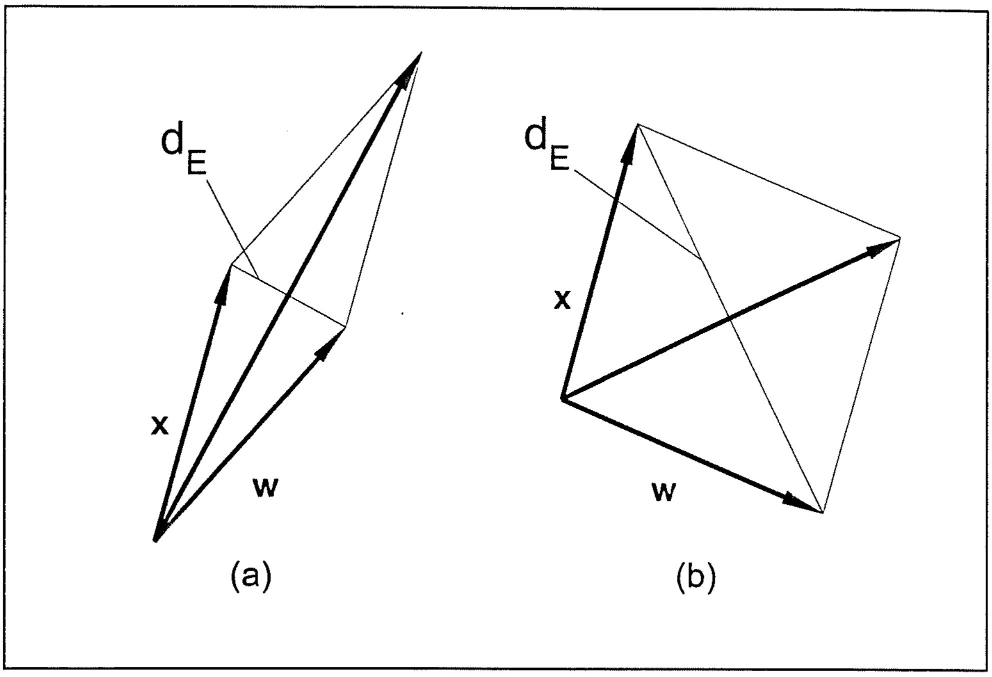

Figure 3.19 Euclidean distance between two vectors

After selecting the winning neuron, the weights coming to this neuron are updated as to increase the dot product of the input and the weight vectors. The update rule is given as follows:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta[\mathbf{x} - \mathbf{w}(t)] \tag{3.20}$$

In practice this update is not done not only for the winning neuron but also for a specified neighbourhood of that neuron as shown in Figure 3.20. The boundaries of this neighbourhood will shrink with iteration until it only includes the winning neuron.

Different than back-propagation algorithm, the number of iterations is not decided by a permissible error since there is no term of error. The system is self-organising. After the training process, every input pattern will map an output neuron in the output layer. This final vision of mapping is called future map. The system is then ready to map any other pattern that was not used to form the future map.

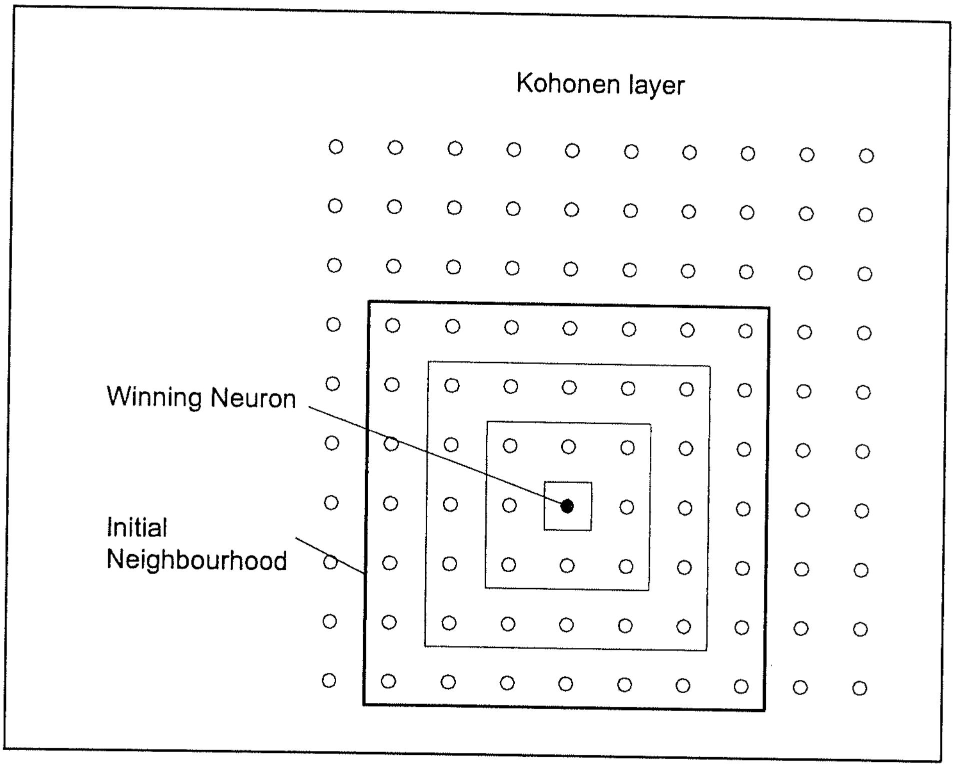

Figure 3.20 Square neighbourhoods of the winning neuron

3.5 Closing Remarks

Although back-propagation is the most popular algorithm for learning several other algorithms that show better performance are developed for multilayer networks such as CMAC. Pure back propagation algorithm may not be efficient for most problems. Alternate algorithms or modifications of back-propagation may work well for this type of problems.

CHAPTER 4

CASE STUDIES SOLVED WITH IDARC

4.1 General

As the first phase of this study, three frames and one single-degree-of freedom system are analysed with IDARC to observe the effect of different characteristics of structural systems and earthquakes on the structural damage. In this chapter, a frame with shear walls was analysed with IDARC (FR1). Due to some problems faced with IDARC, a simpler wire-frame, which is designed according to TS-500, EUROCODE2 and ACI is analysed (FR2). Then a frame (FR3) similar to FR2 is analysed for the next part of the study. To check some numerical errors occurred in IDARC a single degree of system is analysed also (Table 4.1).

Table 4.1 Case studies

NAME	DESIGN	SHEAR WALL
FR1	TS-500	EXISTS
SDOF	NOT DESIGNED	-
FR2	TS-500, EUROCODE-2 ACI	NO
FR3	NOT DESIGNED	NO

4.2 Case 1: FR1

The frame FR1 is a four-storey, four bay frame as shown in Figure 4.2. This frame is one of the frames of a 3D structure (Figure4.1). This structure is analysed with SAP90 and designed according to TS-500 and one of the frames is isolated for analyses.

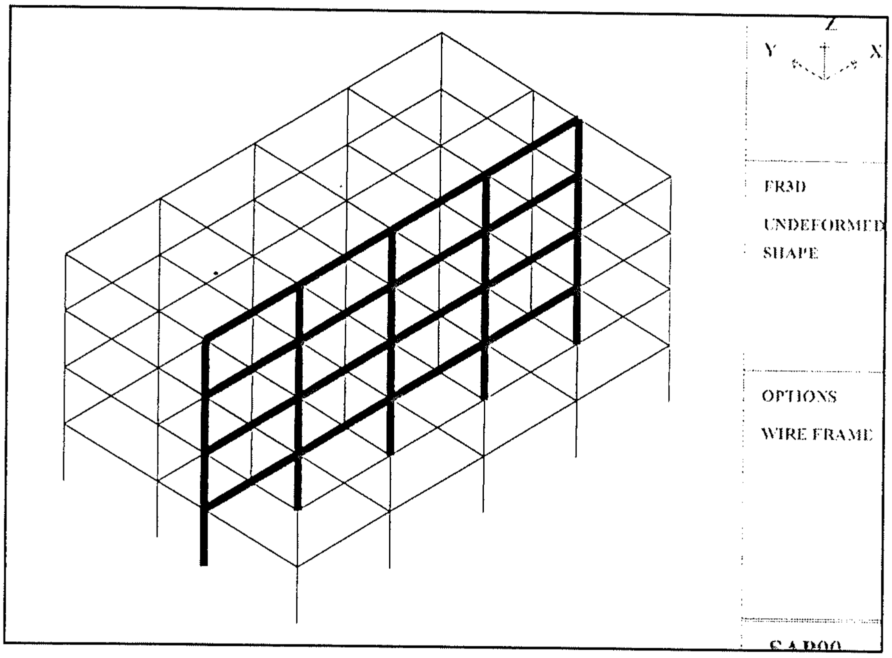

Figure 4.1 3D building employed for FR1

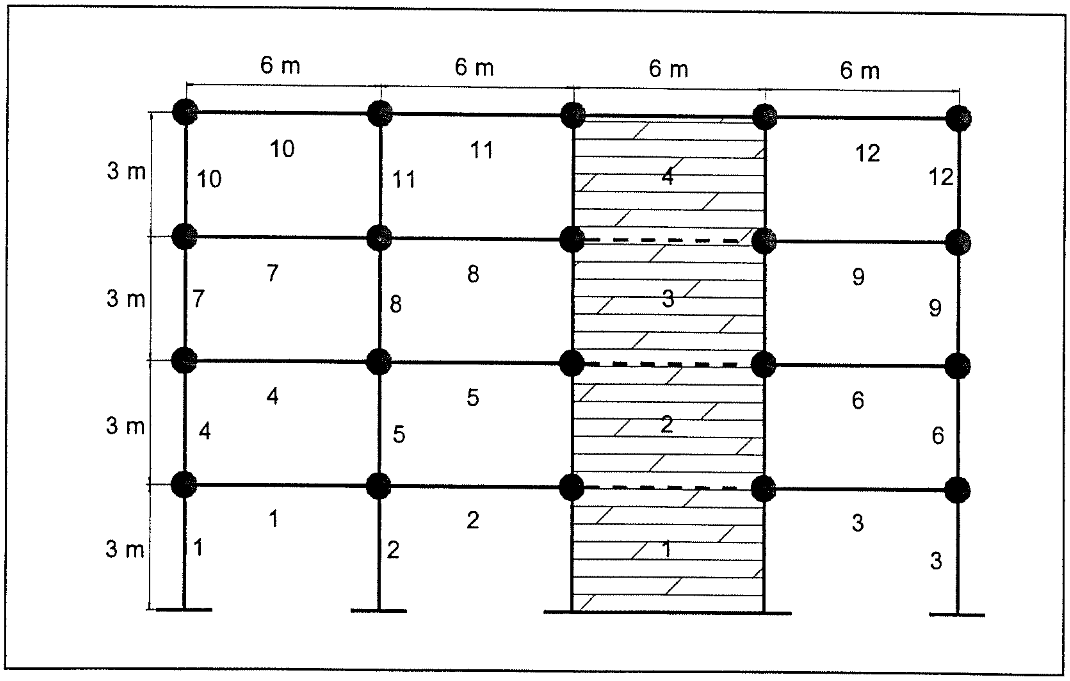

Figure 4.2 General view of FR1

Dimensions of elements of FR1 are given in Table 4.2.

Table 4.2 Element properties of FR1

Column dimensions (mm)	400x400
Beam dimensions (mm)	500x300
Shear wall dimensions (mm)	5600x200 and 5600x300
Edge column dimensions (mm)	400x400

4.2.1 Columns

All column elements used in FR1 are the same. Typical column is 400x400mm and 3m in height with 250mm rigid zones at top and bottom, except that the columns in the first storey are 2750mm in height and they have no rigid zone at bottom. The initial axial loads on the columns are given in Table 4.3. Hoop bar diameter is 14mm with spacing of 100mm. The effectiveness of hoop bar is taken as 0.66. The area of vertical steel on one face is 942.5mm^2. Hence the steel ratio is 0.0118.

4.2.2 Beams

Every parameter of the beam elements used in FR1 is the same except the vertical steel areas. The top and bottom steel areas are given in Table 4.4. One beam element is defined as two beam elements to define the steel areas at the mid-section. Hence left parts of beams have 200mm rigid zone on the left and no rigid zone on the right while right parts no rigid zone on the left and 200mm rigid zone on the right and both of them are 3000mm in length. Hoop bar diameter is 14mm with 100mm spacing at the joints and 150mm spacing at the mid-section of the beams.

Table 4.3 Axial loads on columns of FR1

Columns	Axial Load (kN)	Columns	Axial Load (kN)
1	620	7	284
2	1030	8	490
3	620	9	284
4	448	10	120
5	760	11	220
6	448	12	120

Table 4.4 Beam steel areas of FR1 (top/bottom)

BEAM NO	LEFT (mm^2)	MIDDLE (mm^2)	RIGHT (mm^2)
1	1592/904	760/1357	2044/1017
2	2044/1017	760/1357	1972/904
3	2116/1058	760/1357	1840/904
4	1972/904	760/1357	1840/904
5	1840/904	760/1357	2568/1284
6	2568/1284	760/1357	1972/904
7	2352/1212	760/1357	1978/904
8	1978/904	760/1357	2805/1412
9	2805/1412	760/1357	2228/1105
10	1614/904	760/1357	1664/904
11	1664/904	760/1357	2805/1412
12	2805/1412	760/1357	1592/904

4.2.3 Shear walls

Shear walls are defined as 5600x200mm, which is actually an unpractical value since the first aim was to develop practice with IDARC. Damage indices came out to be very large values and shear wall dimensions increased to 5600x300mm. This is a large value. As explained in Chapter 2, shear wall steel parameters are given as steel ratios in IDARC. Three subregions are used for this purpose (Figure 4.3). The axial loads, vertical and horizontal steel ratios and heights of the shear walls are given in Table 4.5.

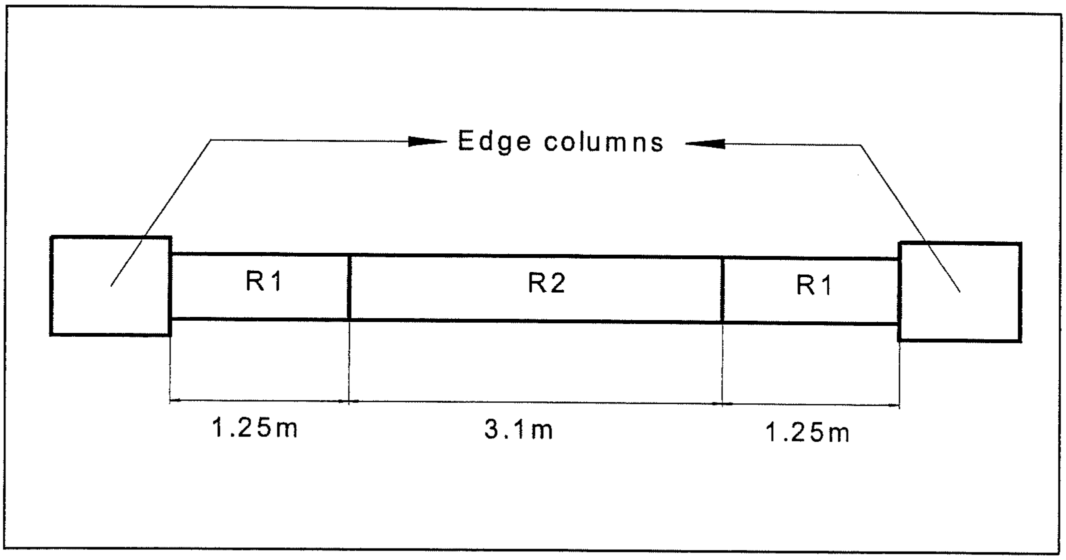

Figure 4.3 Shear wall regions of FR1

Table 4.5 Shear wall properties of FR1

Shear Wall	A. Load (kN)	R2 (ρ_v/ρ_h)	R1 (ρ_v/ρ_h)	Height (mm)
1	1900	0.005/0.007	0.0122/0.014	2750
2	1400	0.005/0.007	0.0122/0.014	3000
3	890	0.005/0.007	0.0122/0.014	3000
4	700	0.005/0.007	0.0122/0.014	3000

4.2.4 Edge Columns

Edge columns are 400x400mm and other properties for one edge column are given in Table 4.6.

Table 4.6 Edge column properties of FR1

Storey	Axial Load (kN)	Steel Area (mm^2)	Height (mm)
1	200	1885	2750
2	145	1885	3000
3	95	1885	3000
4	50	1885	3000

4.2.5 Masses

Masses are given as nodal weights at the joints at each storey. The values of masses are given in Table 4.7.

Table 4.7 Nodal weights of FR1 at the joints (from left to right)

Storey	Joint 1(kN)	Joint 2(kN)	Joint 3(kN)	Joint 4(kN)	Joint 5(kN)
1,2,3	164	270	305	305	164
4	121.4	222.2	238.7	238.7	121.4

4.2.6 Concrete and Steel Properties

Unconfined concrete strength is taken as 0.025 kN/mm^2, and strain at max compression is 0.2% while the ultimate strain is 0.3%. Yield and ultimate strength of steel are 0.22 kN/mm^2 and 0.34 kN/mm^2, respectively.

4.2.7. Dynamic Analysis Options

The time interval for analysis is taken as 0.001sec. El Centro – Imperial Valley Earthquake data is used as ground motion data. Earthquake properties are given in Table 4.6. Mass proportional damping is used with 2% critical damping.

Table 4.8 Properties of El Centro Earthquake

PGA(mm/s^2)	231.44
PGV(mm/s)	65.87
PGD(mm)	29.62
Time Interval (sec)	0.01
Effective Duration(sec)	11.28
V/A(sec)	0.2846
Data	3000

The parameters examined in the first frame are peak ground acceleration (PGA), shear wall area ratio (SWAR), shear wall steel ratio (SWSR), column area ratio (CAR), column steel ratio (CSR), storey height (SH), damping ratio (DR) and damping type. Due to problems occurred during analysis, input files are prepared for kips-inch system of units also

SWAR is defined by the ratio of shear wall area to the effective plan area of the building (Figure 4.4). For 5600x300 shear wall SWAR is 0.009. CAR is calculated similar to SWAR. Although in the graphs presented later SWSR is the vertical steel ratio value, horizontal steel ratio is changed according the percent change in the horizontal steel. In the following graphs abbreviations are as follows: RZ is Rayleigh damping, SP is stiffness proportional damping, MP is mass proportional damping, k-i is kips-inch system of units, k-mm, is kN-mm system of units.

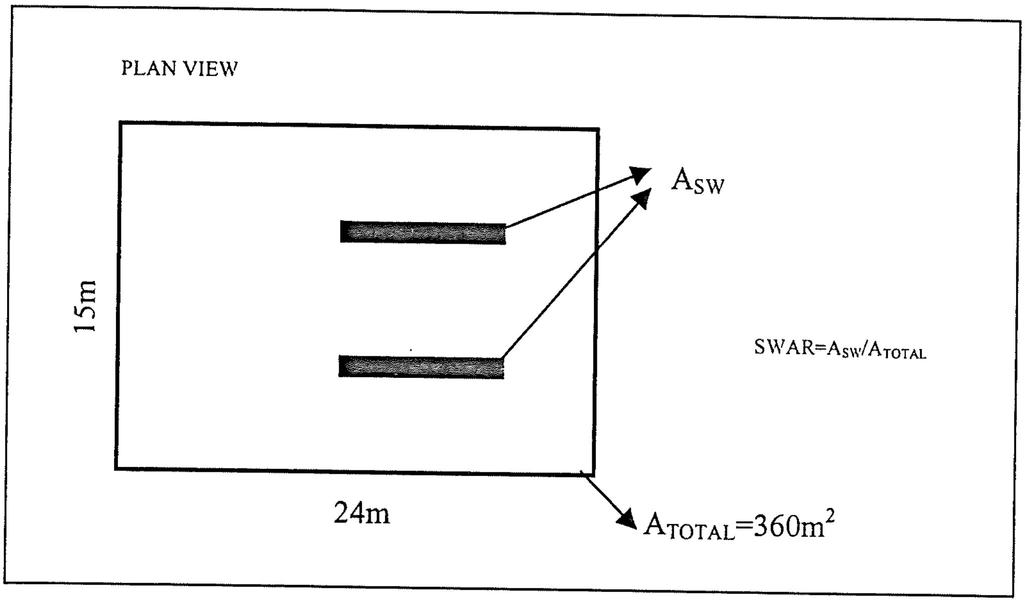

Figure 4.4 Definition of SWAR

4.2.8 Results

The points marked with circle represents huge amounts of damage value which results from the instabilities of the IDARC, but just to visualise the behaviour, their damage value is shown as 1.

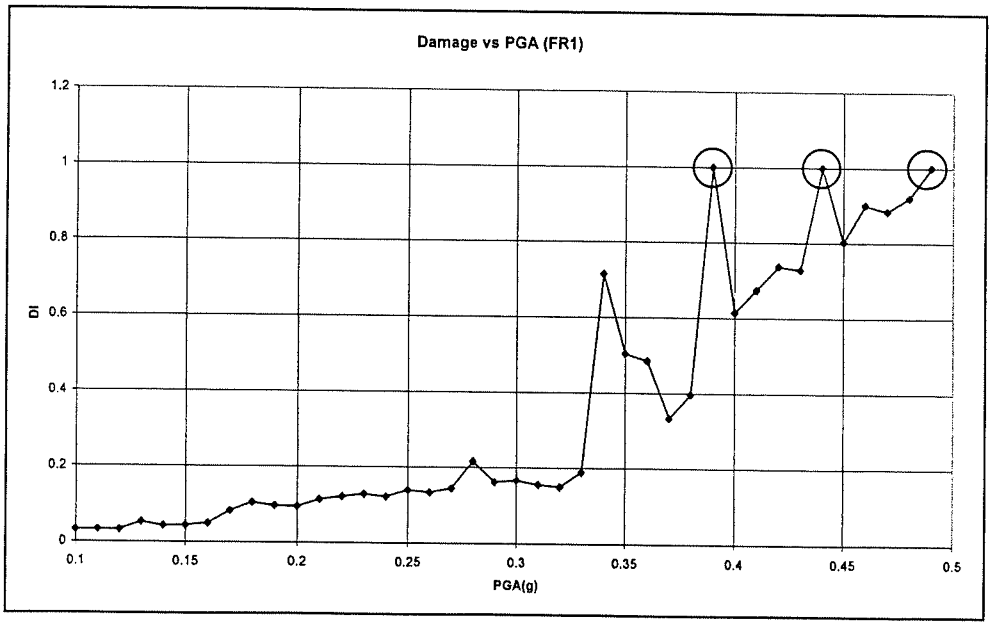

Figure 4.5 Effect of PGA on damage of FR1

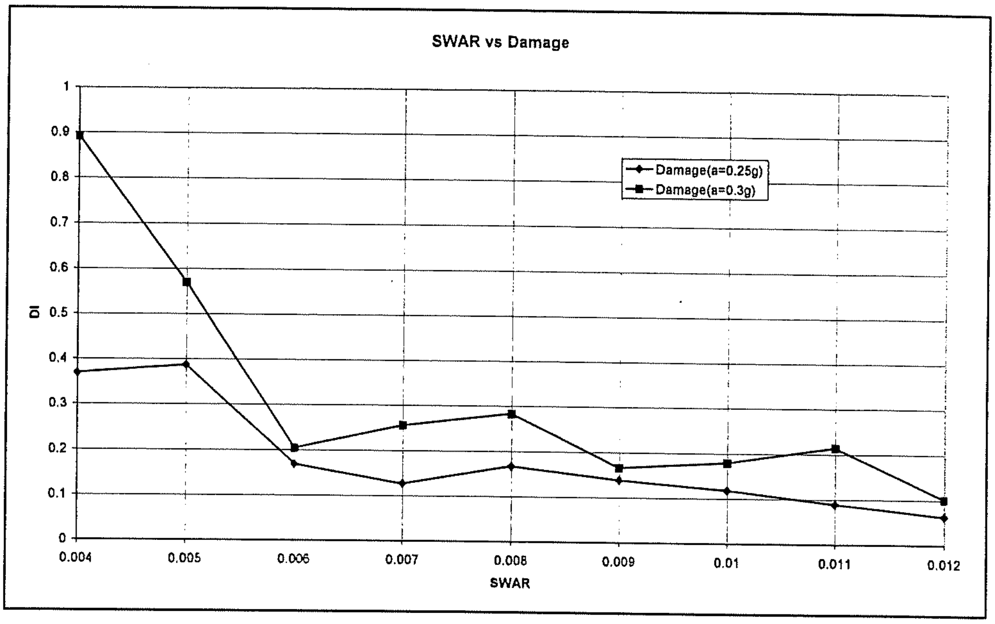

Figure 4.6 Effect of SWAR on damage of FR1

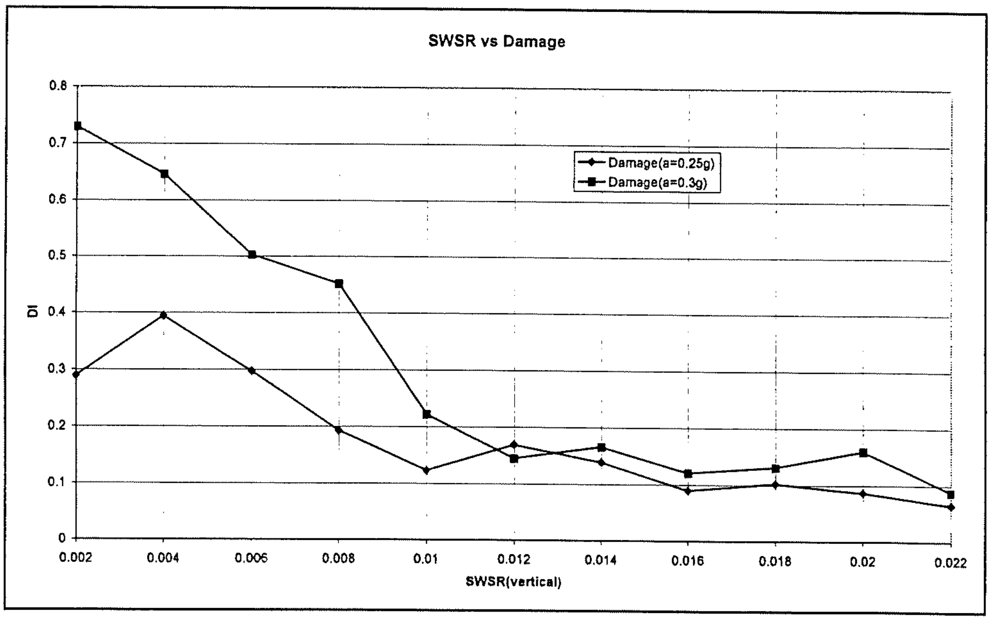

Figure 4.7 Effect of SWSR on damage of FR1

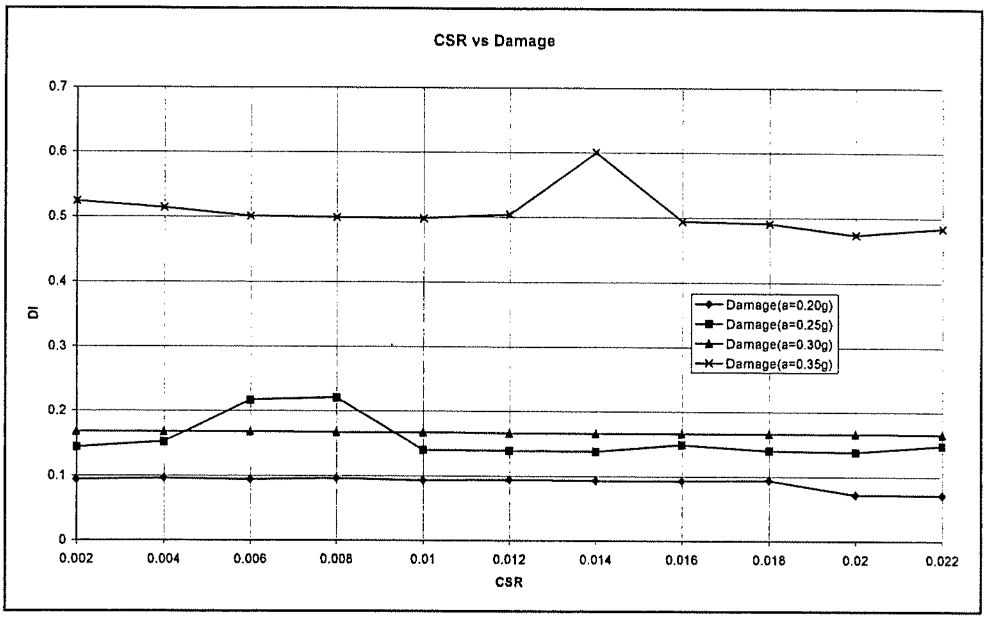

Figure 4.8 Effect of CSR on damage of FR1

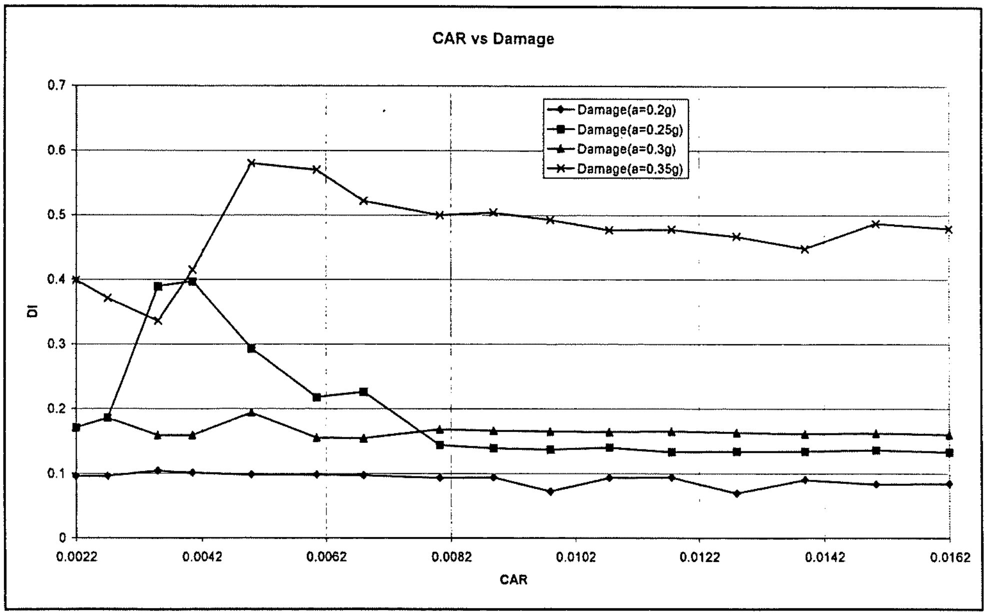

Figure 4.9 Effect of CAR on Damage of FR1

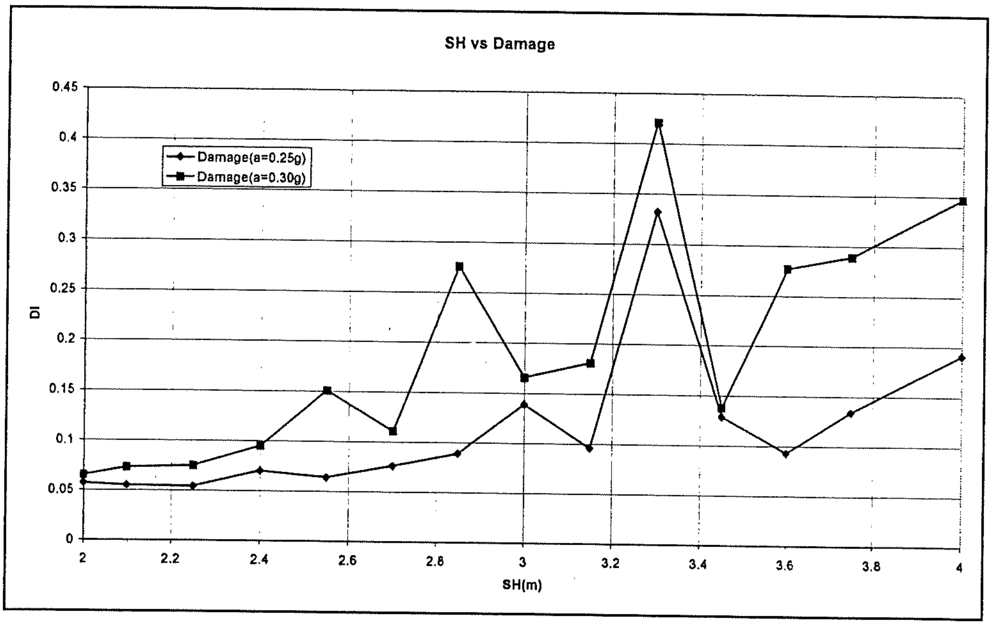

Figure 4.10 Effect of SH on damage of FR1

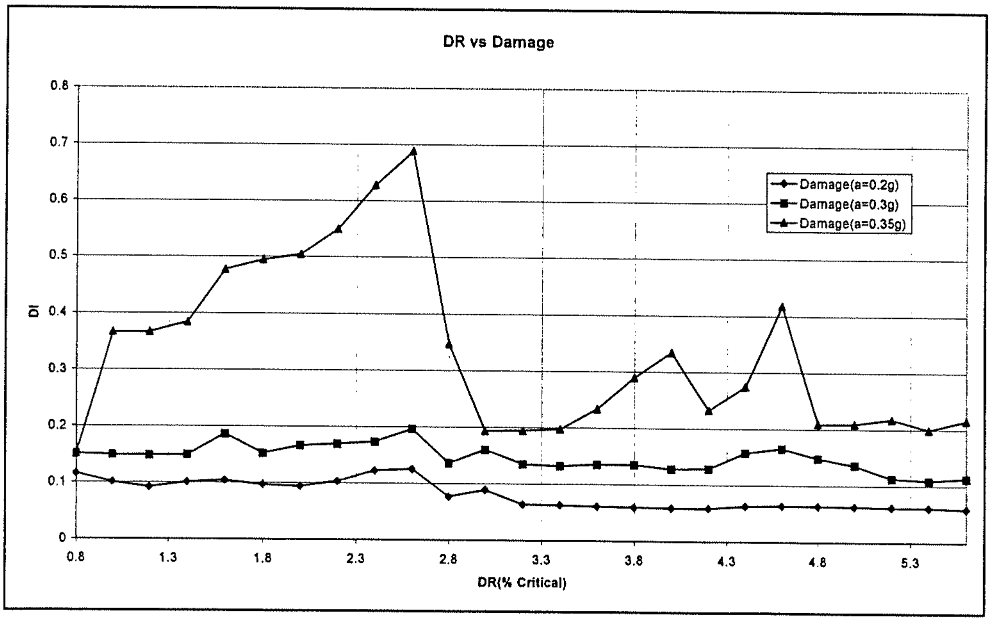

Figure 4.11 Effect of DR on damage of FR1

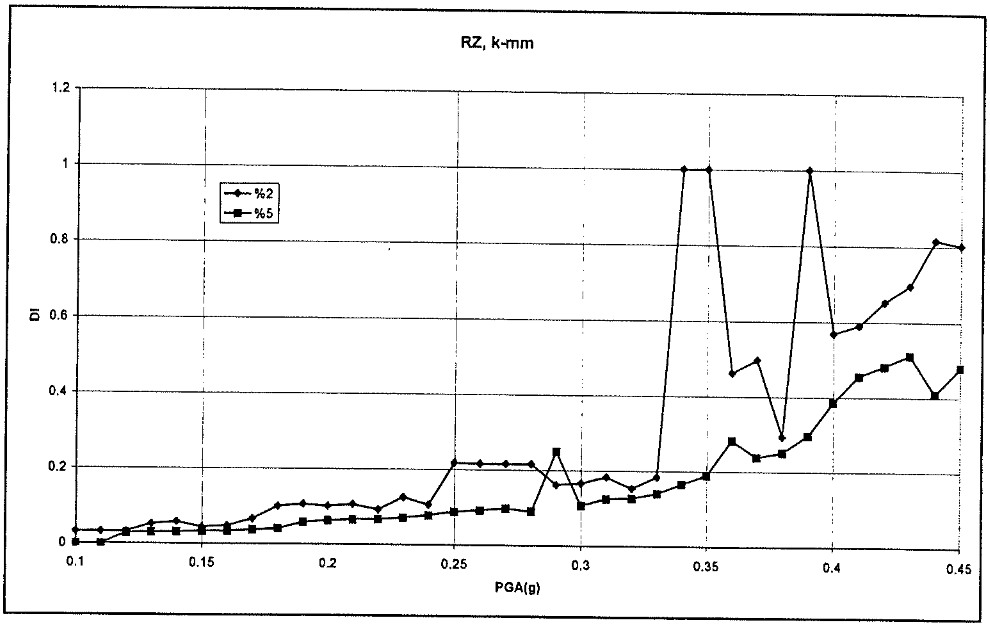

Figure 4.12 Effect of PGA on damage of FR1 when RZ, k-mm is used for DR=2,5%

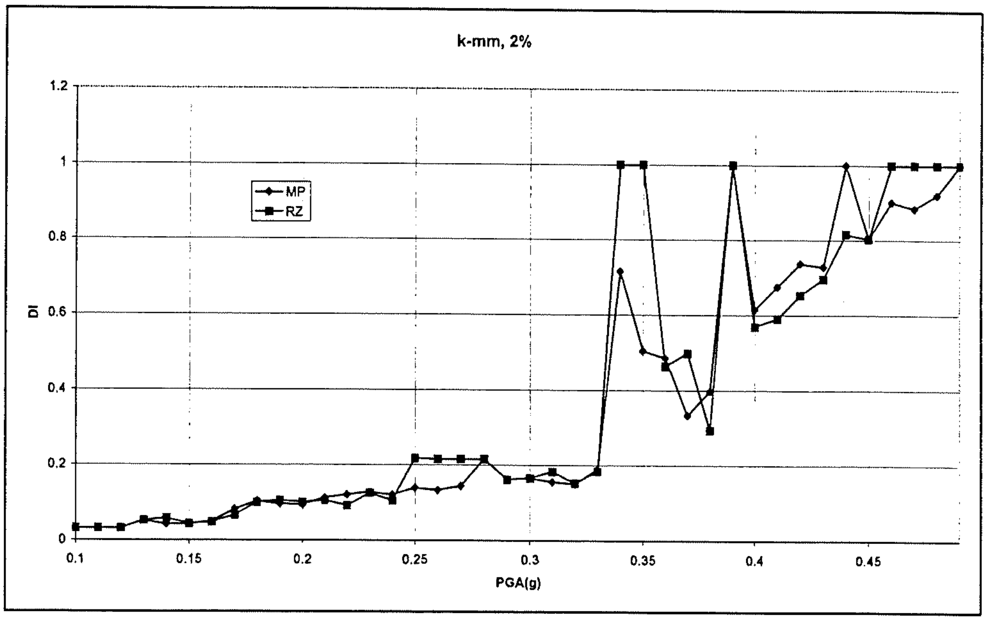

Figure 4.13 Effect of PGA on damage of FR1 when DR=2%, k-mm is used for MP and RZ damping types

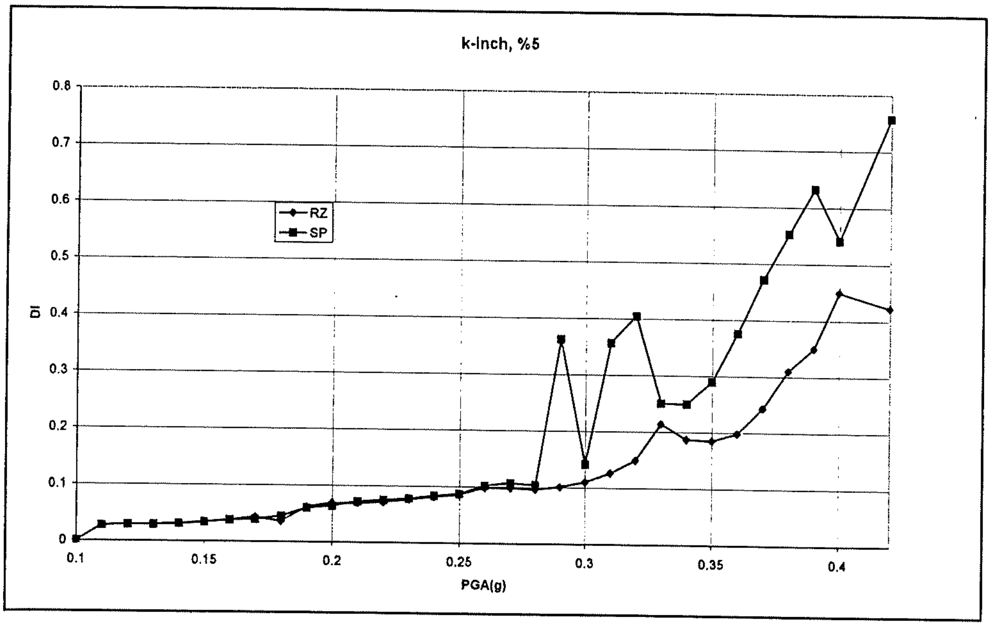

Figure 4.14 Effect of PGA on damage of FR1 when DR=5%, k-inch is used for MP and RZ damping types

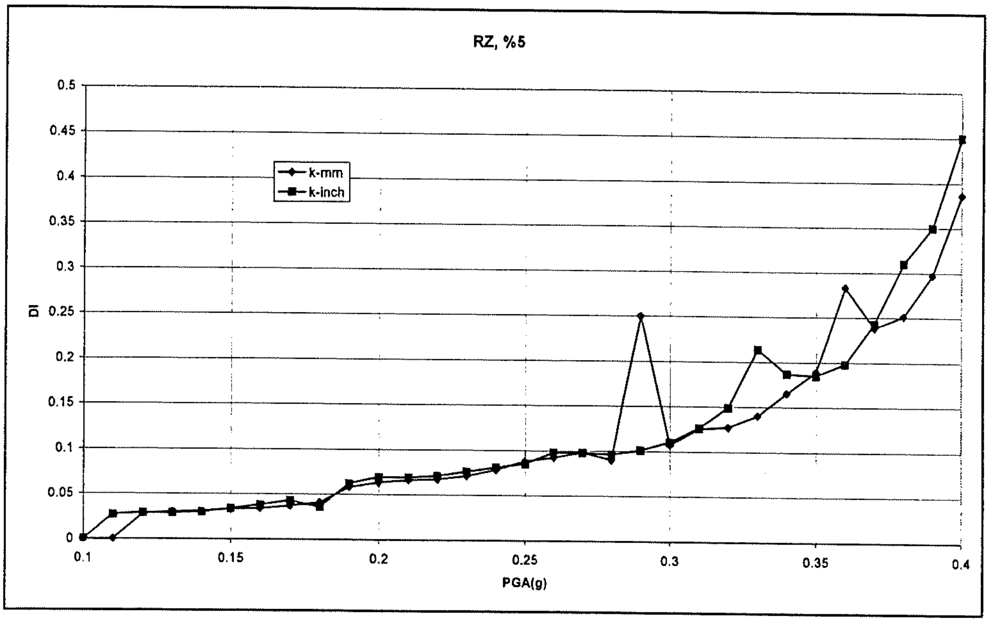

Figure 4.15 Effect of PGA on damage of FR1 when DR=5%, RZ is used for two types of system of units

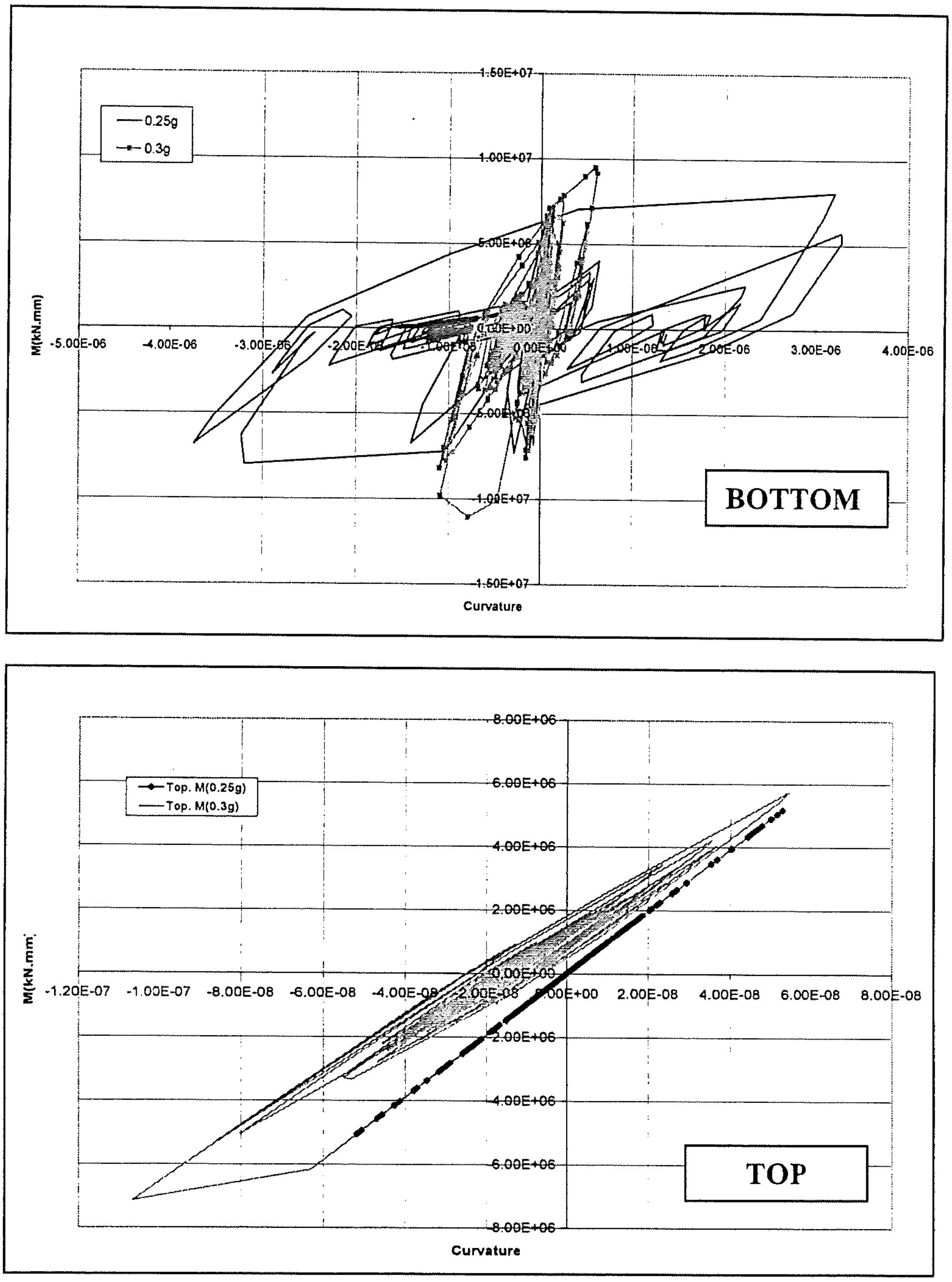

Figure 4.16 Moment-Curvature curves of the top and bottom cross-section of the first shear wall of FR1

4.3 Case 2: SDOF

Although it is expected that in the analyses of FR1 the PGA-DI graphs for both system of units to be same, they are very different as shown in Figure 4.15. A simple SDOF system as shown in Figure 4.15 is analysed to see the difference. Two graphs for the SDOF system are presented here.

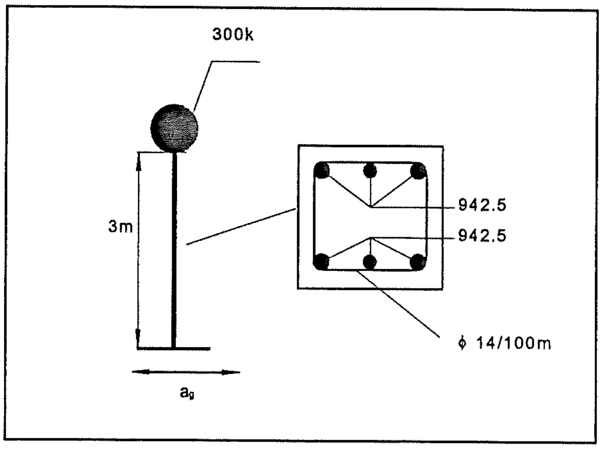

Figure 4.17 Properties of SDOF

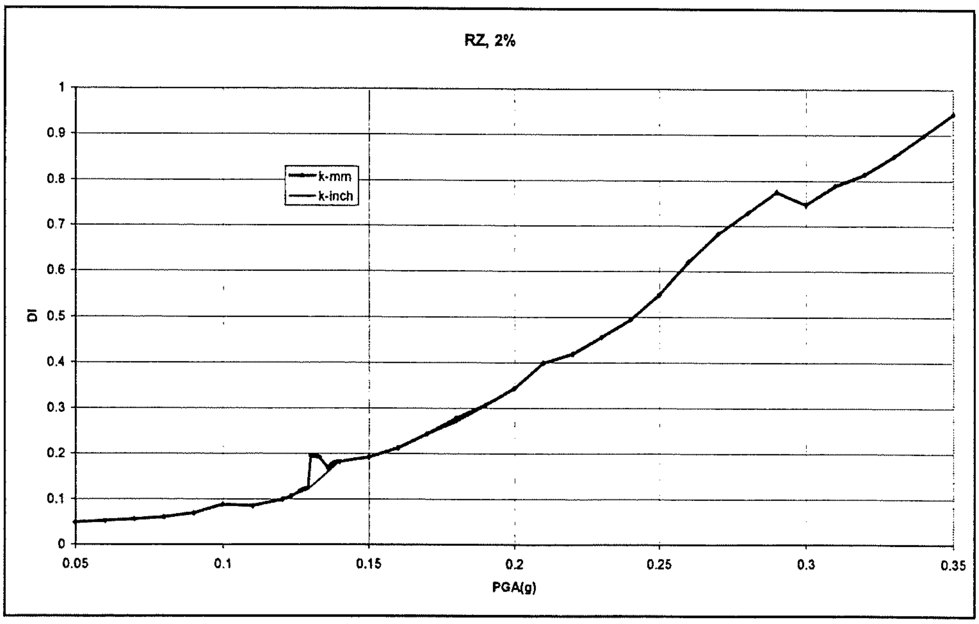

Figure 4.18 Effect of PGA on damage of SDOF for two system of units

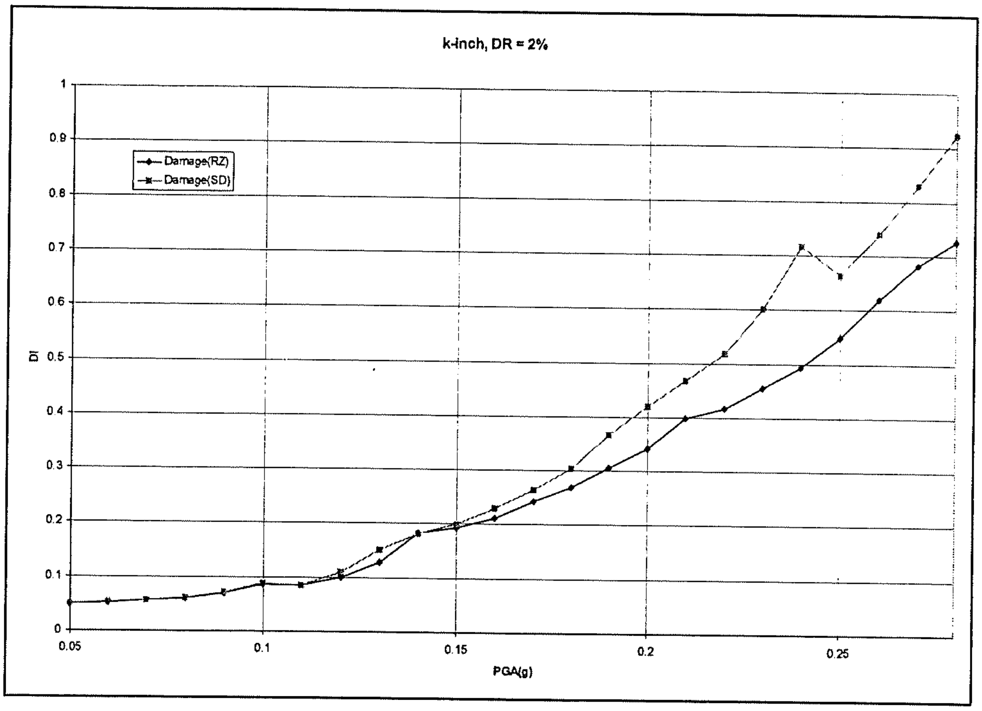

Figure 4.19 Effect of PGA on damage of SDOF for two damping types

4.4 Case 3: FR2

The frame FR2 is a five-storey, four bay frame as shown in Figure 4.20. T This frame is designed according to TS-500, EUROCODE-2, ACI specifications.

4.4.1 Columns

All column elements used are the same. Typical column is 19.68x19.68in and 118.1in in height with 9.84in rigid zones at top and bottom, except the columns in the first storey they have no rigid zone at bottom. The initial axial loads on the columns are given in Table 4.9. Hoop bar diameter is 0.47in with spacing of 2.95in. The effectiveness of hoop bar is taken as 0.5. The area of vertical steel on one face is 1.95in^2. Hence the steel ratio is 0.010.

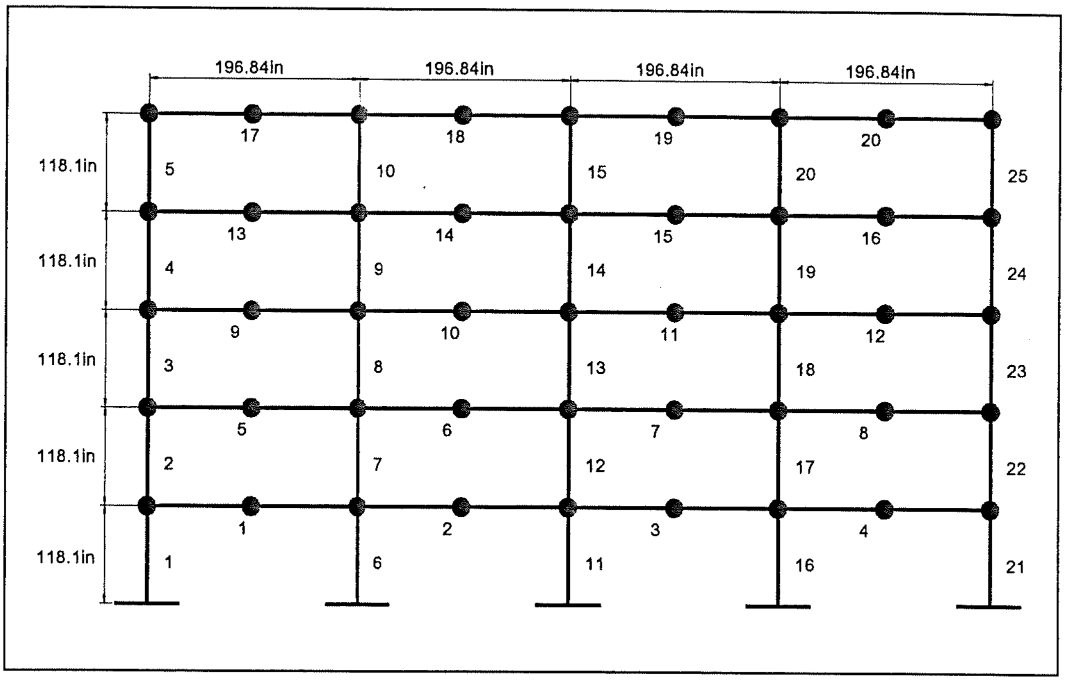

Figure 4.20 General view of FR2

Table 4.9 Axial loads on columns of FR2

Column	Axial Load(kips)	Column	Axial Load(kips)
1,21	141.19	6,11,16	274.5
2,22	121.4	7,12,17	219.76
3,23	91.05	8,13,18,	164.34
4,24	60.70	9,14,19	109.7
5,25	29.90	10,15,20	55.30

4.4.2 Beams

Every parameter of beams is the same except the vertical steel areas. The top and the bottom steel areas are given in Table 4.10. One beam element is defined as two beam elements to define the steel areas at the mid-section. Hence left parts of beams have 9.84 in rigid zone on the left and no rigid zone on the right while right parts no rigid zone on the left and 98.4in rigid zone on the right and both of them are 98.42 in in length. Hoop bar diameter is 0.39 in with 5.31 in spacing throughout the beam.

Table 4.10 Beam steel areas of FR2 (top/bottom)

BEAM NO	LEFT(in^2)	MIDDLE(in^2)	RIGHT(in^2)
1,5,9,13,17	1.78/0.95	0.59/1.43	1.78/1.43
2,6,10,14,18	1.78/1.43	0.59/1.43	1.78/1.43
3,7,11,15,19	1.78/1.43	0.59/1.43	1.78/1.43
4,8,12,16,20	1.78/1.43	0.59/1.43	1.78/0.95

4.4.3 Masses

Masses are given as nodal weights at the joints at each storey. The values of nodal weights are given in Table 4.10.

Table 4.11 Nodal weights of FR2 at the joints (from left to right)

Storey	J1(kN)	J2(kN)	J3(kN)	J4(kN)	J5(kN)	J6(kN)	J7(kN)	J8(kN)	J9(kN)
1,2,3,4	10.70	13.31	17.35	13.31	17.35	13.31	17.35	13.31	10.70
5	7.83	13.31	15.33	13.31	15.33	13.31	15.33	13.31	7.83

4.4.4 Concrete and Steel Properties

Unconfined concrete strength is taken as 2.9 kips/in^2, and strain at max compression is 0.2% while the ultimate strain is taken as 2.7% for columns and 1.2% for beams. Yield and ultimate strength of steel are 60.9 kips/in^2, and 72.5 kips/in^2 respectively.

4.4.5. Dynamic Analysis Options

The earthquake data used for FR2 is same as FR1 namely El Centro earthquake. The parameters examined in FR2 are peak ground acceleration (PGA), column dimensions percent (CDP), column steel percent (CSP), beam dimensions percent (BDP), beam steel percent (BSP), damping ratio (DR) and mass percent (MP). The original structure assumed to be 100% and the changes were done accordingly. For example 100% of CDP is 19.68x19.68in hence 80% CDP is 15.74x15.74in column. The same rule applies to other parameters.

4.4.6 Results

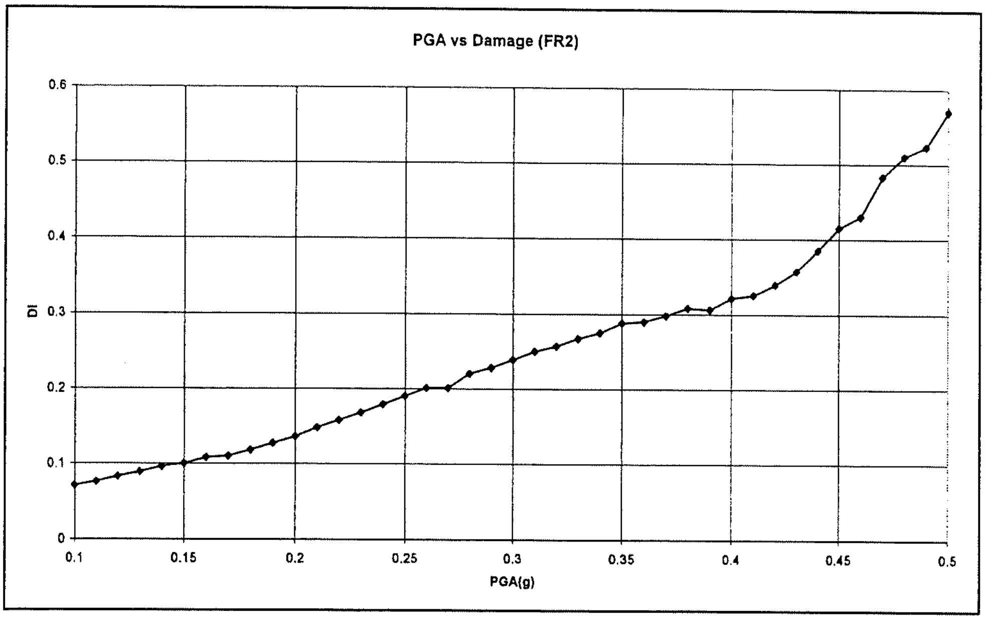

Figure 4.21 Effect of PGA on damage of FR2

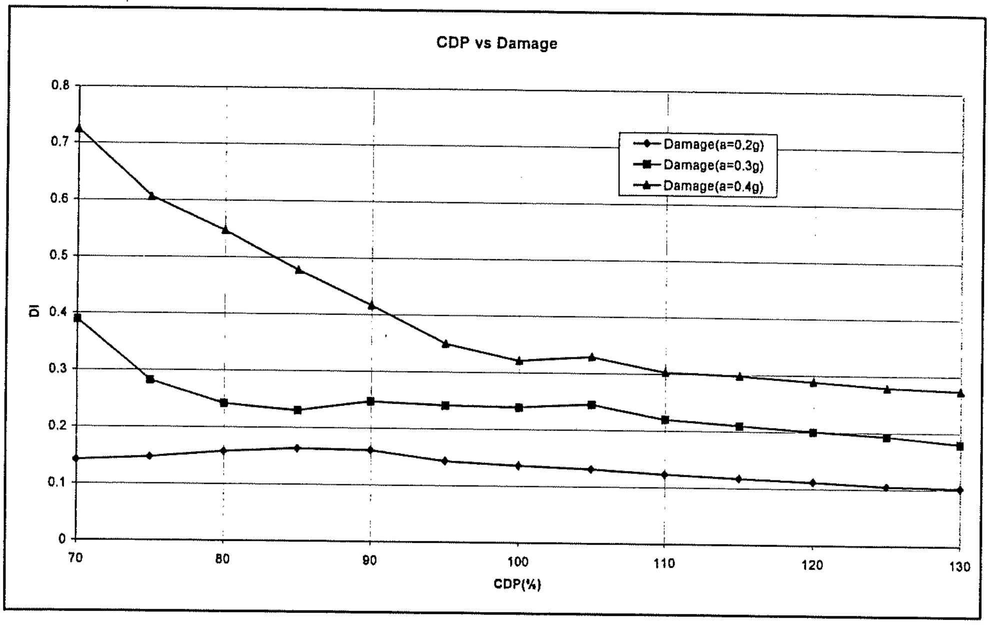

Figure 4.22 Effect of CDP on damage of FR2

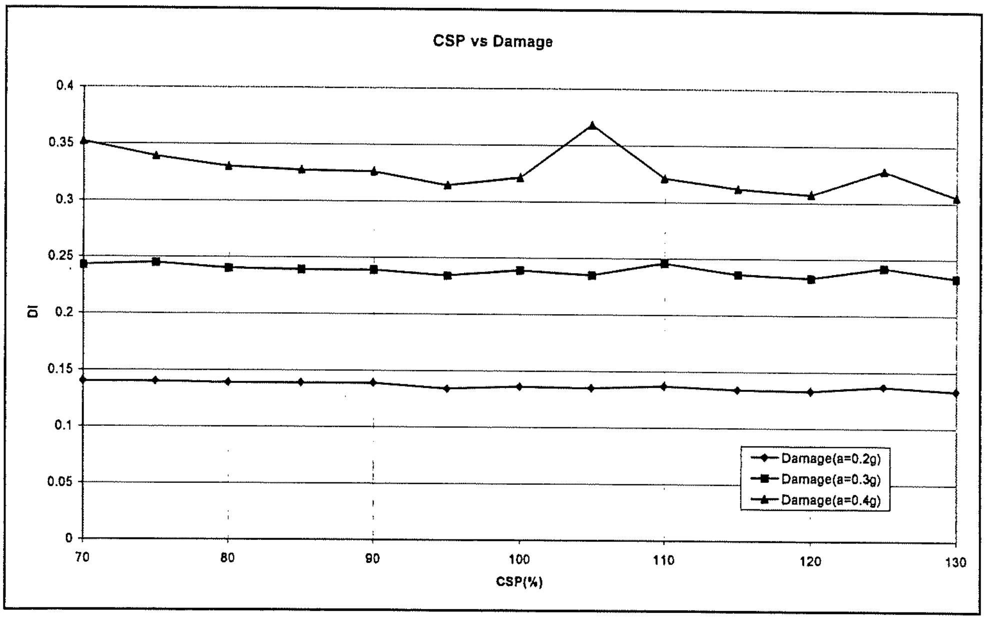

Figure 4.23 Effect of CSP on damage of FR2

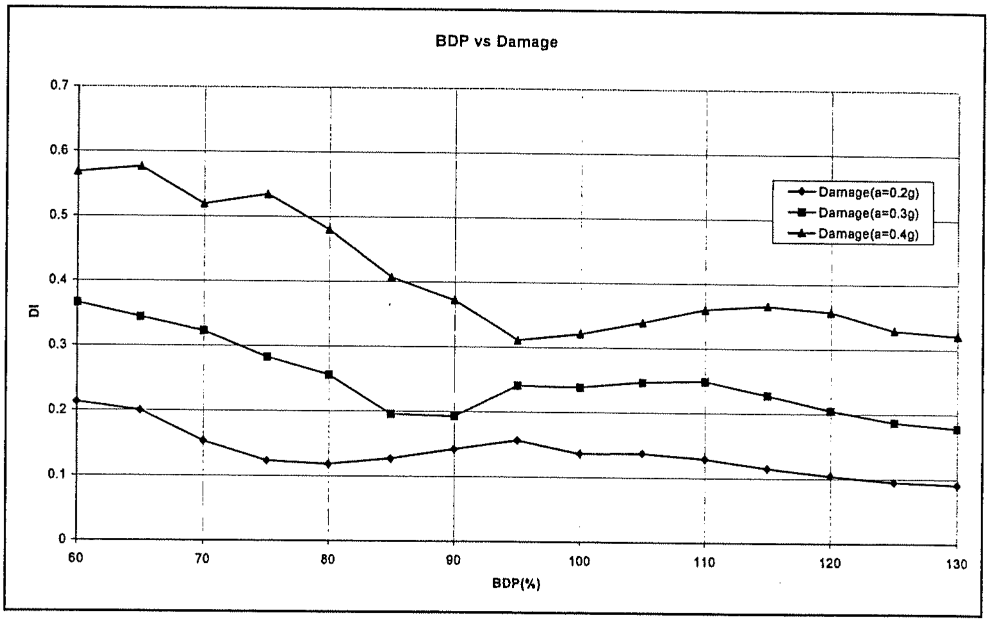

Figure 4.24 Effect of BDP on damage of FR2

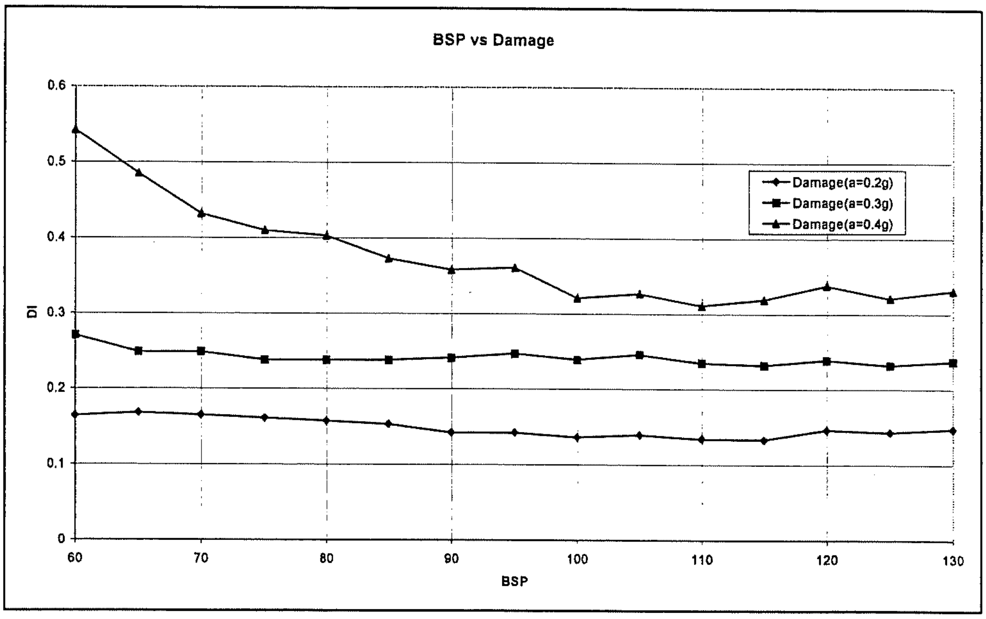

Figure 4.25 Effect of BSP on damage of FR2

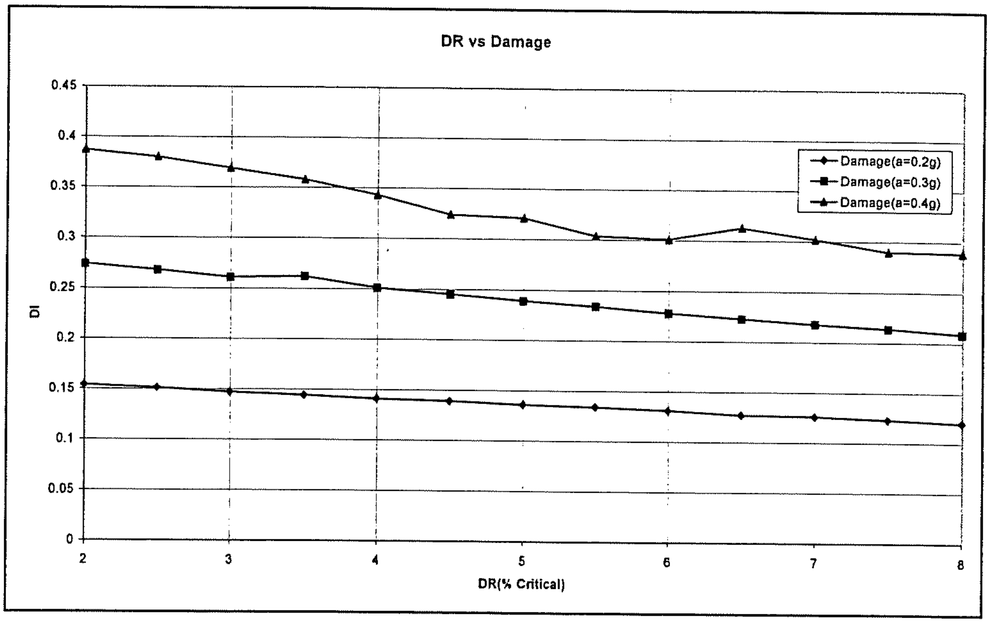

Figure 4.26 Effect of DR on Damage of FR2

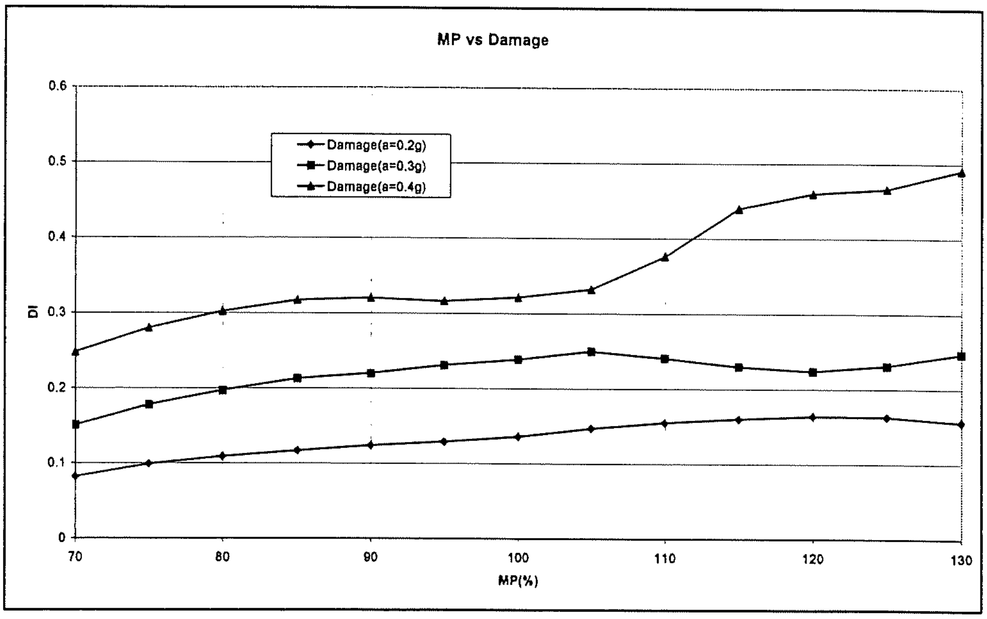

Figure 4.27 effect of MP on damage of FR2

4.5 Case 4: FR3

The frame FR3 is a four-storey, four bay frame as shown in Figure 4.28.

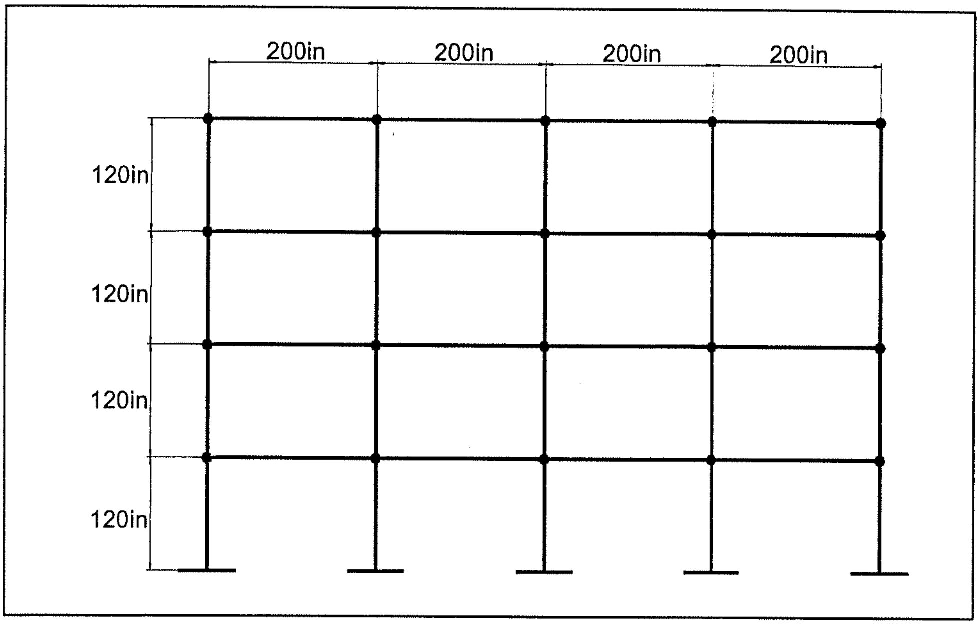

Figure 4.28 General view of FR3

4.5.1 Columns

All column elements used are same. Typical column is 20x20in and 120in in height with 10in rigid zones at top and bottom, except the columns in the first storey they have no rigid zone at bottom. The initial axial loads on the columns are given in Table 4.12. Hoop bar diameter is 0.5in with spacing of 3in. The effectiveness of hoop bar is taken as 0.5. The area of vertical steel on one face is $2in^2$. Hence the steel ratio is 0.010.

4.5.2 Beams

Every parameter of beams is same including the vertical steel areas. The top and bottom steel areas for all of the beams is $1.5in^2$. Beam elements are 200in in length and they have 10in rigid zones at both ends.

Table 4.12 Axial loads on columns of FR3

Storey	Exterior(kips)	Interior(kips)
1	75	110
2	55	80
3	35	50
4	15	20

4.5.3 Masses

Nodal weights are given as in Table 4.13.

Table 4.13 Nodal weights of FR3 at the joints

Storey	Exterior(kips)	Interior(kips)
1,2,3	20	30
4	15	20

4.5.4 Concrete and Steel Properties

Unconfined concrete strength is taken as 2.9 kips/in^2, and strain at max compression is 0.2% while the ultimate strain is taken as 3.0%. Yield and ultimate strength of steel are 60.0 kips/in^2 and 75.0 kips/in^2 respectively.

4.5.5. Dynamic Analysis Options

The earthquake data used for FR3 is same as FR1 and FR2. The parameters examined in FR3 are same as in the previous case.

4.5.6 Results

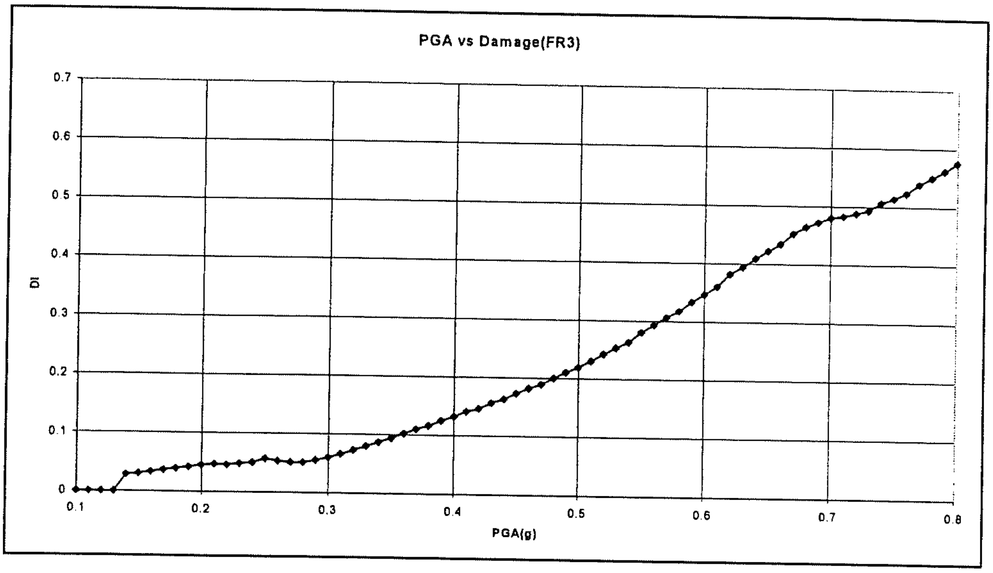

Figure 4.29 Effect of PGA on damage of FR3

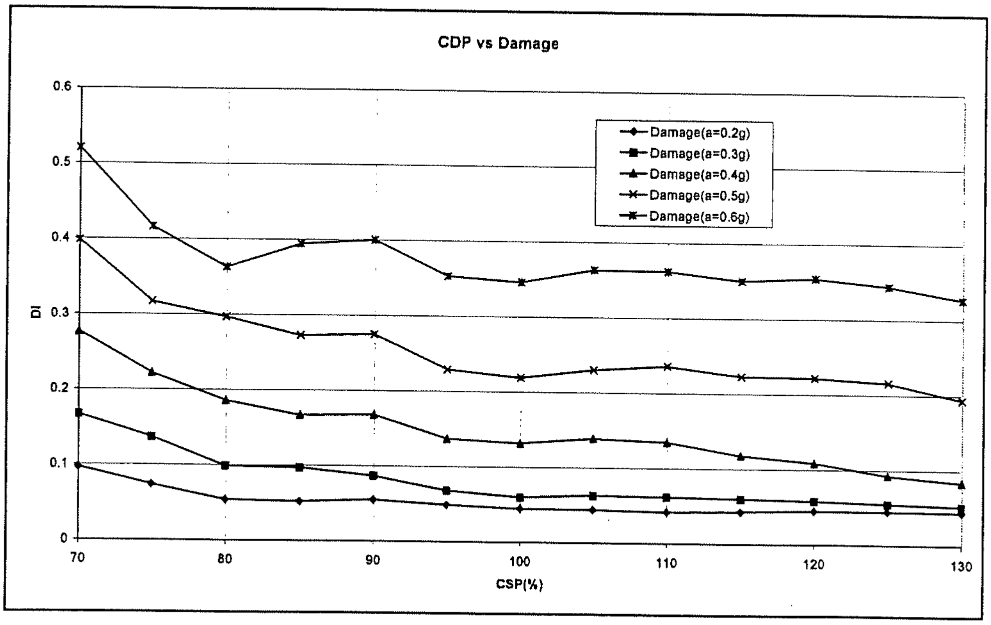

Figure 4.30 Effect of CDP on damage of FR3

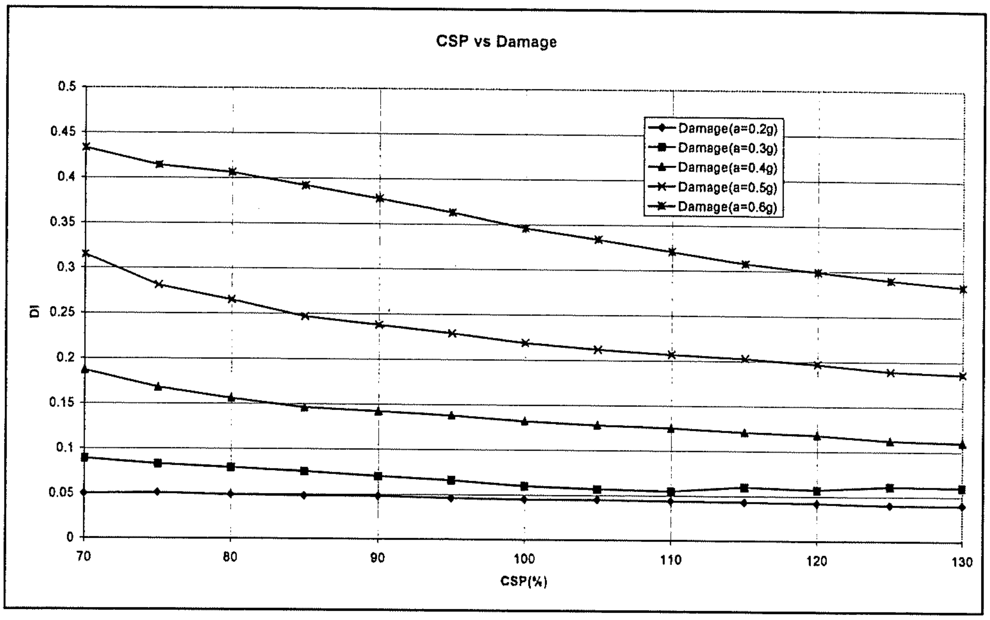

Figure 4.31 Effect of CSP on damage of FR3

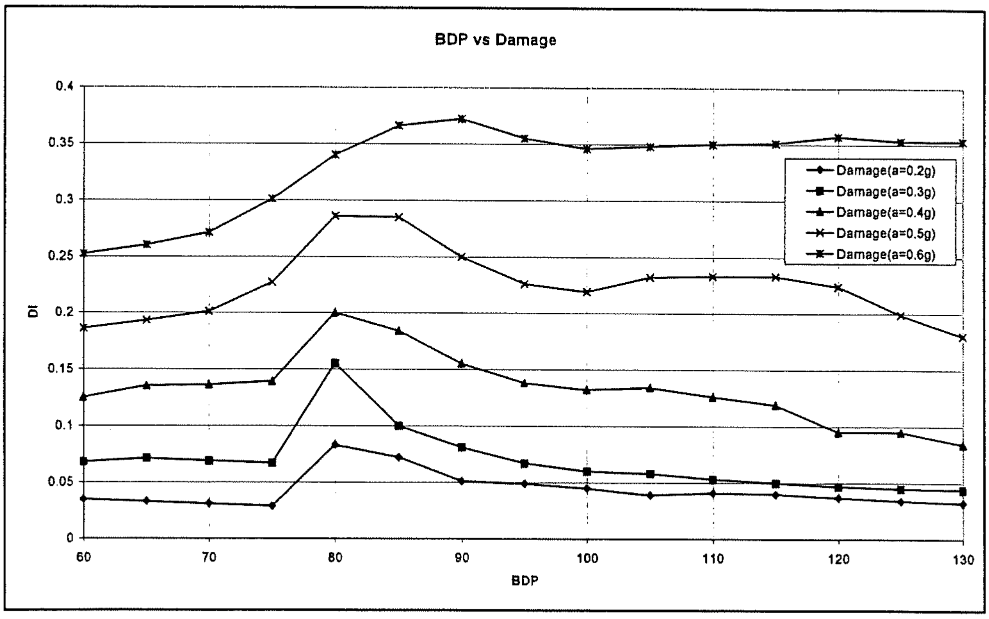

Figure 4.32 Effect of BDP on damage of FR3

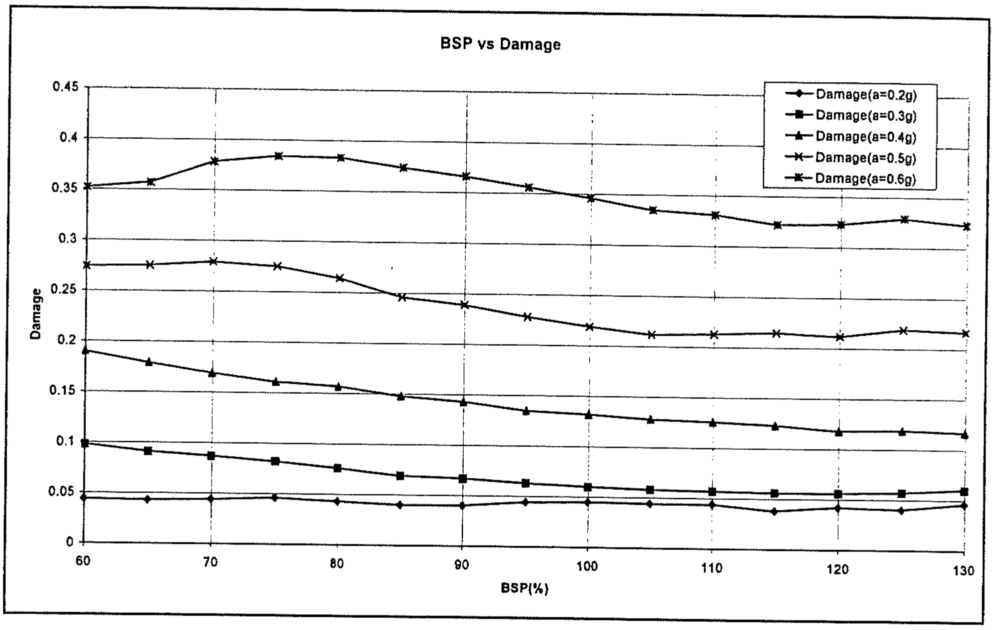

Figure 4.33 Effect of BSP on damage of FR3

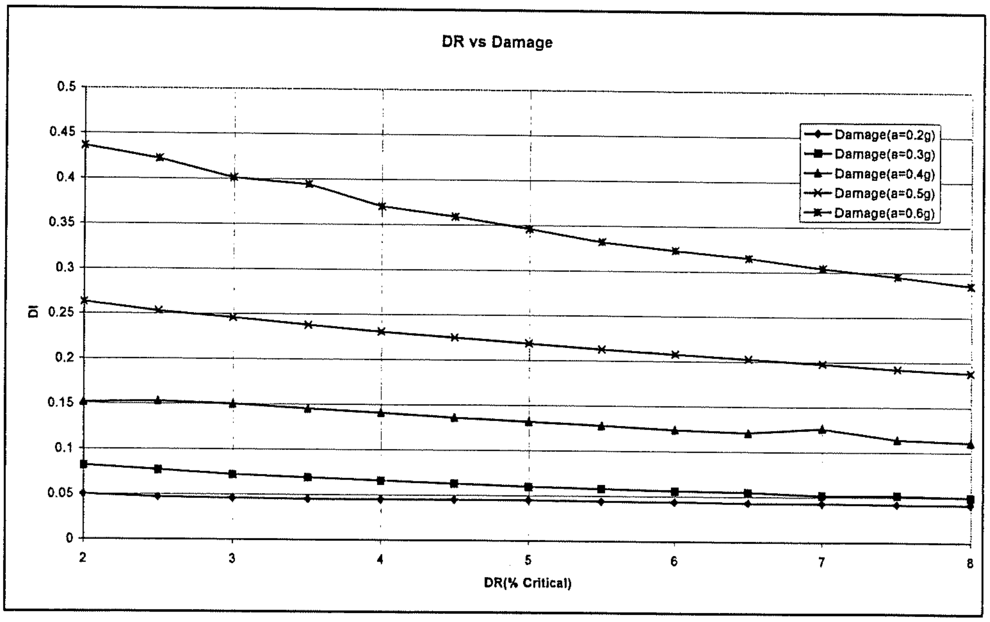

Figure 4.34 Effect of DR on damage of FR3

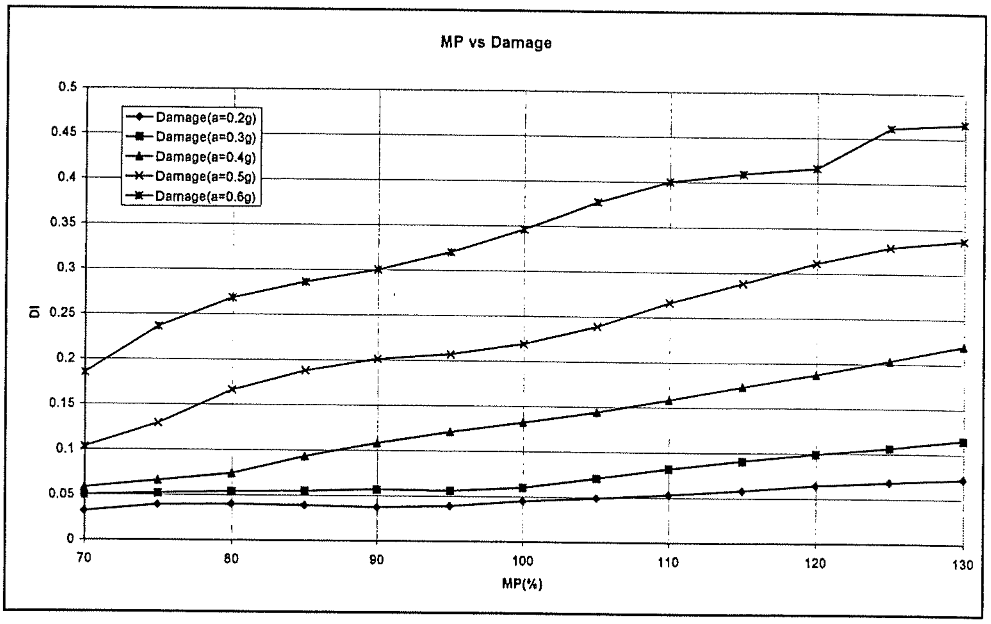

Figure 4.35 Effect of MP on damage of FR3

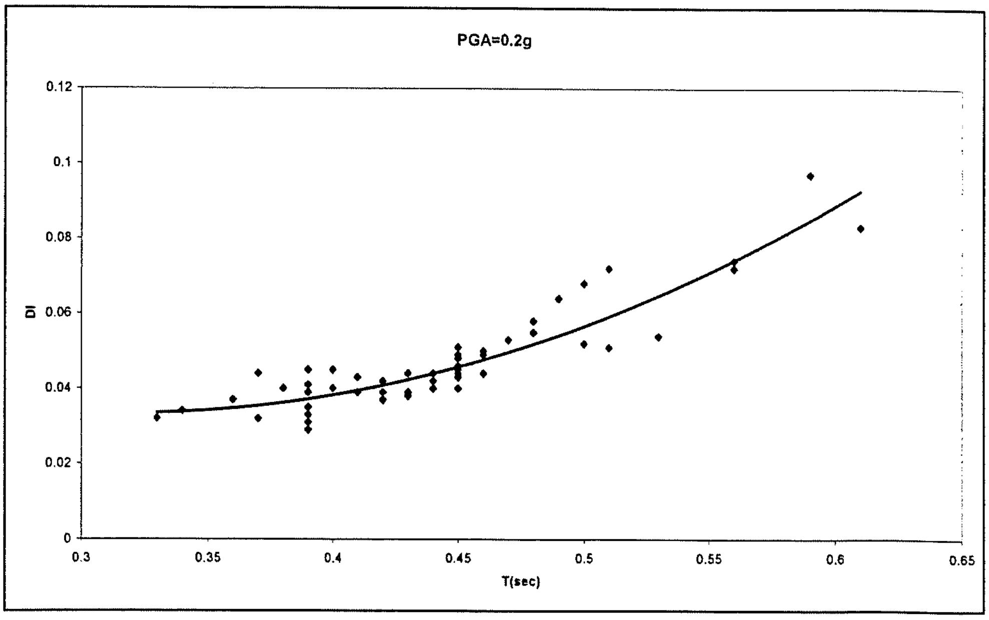

Figure 4.36 Effect of T on damage of FR3 when PGA=0.2g

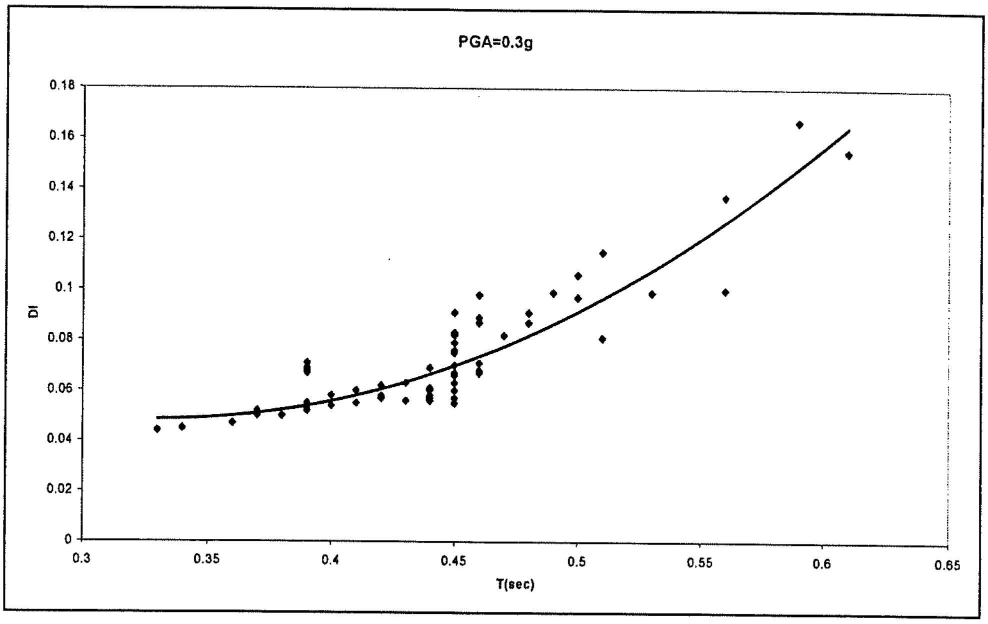

Figure 4.37 Effect of T on damage of FR3 when PGA=0.3g

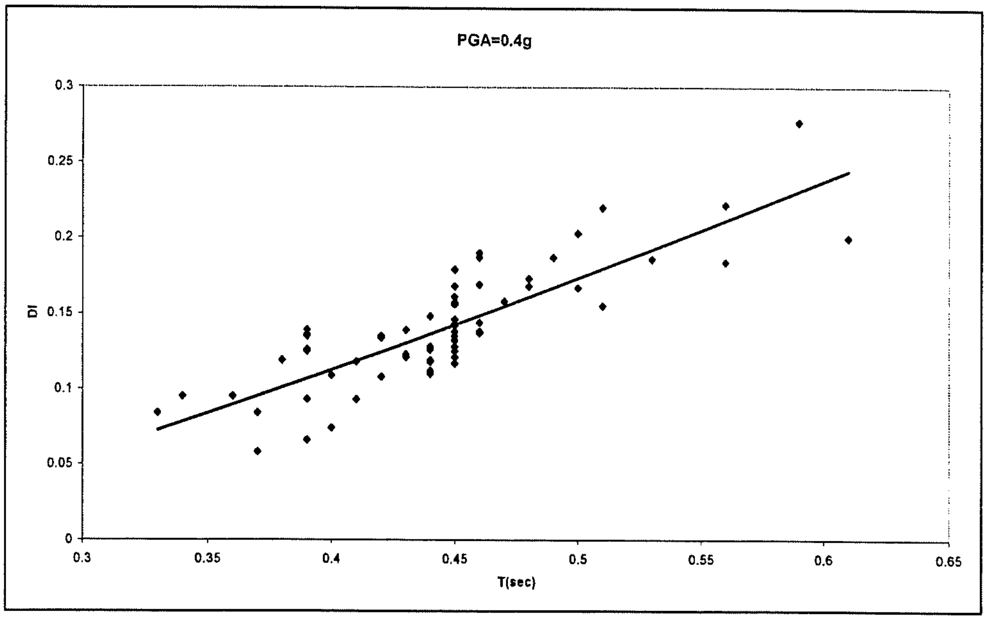

Figure 4.38 Effect of T on damage of FR3 when PGA=0.4g

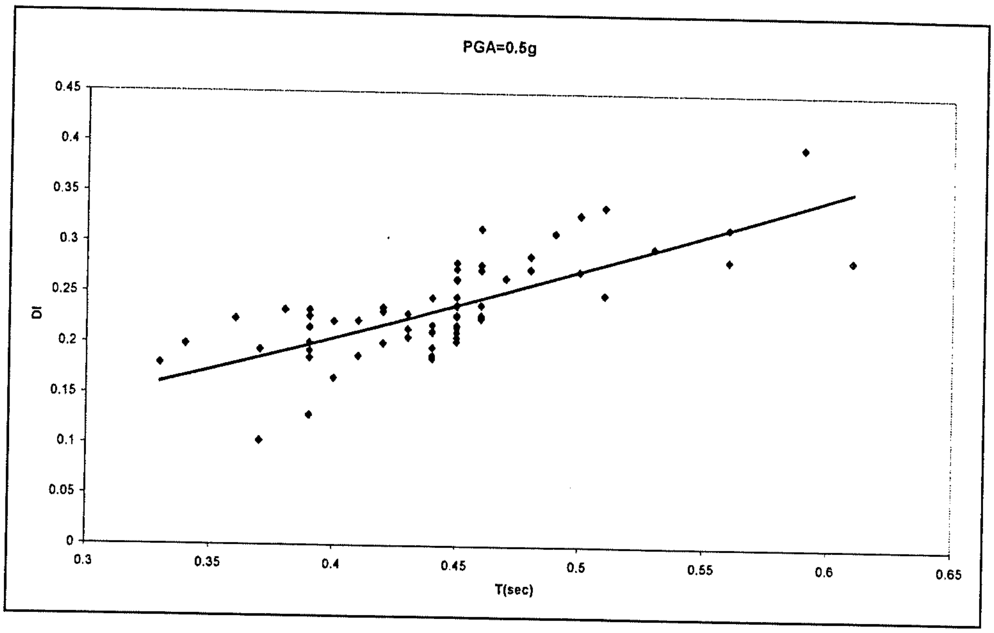

Figure 4.39 Effect of T on damage of FR3 when PGA=0.5g

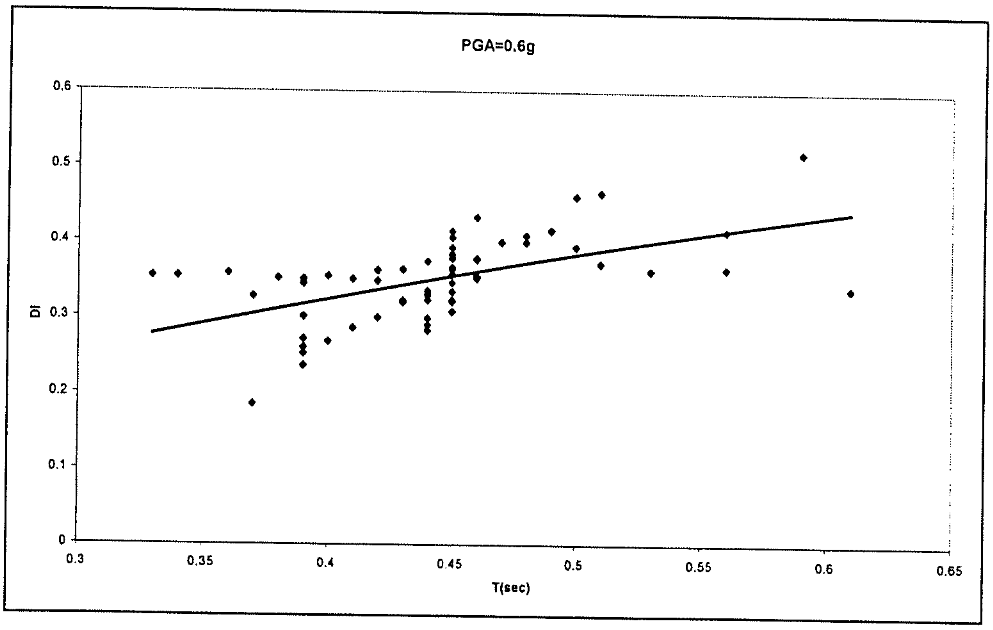

Figure 4.40 Effect of T on damage of FR3 when PGA=0.6g

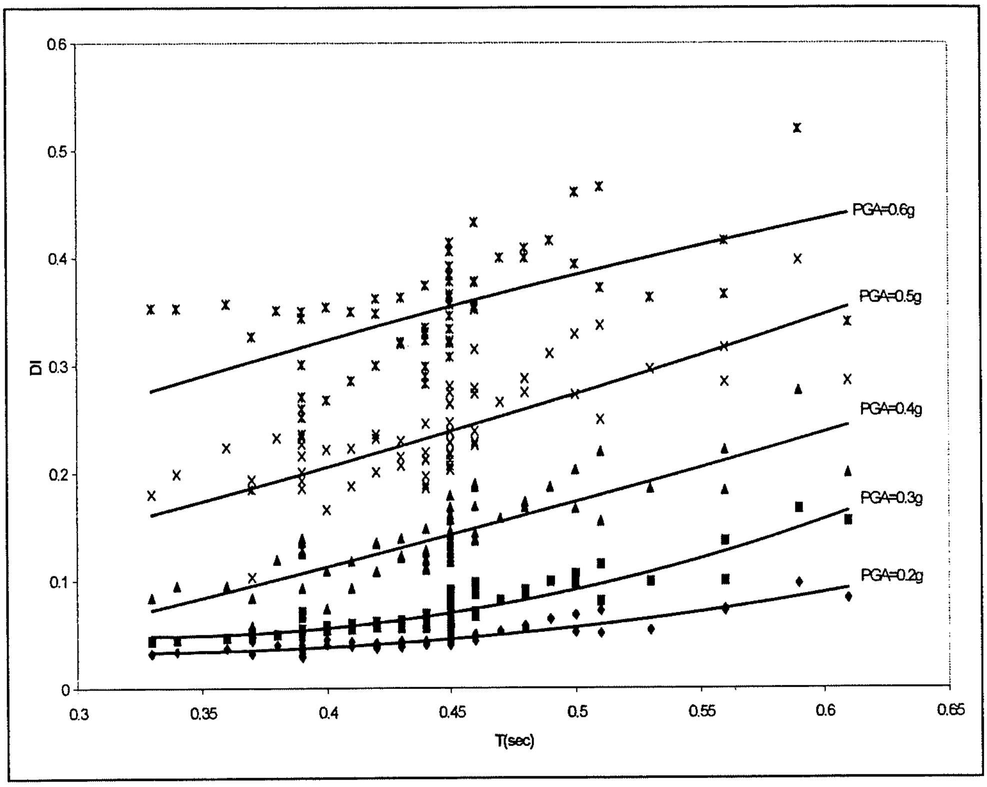

Figure 4.41 Effect of T on damage of FR3

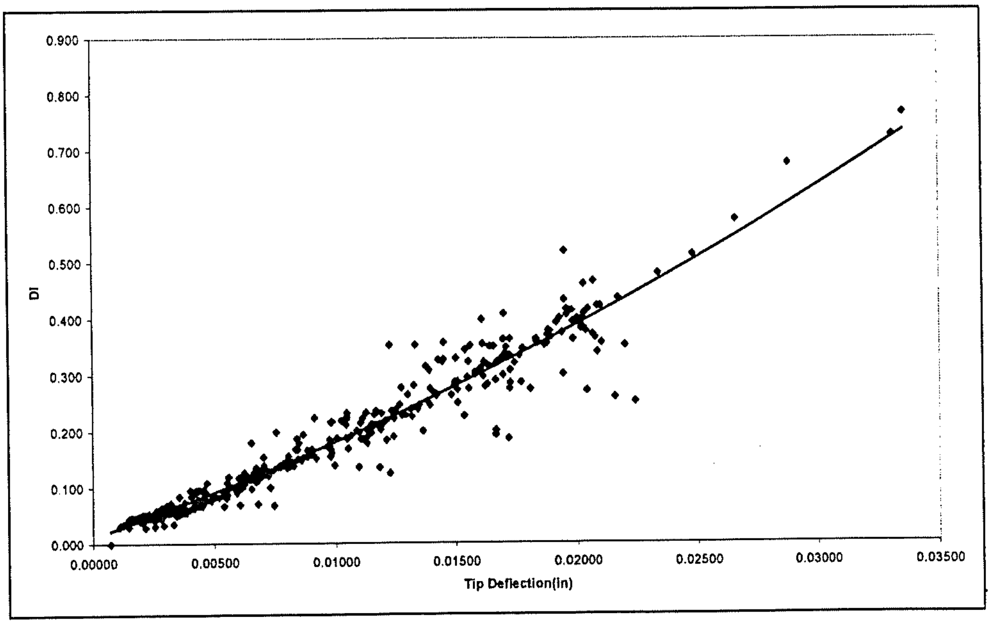

Figure 4.42 Effect of tip deflection on damage of FR3

4.6 Closing Remarks

The work done up to this point deals with the structural parameters. FR3 is analysed for the earthquakes given in Table 4.13, but not significant correlation is obtained for the parameters T_{eff}, and V/A ratio. Hence the results will be only given in tabular format. Actually for the neural network analysis it is better to enlarge the training set by adding results of analysis of different earthquakes.

Table 4.14 Outputs for other earthquakes

EQ	PGA(g)	PGV(m/s^2)	PGD(m)	V/A(sec)	T_{eff}(sec)	DI
ERZIN1.EQ	0.20	0.2614	0.0613	0.1331	7.44	0.030
SALVAD.EQ	0.20	0.2846	0.1033	0.1453	6.26	0.047
EMERVIL2.EQ	0.20	0.3129	0.1045	0.1594	8.88	0.051
SANFER1.EQ	0.20	0.3812	0.2070	0.1944	21.68	0.058
TOKACHI.EQ	0.20	0.4643	0.1132	0.2367	24.87	0.064
ERZIN1.EQ	0.30	0.3921	0.0920	0.1331	7.44	0.050
SALVAD.EQ	0.30	0.4269	0.1550	0.1453	6.26	0.070
EMERVIL2.EQ	0.30	0.4693	0.1568	0.1594	8.88	0.096
SANFER1.EQ	0.30	0.5718	0.3105	0.1944	21.68	0.090
TOKACHI.EQ	0.30	0.6965	0.1698	0.2367	24.87	0.108
ERZIN1.EQ	0.50	0.6536	0.1533	0.1331	7.44	0.107
SALVAD.EQ	0.50	0.7115	0.2584	0.1453	6.26	0.154
EMERVIL2.EQ	0.50	0.7821	0.2614	0.1594	8.88	0.308
SANFER1.EQ	0.50	0.9531	0.5175	0.1944	21.68	0.226
TOKACHI.EQ	0.50	1.1608	0.2830	0.2367	24.87	0.408
ERZIN1.EQ	0.70	0.9150	0.2146	0.1331	7.44	0.179
SALVAD.EQ	0.70	0.9961	0.3617	0.1453	6.26	0.326
EMERVIL2.EQ	0.70	1.0950	0.3659	0.1594	8.88	0.766
SANFER1.EQ	0.70	1.3343	0.7245	0.1944	21.68	0.676
TOKACHI.EQ	0.70	1.6252	0.3962	0.2367	24.87	0.726

4.7 Discussion of Results

4.7.1 General

More than 2000 runs were made with IDARC. These runs show that the software is very sensitive to changes in PGA and time interval for some case studies. For example 0.01g change in PGA caused 0.1 change in damage indicia. The reason for that is the collected error during the nonlinear analysis.

The software has several options for different parameters such as hysteretic behaviour, concrete properties, damping, etc. The user should select the proper combination for logical results. Experimental results can be utilised for this purpose. For example the effectiveness of hoop bars of columns and damping type may not reflect the actual behaviour of the hoop bars and damping. This will increase the reliability of the results. In this study the aim was not obtain the exact response of the structure but to obtain general trend of the damage and sets for training and learning for neural network implementations. Hence the parameters are not calibrated to obtain the exact response using experimental data.

Damage is observed to be very sensitive to the parameters for larger PGA. This does not mean that the structure stays in its elastic range since for elastic response the modified Park and Ang damage model will give zero damage indices. Both mass and the column area have direct effect on the damage. Assuming the flexibility of the structure for the horizontal degree of freedoms is determined by columns, it is possible to take modal periods of the structure as the most important parameter since the period of a SDOF system is estimated by

$$T = 2\pi\sqrt{\frac{m}{k}} \tag{4.1}$$

Where m is the nodal mass and k is the lateral stiffness. As in the Equation 4.1 period is proportional to the square root of mass and inversely proportional to the stiffness. In this respect beams has no direct effect on the lateral stiffness and hence on the period since they have no axial degree of freedoms. From the above discussion mass and elements that determine the lateral stiffness have direct effect on the damage.

Mass proportional damping will not be modified during the analysis since the mass is not changing during the analysis, but damping ratio should be lowered considering the damage on the structural system. To take the effect lowered damping ratio, stiffness proportional damping seems to be more logical to employ since the stiffness matrix is modified during the analysis. In that case huge loss in stiffness will yield huge loss in damping. Hence Rayleigh damping will be more realistic to use. In the later parts, the specific observations for the case studies are given.

4.7.2 FR1

Although it is expected to have a gradually increasing damage effect of PGA on FR1, the effect of PGA is very irregular for the PGA>0.3g (Figure 4.5). Several other analyses were done for different values of SWAR and time interval of nonlinear dynamic analysis but the results were not satisfactory. The effect of system of units can be observed from Figure 4.15. These improper results can be observed for the Figures from 4.8 to 4.15 also. This is thought to be a result of possible numerical instabilities resulting from the nonlinear analysis or the shear wall modelling of IDARC. Figure 4.16 shows that shear wall is damaged more when PGA is 0.25g. In literature there is not any reference that points to these problems but several researchers reported these kind of errors in IDARC.

Although the results were not satisfactory, it is possible to observe that the SWAR and SWSR are the most important parameters for damage. Another interesting observation is that the effect of shear wall is so small when

SWAR>0.006 and SWSR>0.01. Another point is when PGA>0.2g the damage increased suddenly for small values of SWAR and SWSR (Figures 4.6, 4.7).

4.7.3 SDOF

Figure 4.18 shows that there is a possible error resulting from the system of units in IDARC. Figure 4.19 shows that stiffness proportional damping modelling causes more damage in SDOF. This is an expected result as told in 4.7.1.

4.7.4 FR2

Figures 4.22, 4.24, 4.25, 4.27 show that the design specifications supplied the most economical designs for the PGA>0.2g. For PGA<0.3g, the effects of column and beam parameters are negligible. Although damping should be an effective parameter on damage, Figure 4.26 shows that damping has effected the damage relatively small. The mass of the structure seems to be an important parameter for PGA>0.2g. The additional mass caused more damage when PGA>0.2g. The effect of the CSP is very negligible for all PGA values. The structure should be examined for larger PGA values since a peak is observed for PGA=0.4g.

4.7.5 FR3

Effect of CDP, CSP, DR, MP on damage increases with increasing PGA especially for PGA>0.2g. The current design of FR3 seems to be very economical. Effect of DR and MP is more clear in this structure. Figures 4.32, 4.33 show us that beam properties do not have a direct effect on damage, which is concluded in 4.7.1 also. The effect of beams can be considered as the effect of ratio of column strength to beam strength or the occurrence of plastic hinges during the loading. This should be investigated further.

Effect of the fundamental period and tip deflection of the structure is very clear with the shown second order trend lines.

CHAPTER 5

NEURAL NETWORK IMPLEMENTATION OF DAMAGE ANALYSIS

5.1 General

In this chapter the data obtained in the previous study is used to train multilayer networks using standard back-propagation algorithm. This part of the study includes two types of implementation of neural networks. In the first part, neural network is employed as a nonlinear interpolator to determine the damage indices. For this purpose two training sets are composed. First set is composed of 84 input-output data while the second set is composed of 445 data. These sets are obtained from the analysis of the frame FR3. Two different networks are obtained using these two sets and they are tested for the systems that are not included in the training sets. In the second part of the study, neural network is employed as classifier. For this purpose only the first training set is utilised and the network is tested for the second set. In the final part, how Kohonen networks can be used as a classifier, will be explained. For this part of study, software called DAMAGE written in Java 1.1.8 is used. Due to the nature of Java, first applets that run on Internet, second frames that use Java virtual machine are created.

5.2 Obtaining the Training and Test Sets

As stated in Chapter 3, the efficiency of training set is very important for the performance of neural network. It should be broad and the parameters should define the physical system as efficiently as possible. With this respect the following terms are considered as the inputs for the network.

Peak Ground Acceleration, PGA: this is the most important parameter that effects the damage

Peak Ground Velocity, PGV and Peak Ground Distance, PGD: These parameters are used as the earthquake parameters but since they change with PGA, the effect of these parameters is not expected to be visualised explicitly. One of the reasons for this poor design is the lack of analysis for different earthquakes.

V/A ratio and Effective Duration of the Earthquake, T_{eff}: As for PGA and PGD, these parameters are not studied successfully.

(The above parameters are earthquake characteristics. To study on the parameters more successively more earthquake data should be used in the analysis stage.)

Fundamental Period of the Structure (T): This is the parameter that gives an idea about the flexibility of the structure and hence is very important.

CDP, CSP, BDP and BSP These parameters do not directly effect the damage but the behaviour of the structure (T), hence are used as parameters for the sets.

Damping: This parameter has a direct effect on the damage as an energy dissipater.

Ratio of maximum tip deflection to height of structure (TD/H): This parameter has been used as a measure on the damage, but to obtain this parameter a nonlinear analysis should be made. This is a conflict since the main aim of neural networks is to eliminate this nonlinear analysis stage. A more detailed discussion will be given later.

Ratio of column strength to beam strength (CS/BS): This parameter is added in the view of the results of Chapter 4. The yielding moment capacities are used for CS/BS.

In additional to the above list, several other parameters can be used. Parameters that can more efficiently identify the system may be used. But the aim of the study is to provide an understanding of neural networks and possible extrapolations about the future research.

The total number of analysis done on FR3 was 445. 84 of them are selected randomly to form the first training set. The remaining data formed the test set for the first network. The second training set is the whole 445 data. The test set of this network is formed by additional analysis of FR3.

The training sets should be processed before starting the training. In this process the inputs and the outputs are scaled according to maximum and minimum values of that parameter. The following relation is used for scaling

$$x_s = L + (H - L) \times \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} \tag{5.1}$$

where L and H are the lower and the upper limits for the scaled data, $x_{\max}$ and $x_{\min}$ are the maximum and the minimum values of the parameter, respectively. The maximum and minimum values of the parameters are given in Table 5.1. L and H values are given in Table 5.2.

Table 5.1 Maximum and minimum values of the parameters

Parameter	**Minimum**	**Maximum**
PGA(g)	0.2	0.8
PGV(m/s)	0.3921	2.2329
PGD(m)	0.092	1.0041
V/A(s)	0.1331	0.2846
T_{eff}(s)	6.26	24.87
T(s)	0.33	0.61
CDP(%)	70	130
CSP(%)	70	130
BDP(%)	60	130
BSP(%)	60	130
DR(% Cr)	2	8
MP(%)	70	130
TD/H	0.0014517	0.0336946
CS/BS	1.234	3.339
DI	0.032	0.766

Table 5.2 L and H values for the scaling process

Parameter	L	H
Input	-0.8	0.9
Output	0.032	1.0

The reason why the L and H parameters are chosen as in Table 5.2, is related with the choice of the activation function used in the network. Several tests are done to find to efficient activation function and the L and H parameters and the above values give better result with sigmoid activation function in this network. In contrast Molas and Yamazaki [4] and Nakamura et al [5] who suggested to employ tangential sigmoid activation function, sigmoid activation function with a constant is used. This pre-tests show that different networks should be designed for different problems.

5.3 First Neural Network: Damage Indices Approximation

The models suggested by the previous researchers is a multilayer network with one hidden layer (Molas and Yamazaki [4], Nakamura et al [5], Ghaboussi and Garrett [8]). They explained that two-hidden-layer network will not give additional improvement. Therefore the software DAMAGE was written for two-hidden-layer network to examine the performance of the network. Since the results were satisfactory, a network with one hidden layer was not tested. After some pre-tests a 15-9-4-1 network was decided to be used for the first part (NN-1a). During the training process the learning rate is changed adaptively but not as described in Chapter 3. A new method will be proposed by the author to examine the effect of learning rate but not to obtain faster training. According to this new approach, it is assumed that the network has an optimum learning rate or generally a range of learning rate, which gives a good performance for all weights. It is a well-known fact that the performance of the algorithm increases if every weight has its own learning rate, which changes adaptively as mentioned before. This new method assumes that there may be a range of learning rate, which is applicable to all weights. Let us first explain the concept of range of learning rate (LR).

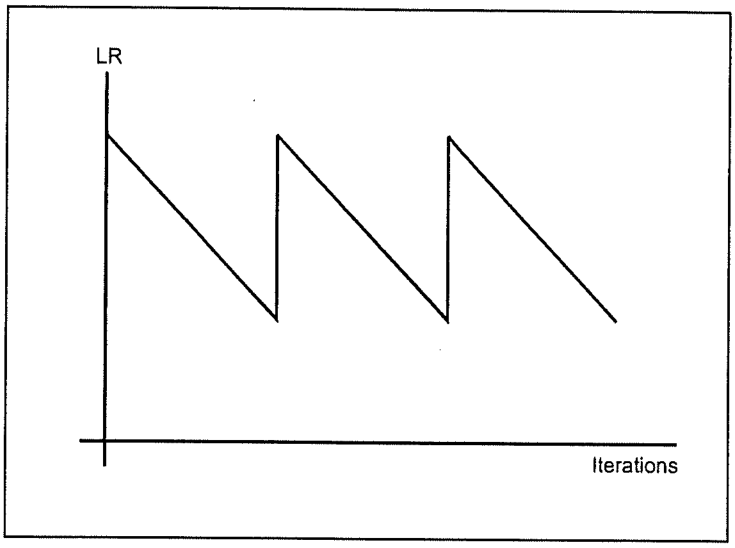

Figure 5.1 Change in LR

The approach is to change LR periodically as shown in Figure 5.1. Other suggestions can be done such as sinusoidal change, but the main idea should be a periodic change in LR. The LR range will be the maximum and the minimum value of LR throughout the iterations.

The LR range and the period of this change may be variable throughout the iterations (Figure 5.2). The proposed approach assumes that the user can change these values throughout the iterations. Throughout the training of the first neural network this approach was used. LR range is compressed and an optimal LR range is obtained as shown in Figure 5.3. For the first neural network with 84 training data, this range is about [0.0001, 0.0002]. Training the network with this type of learning rate results very slow learning but continuous decrease in maximum error. The network is trained, with a sigmoid constant of 3.5 and bias -1, until a maximum error of 0.0018 is reached. Then the network is tested with the testing set and the results are presented below.

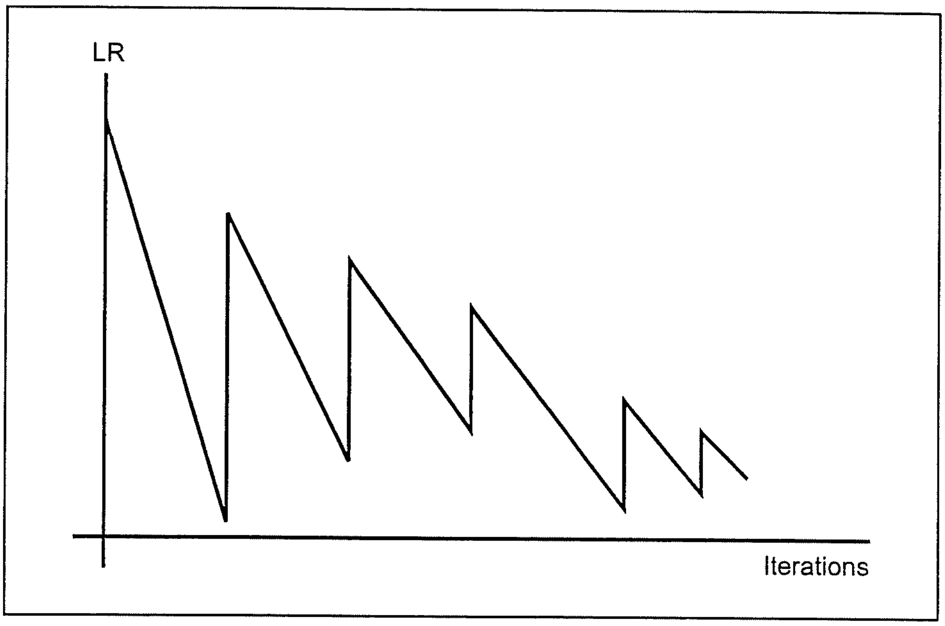

Figure 5.2 Variable LR range and period

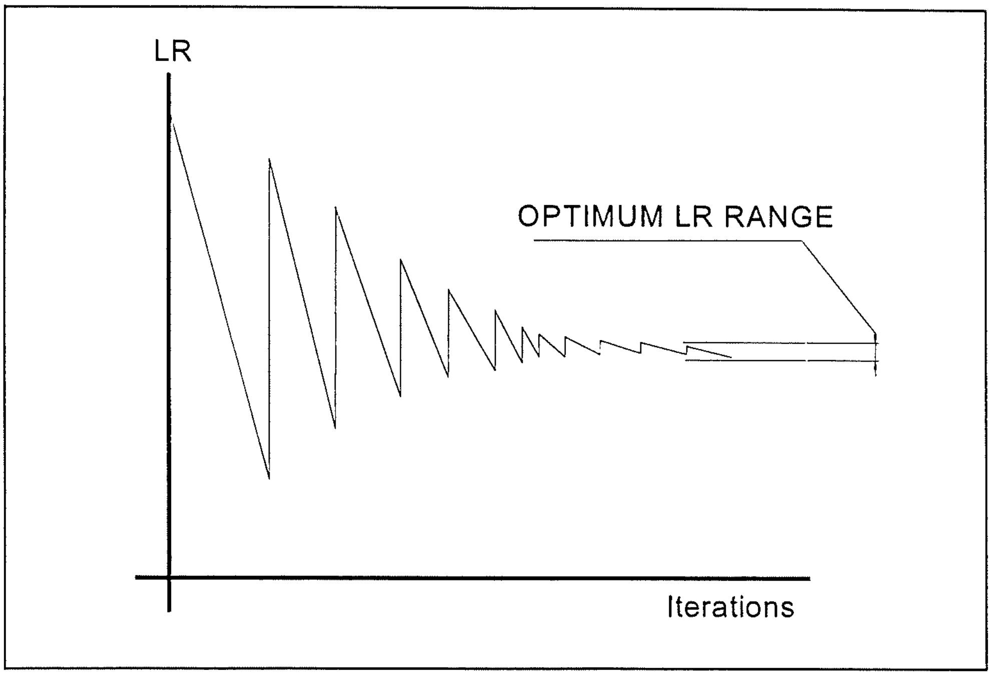

Figure 5.3 Definition of optimum LR range

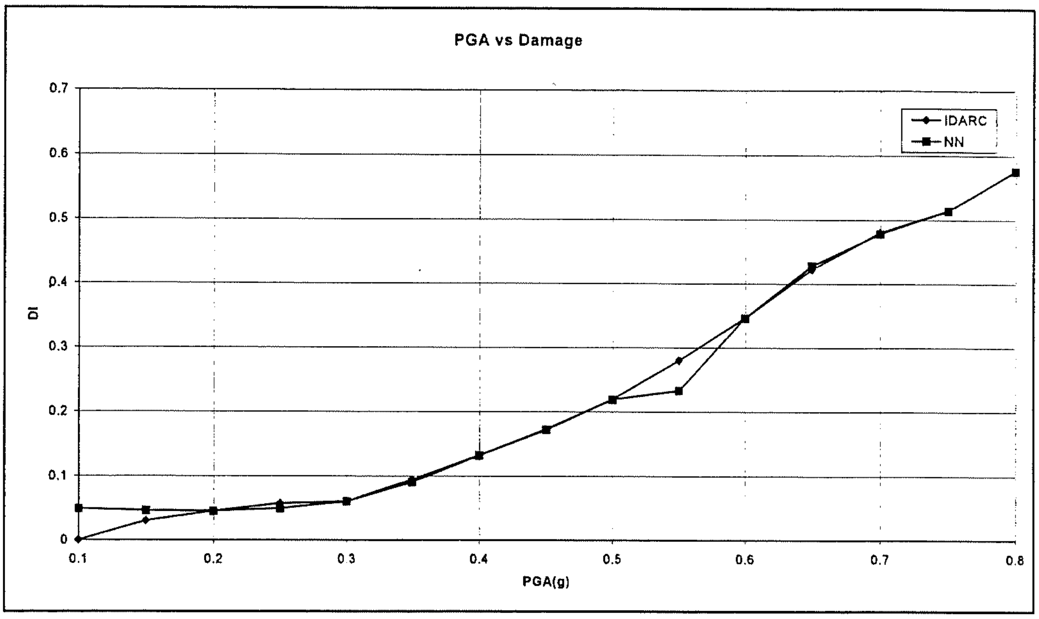

Figure 5.4 NN-1a implementation of effect of PGA on FR3

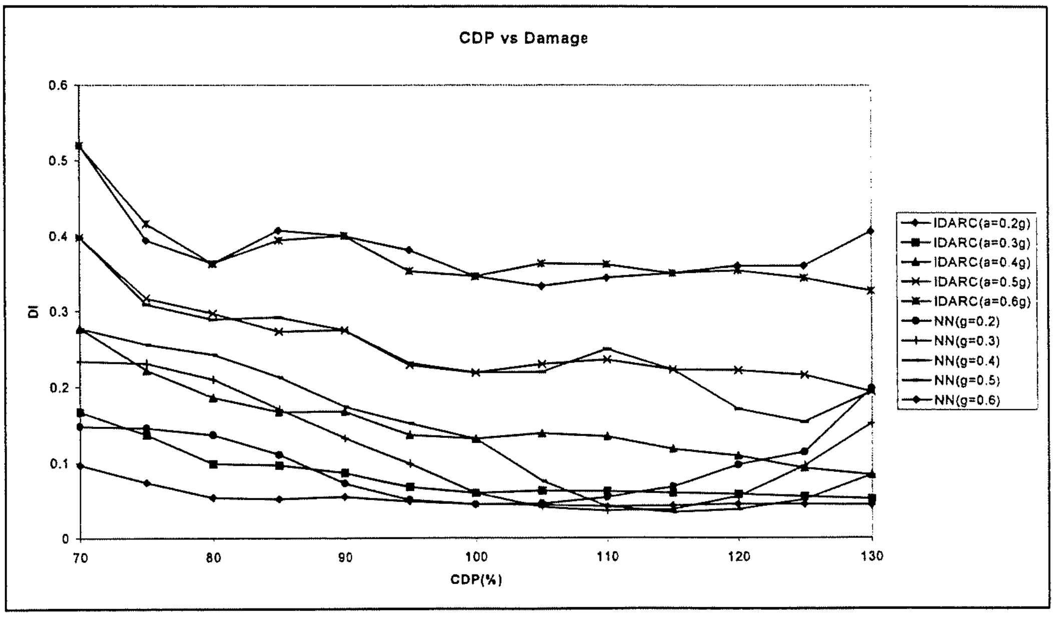

Figure 5.5 NN-1a implementation of effect of CDP on FR3

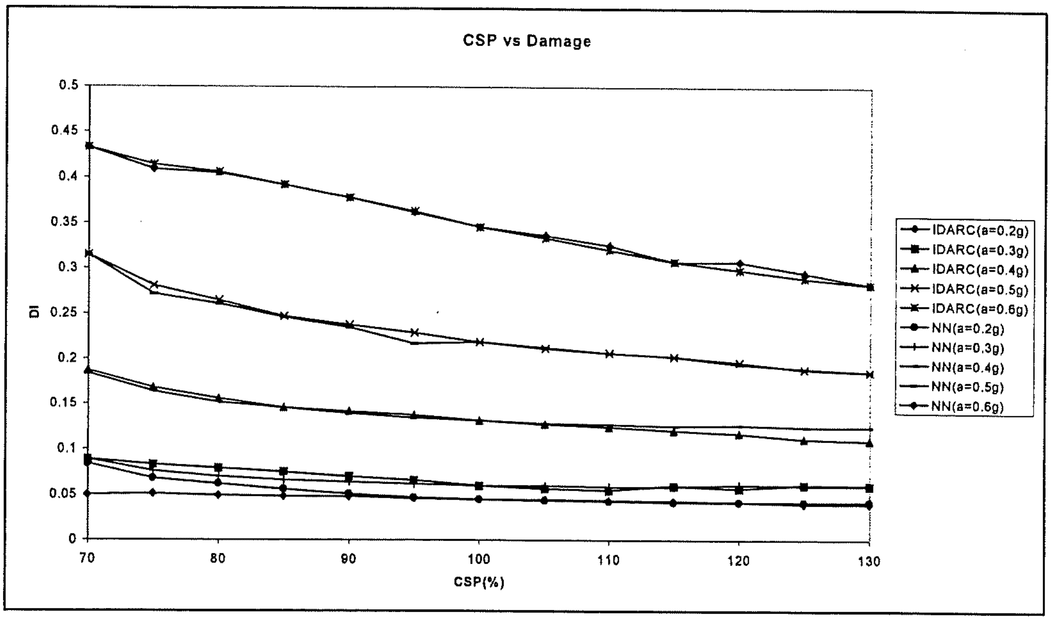

Figure 5.6 NN-1a implementation of effect of CSP on FR3

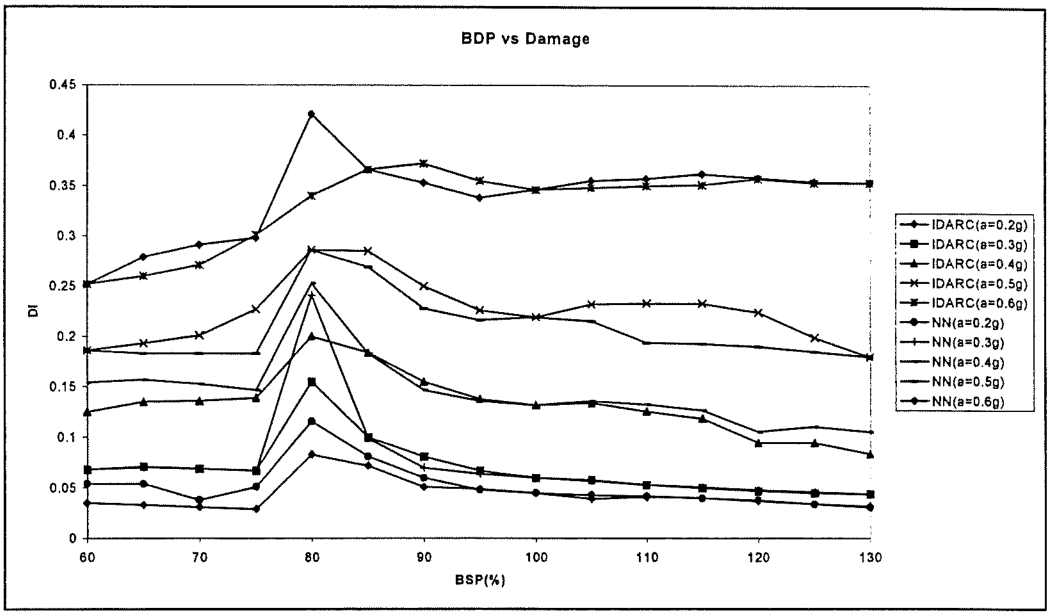

Figure 5.7 NN-1a implementation of effect of BDP on FR3

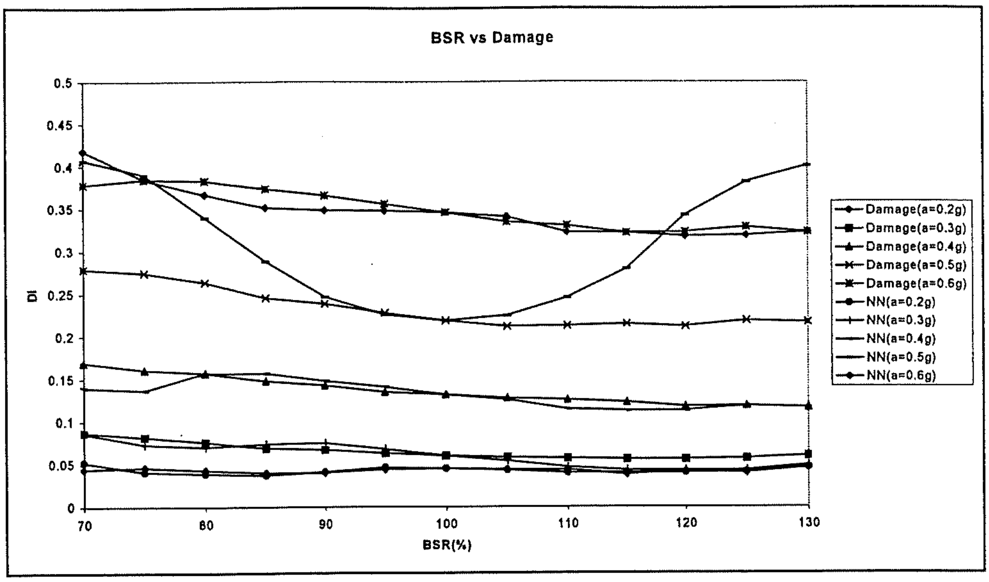

Figure 5.8 NN-1a implementation of effect of BSP on FR3

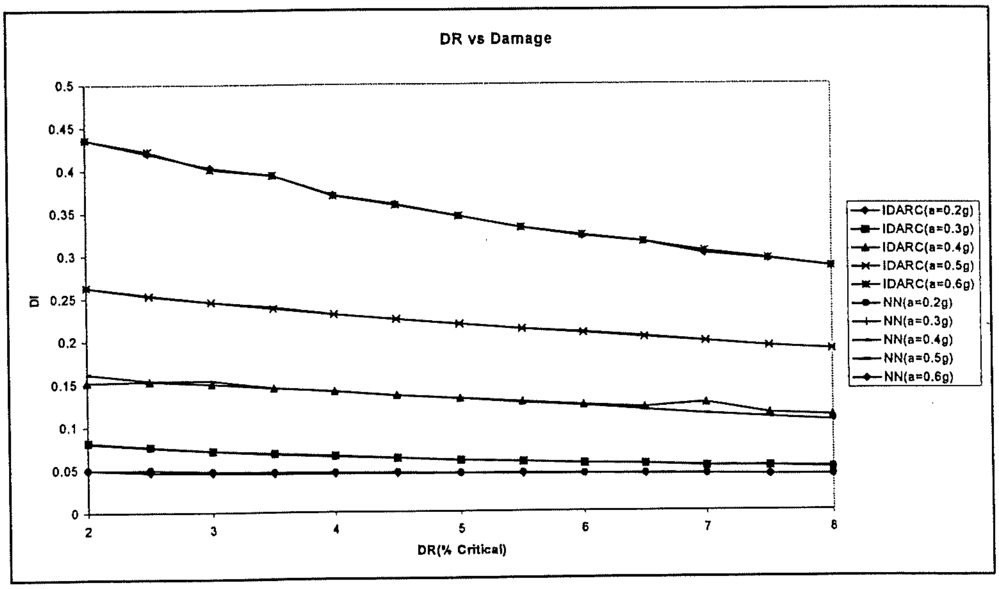

Figure 5.9 NN-1a implementation of effect of DR on FR3

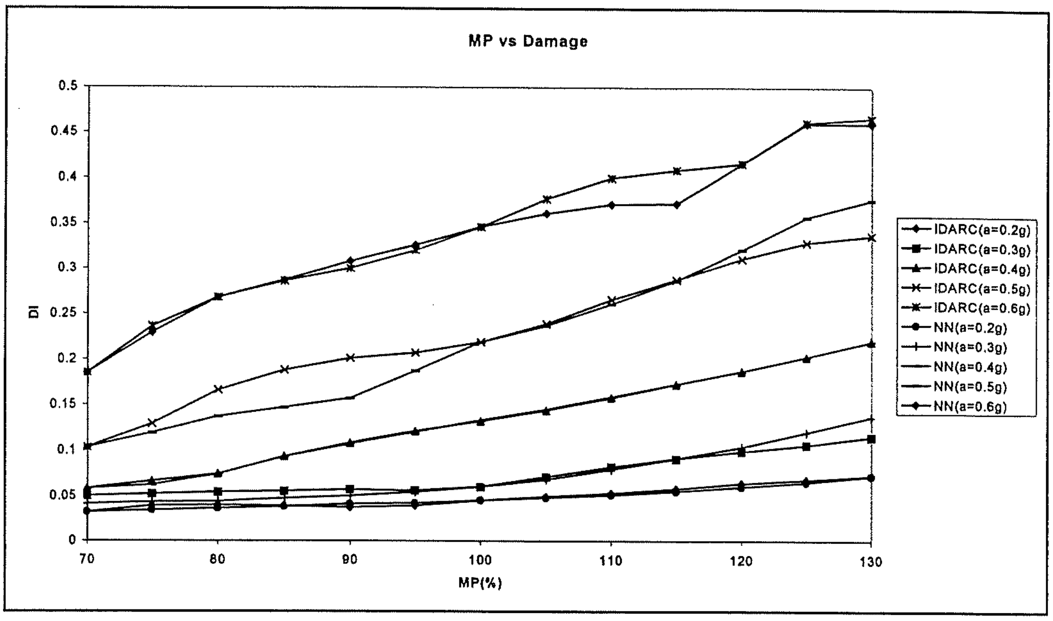

Figure 5.10 NN-1a implementation of effect of MP on FR3

The 15-9-4-1 network is trained for the second training set, which consists of 445 data, but the learning performance was not so efficient. After some tests for the second training set it is decided to use a 15-11-6-1 network with a sigmoid constant -3.5 and a bias -1 for all neurons (NN-1b). LR range came out to be [0.35, 0.42]. Since the whole data obtained from the first analysis of FR3 is used, additional analyses are done for the modified versions of FR3.

Table 5.3 Modified Frames

Frame	CDP(%)	CSP(%)	BDP(%)	BSP(%)	DR(%)	MP(%)
MOD1	120	120	120	120	2	120
MOD2	80	80	80	80	7	80
MOD4	120	120	80	80	3	120

The results are given below. The graphs presented in Figure 5.11 to 5.17 are the results for the training set, they are expected to be very efficient. This successful implementation does not mean that the network will be efficient for the test set. These graphs are presented here merely to show the behaviour of neural network.

The graphs presented by Figure 5.18 to Figure 5.20 are the results for the test set: MOD1, MOD2 and MOD3, respectively. The discussions of these results will be done at the end of this chapter.

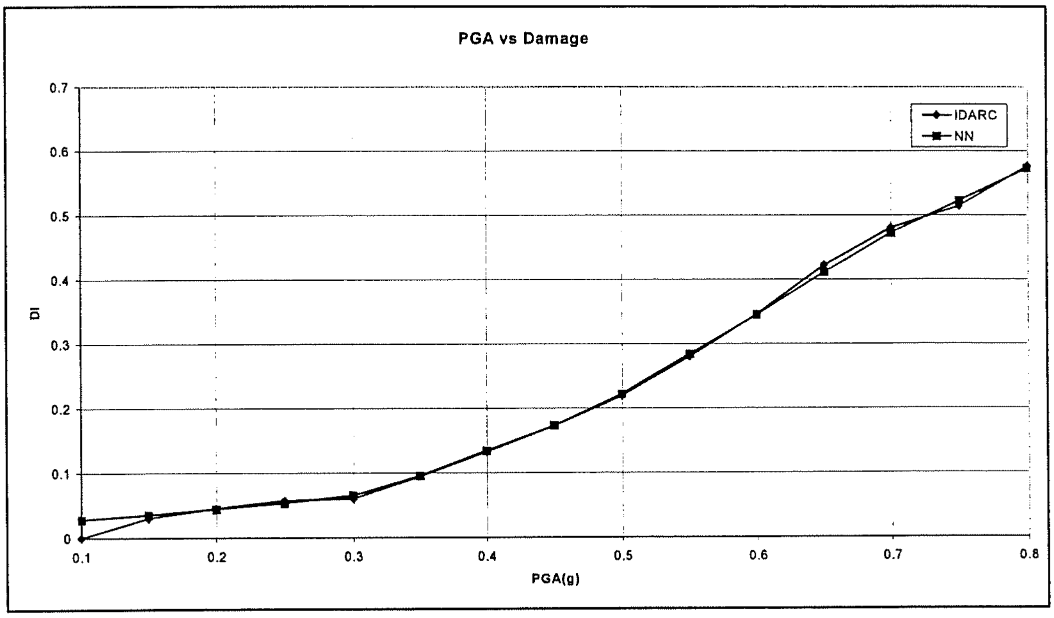

Figure 5.11 NN-1b implementation of effect of PGA on FR3

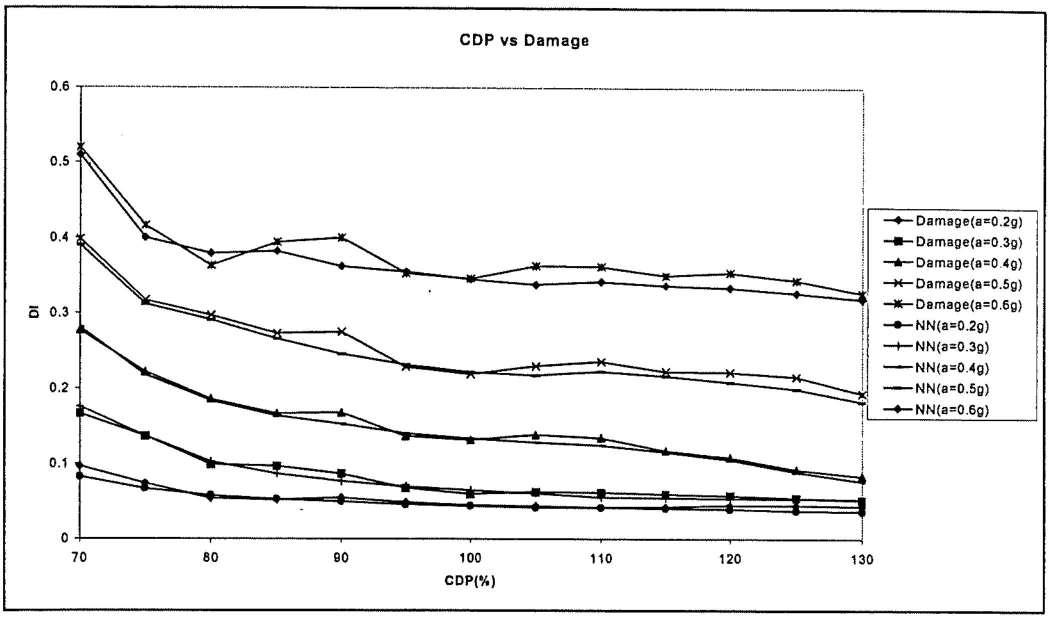

Figure 5.12 NN-1b implementation of effect of CDP on FR3

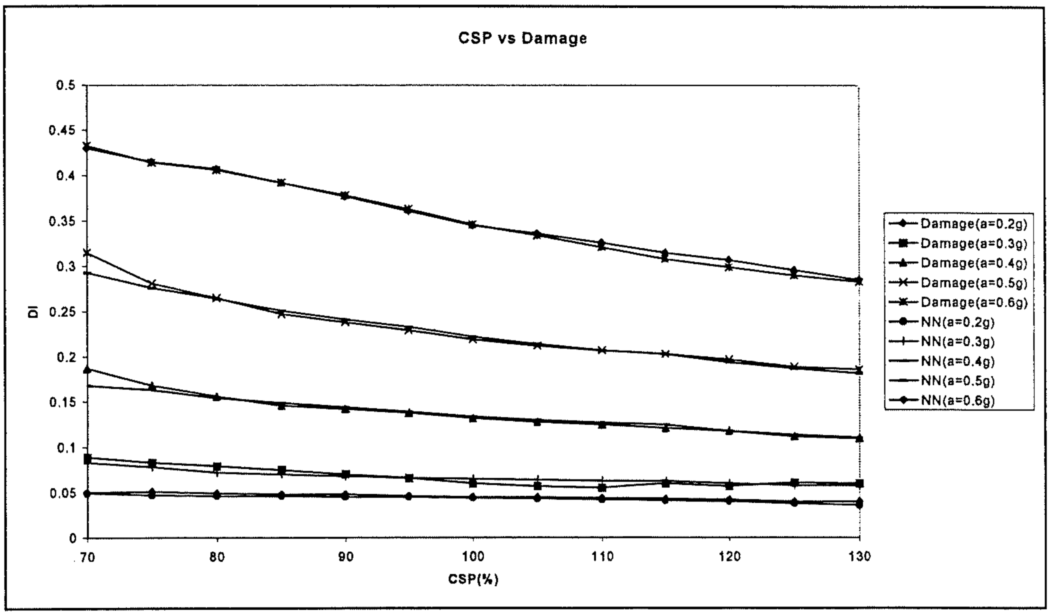

Figure 5.13 NN-1b implementation of effect of CSP on FR3

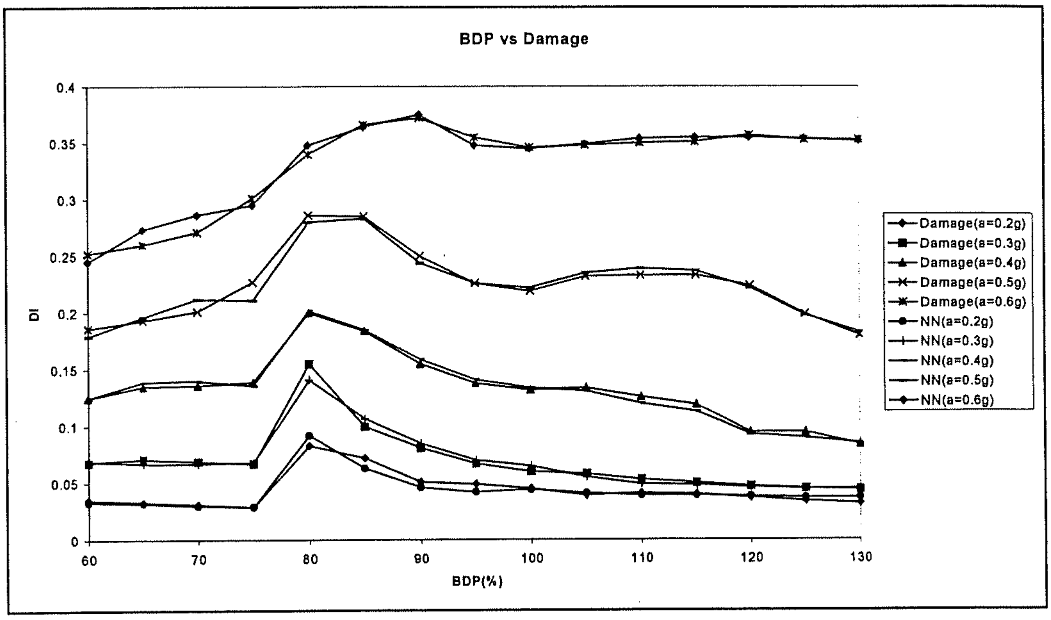

Figure 5.14 NN-1b implementation of effect of BDP on FR3

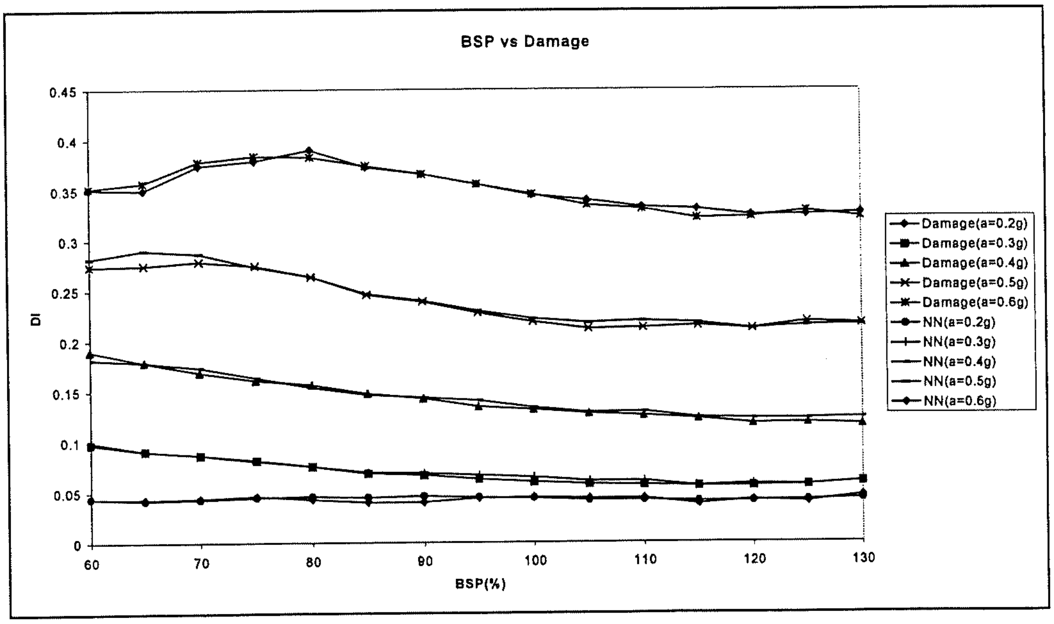

Figure 5.15 NN-1b implementation of effect of BSP on FR3

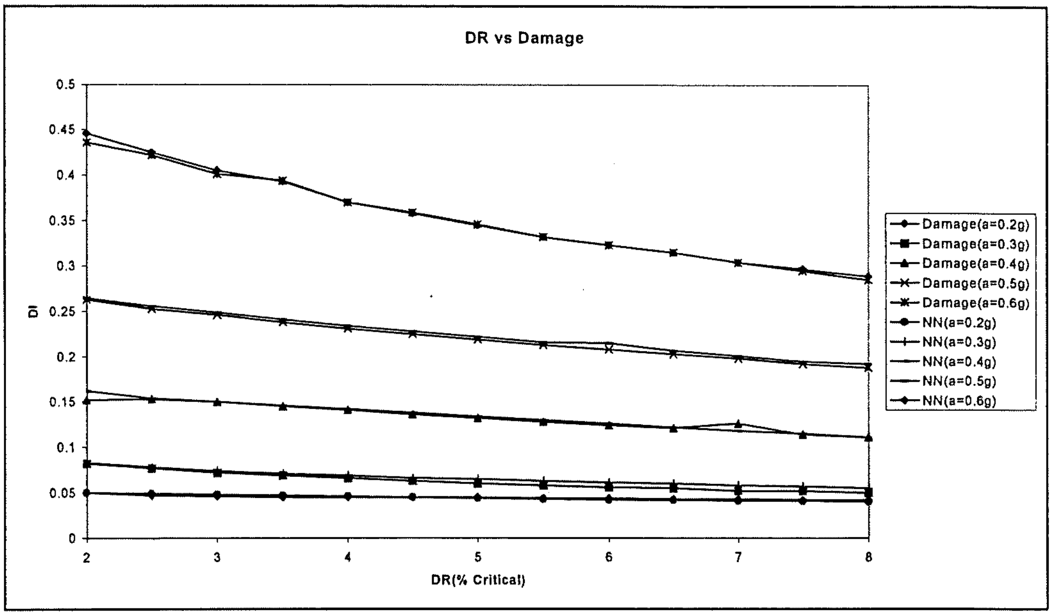

Figure 5.16 NN-1b implementation of effect of DR on FR3

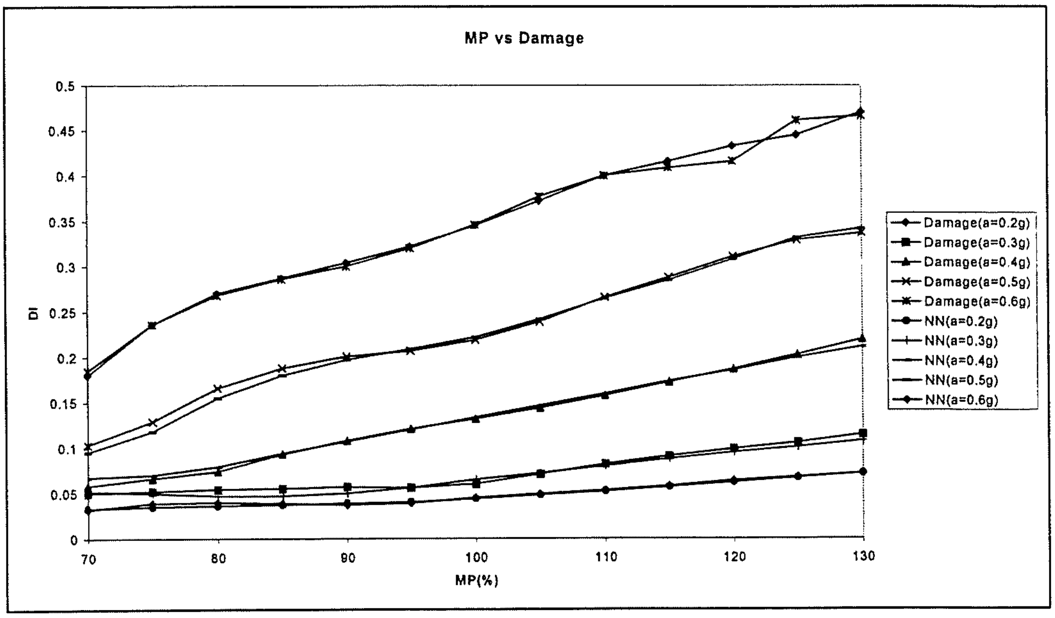

Figure 5.17 NN-1b implementation of effect of MP of FR3

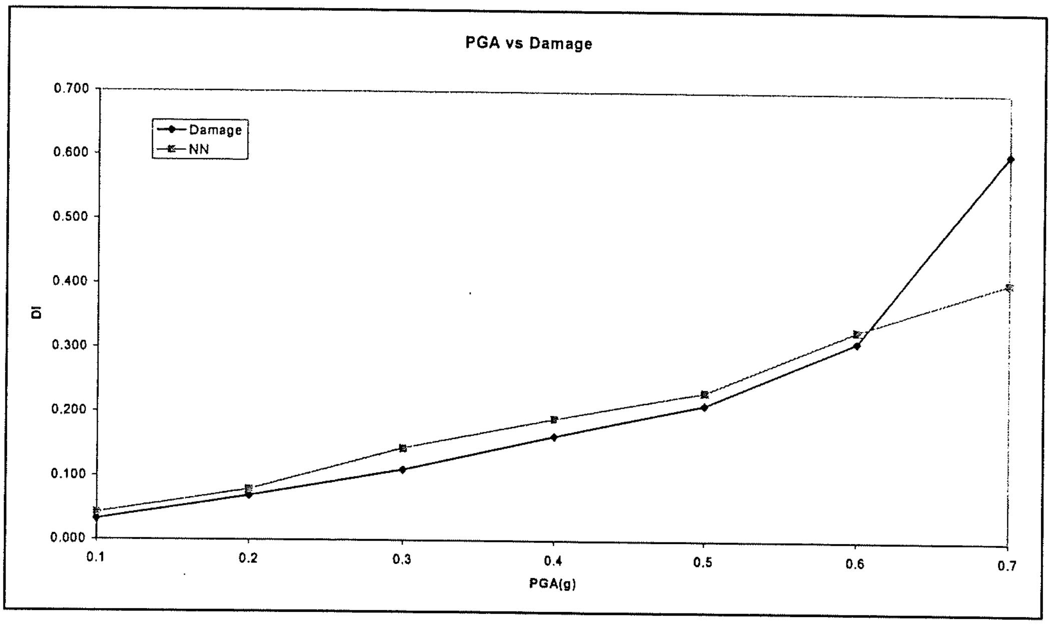

Figure 5.18 Effect of PGA on MOD1

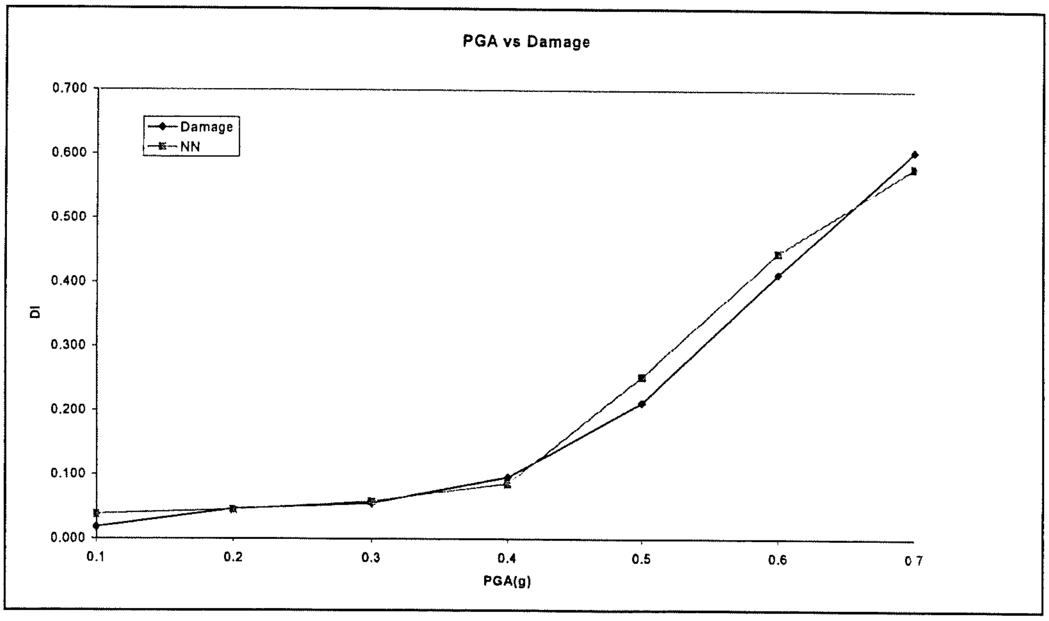

Figure 5.19 Effect of PGA on MOD2

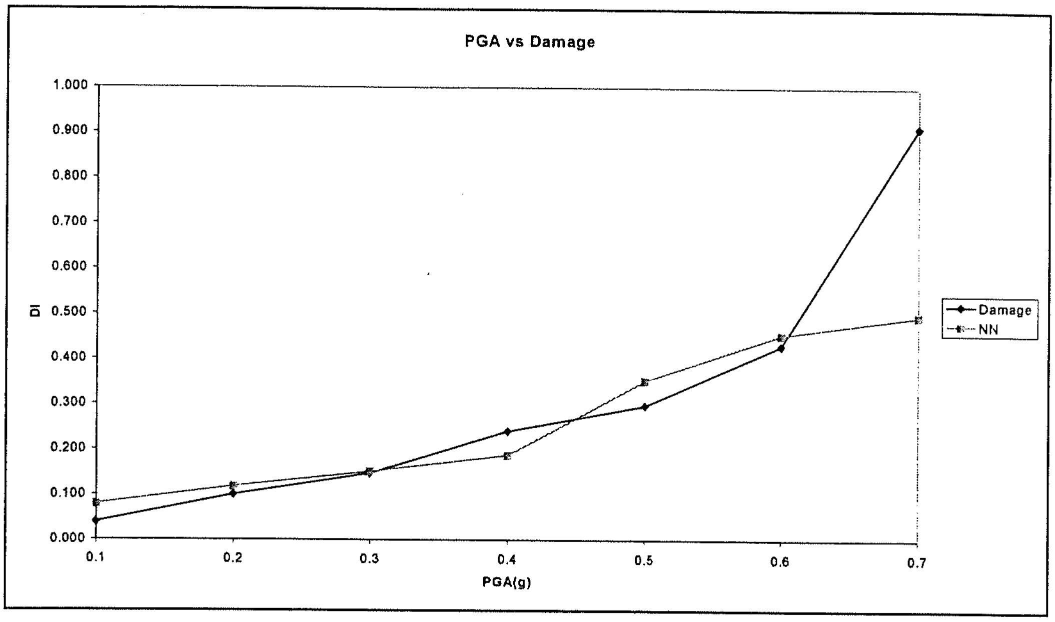

Figure 5.20 Effect of PGA on MOD3

In the above part of the study neural network served as a function whose parameters are unknown with fourteen inputs (Figure 5.21). To visualise how the behaviour of this function, 856 additional runs were made. In these runs the effect of every parameter is examined by changing the value of that parameter and fixing the other parameters for several values of PGA.

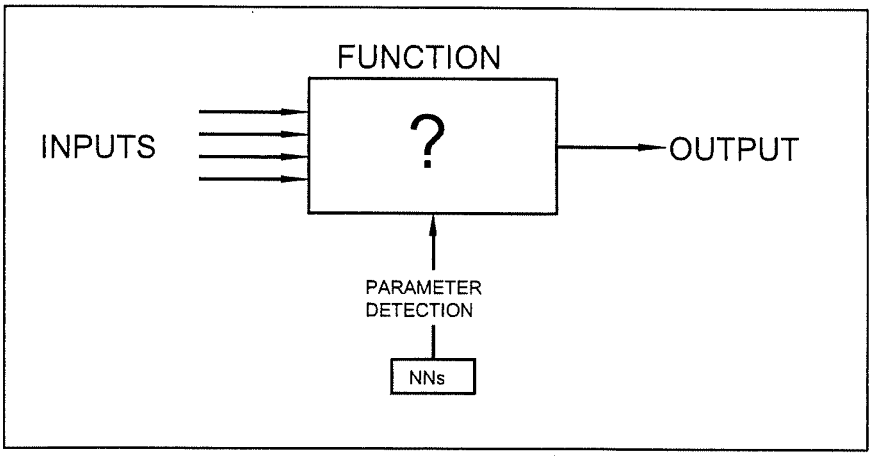

Figure 5.21 Representation of NN-1a and NN-1b

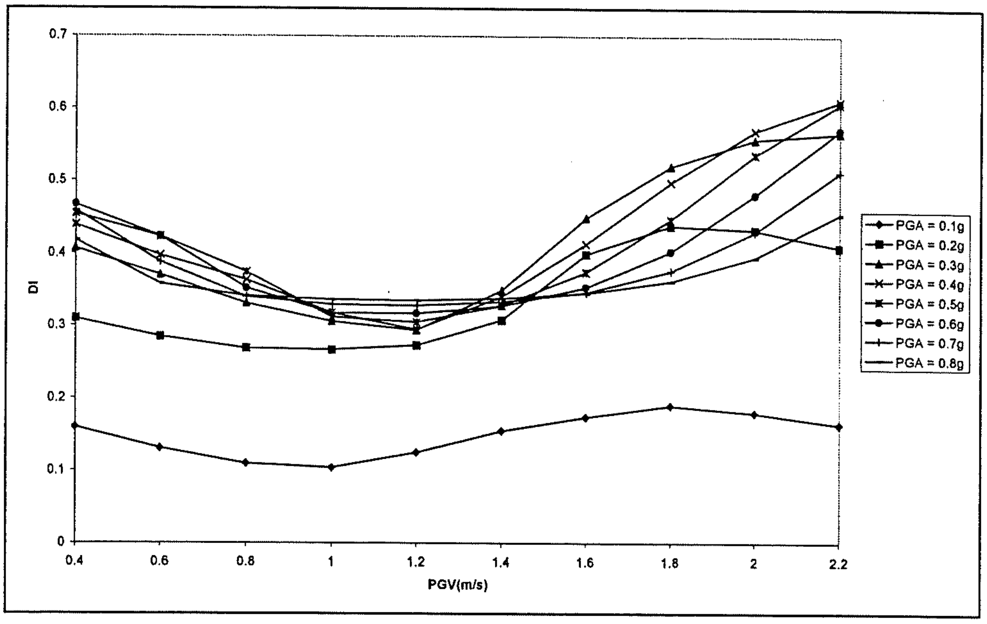

Figure 5.22 PGV variation of NN-1b

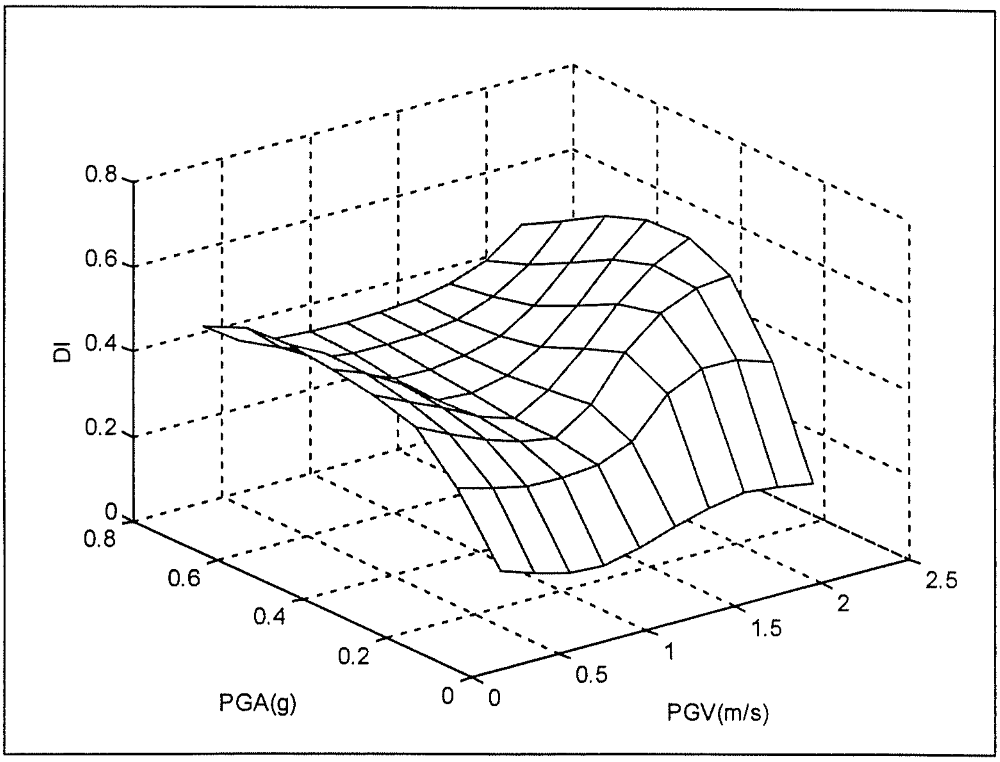

Figure 5.23 PGV variation of NN-1b (3D)

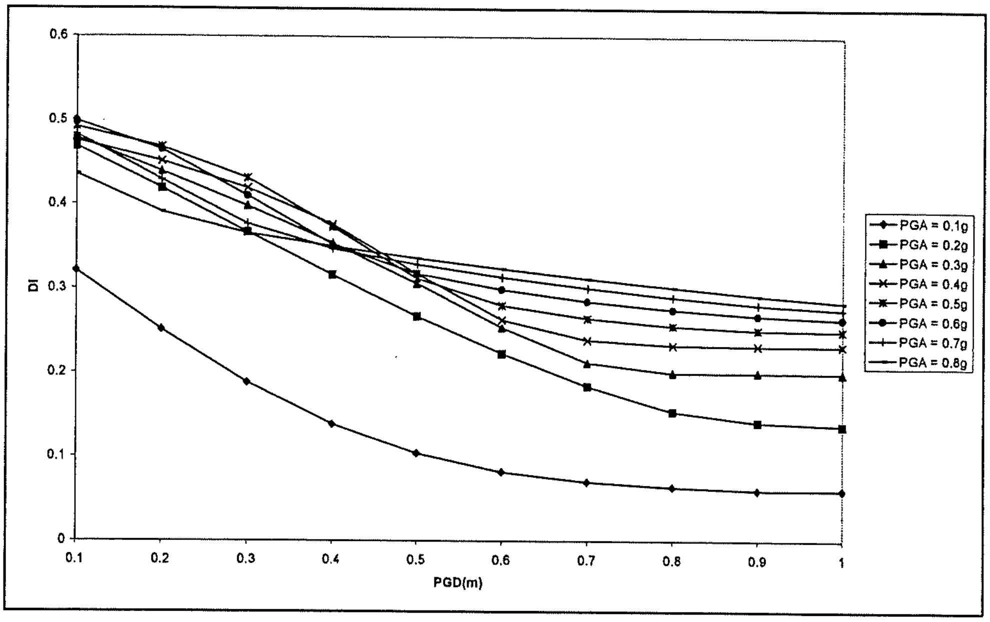

Figure 5.24 PGD variation of NN-1b

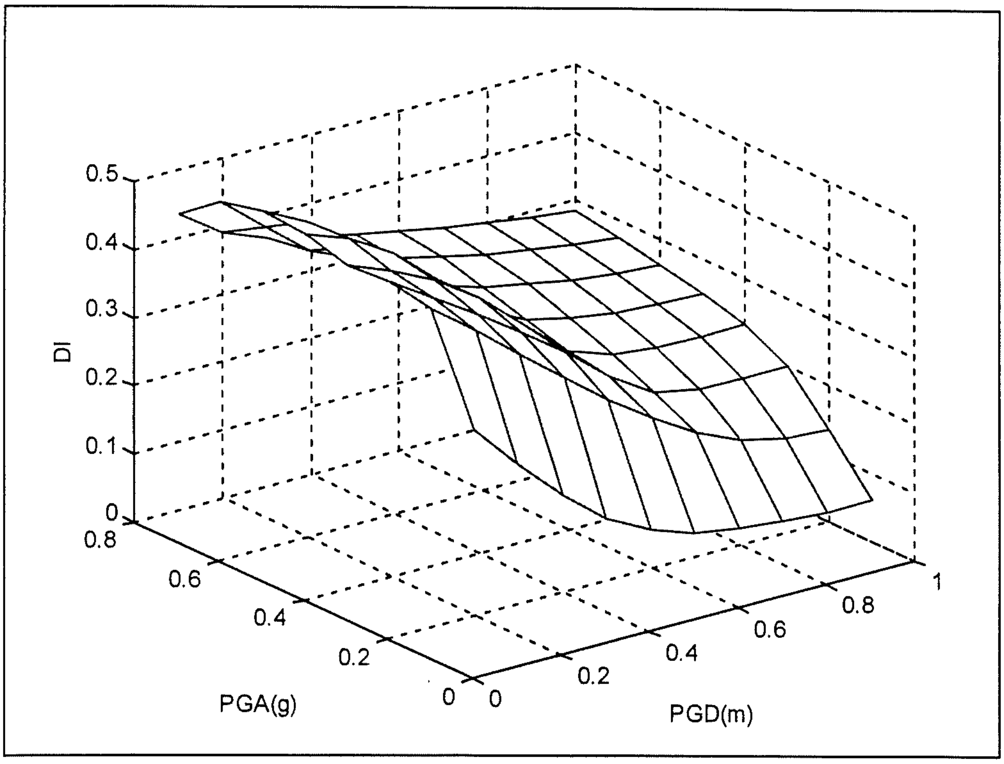

Figure 5.25 PGD variation of NN-1b (3D)

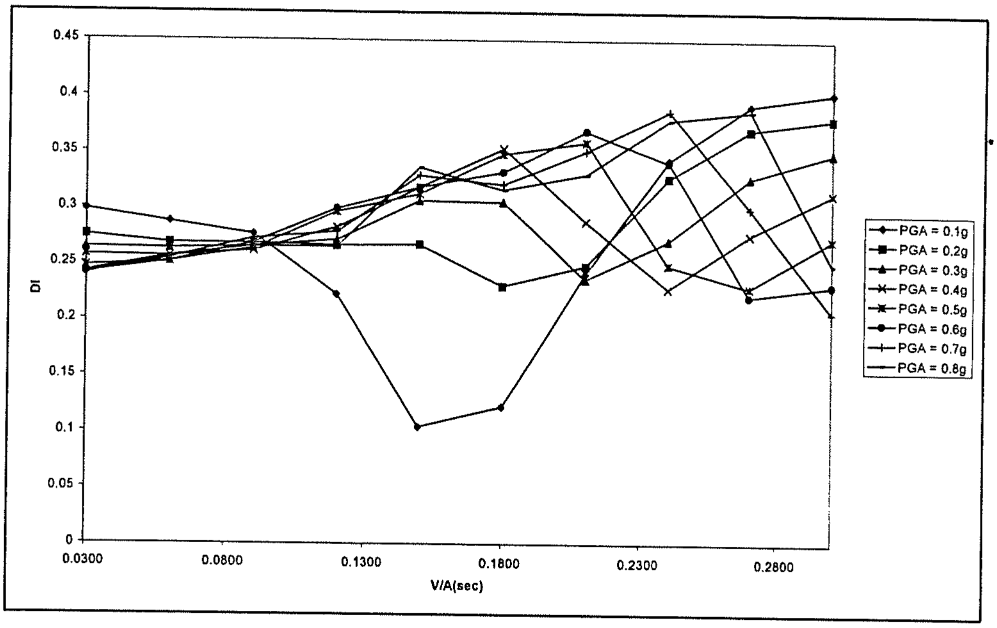

Figure 5.26 V/A variation of NN-1b

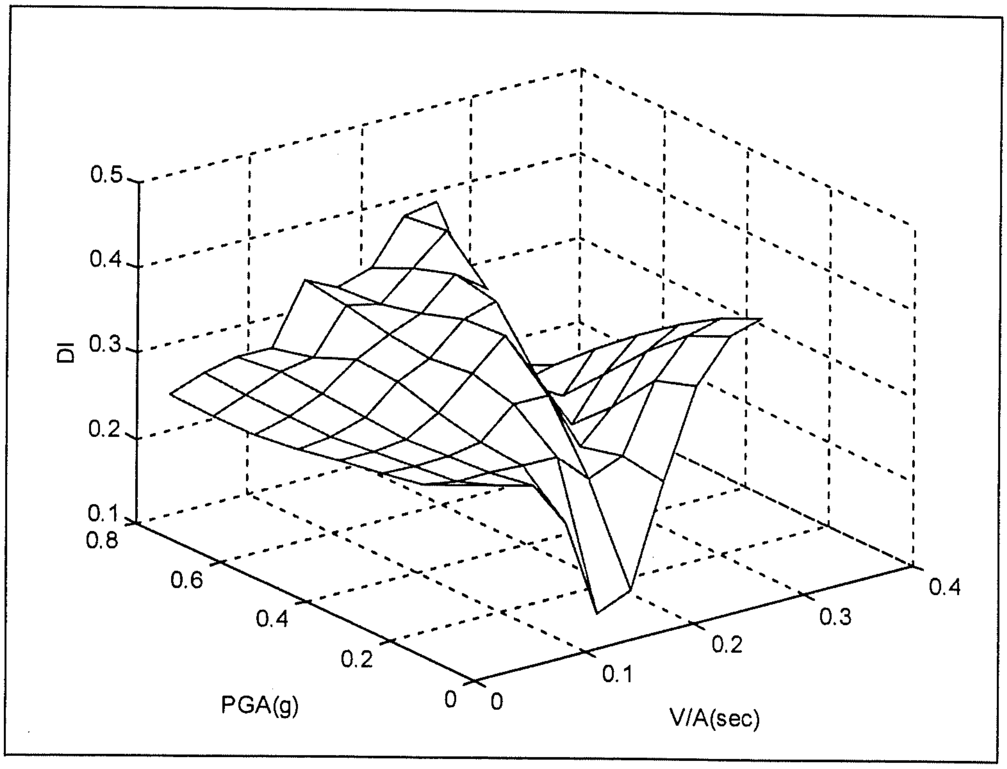

Figure 5.27 V/A variation of NN-1b (3D)

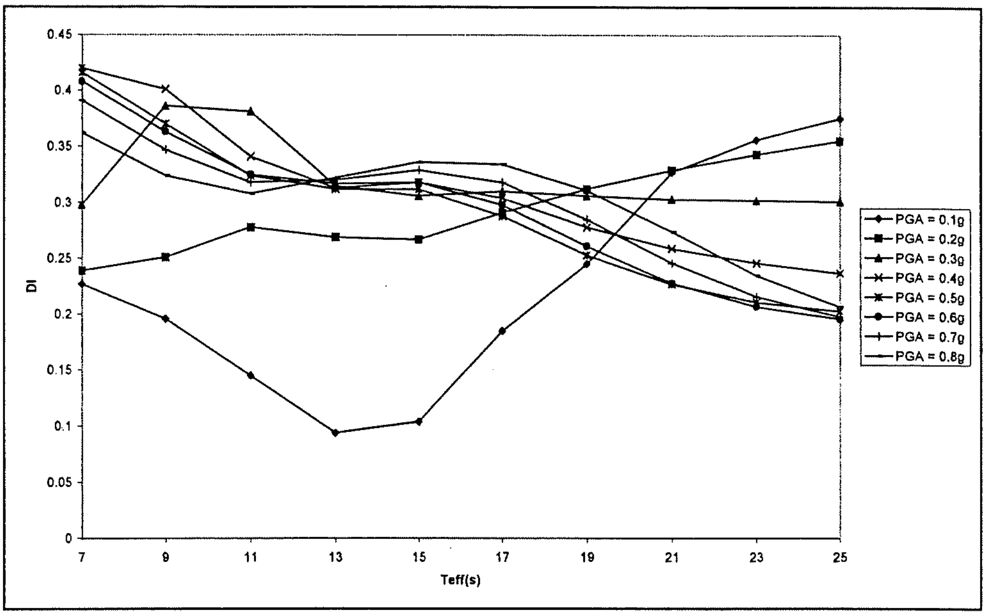

Figure 5.28 T$_{eff}$ variation of NN-1b

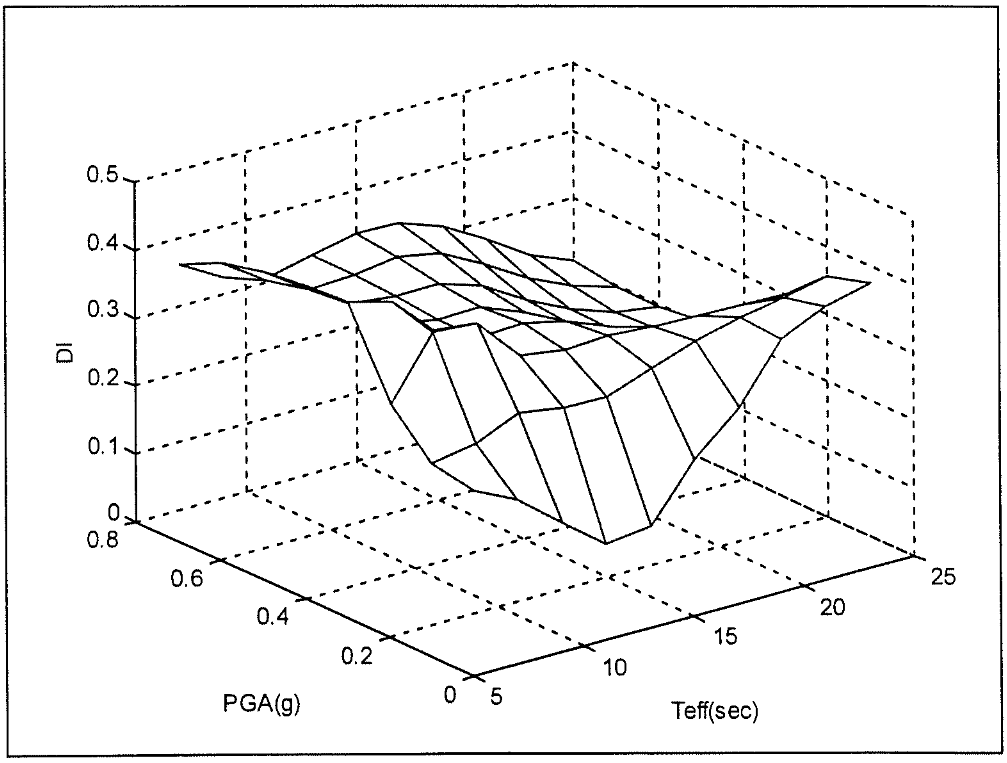

Figure 5.29 T$_{eff}$ variation of NN-1b (3D)

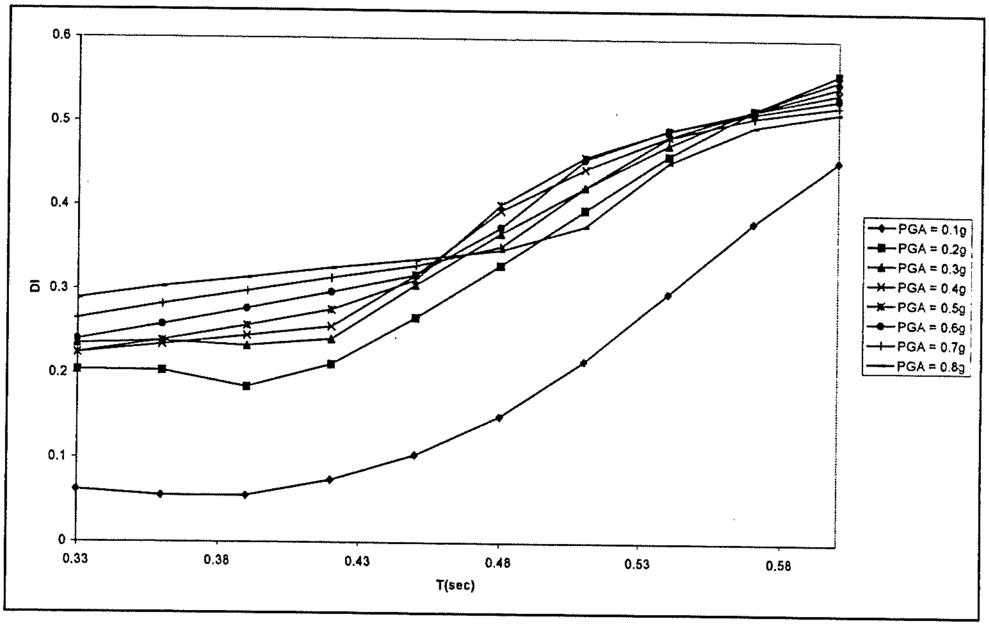

Figure 5.30 T variation of NN-1b

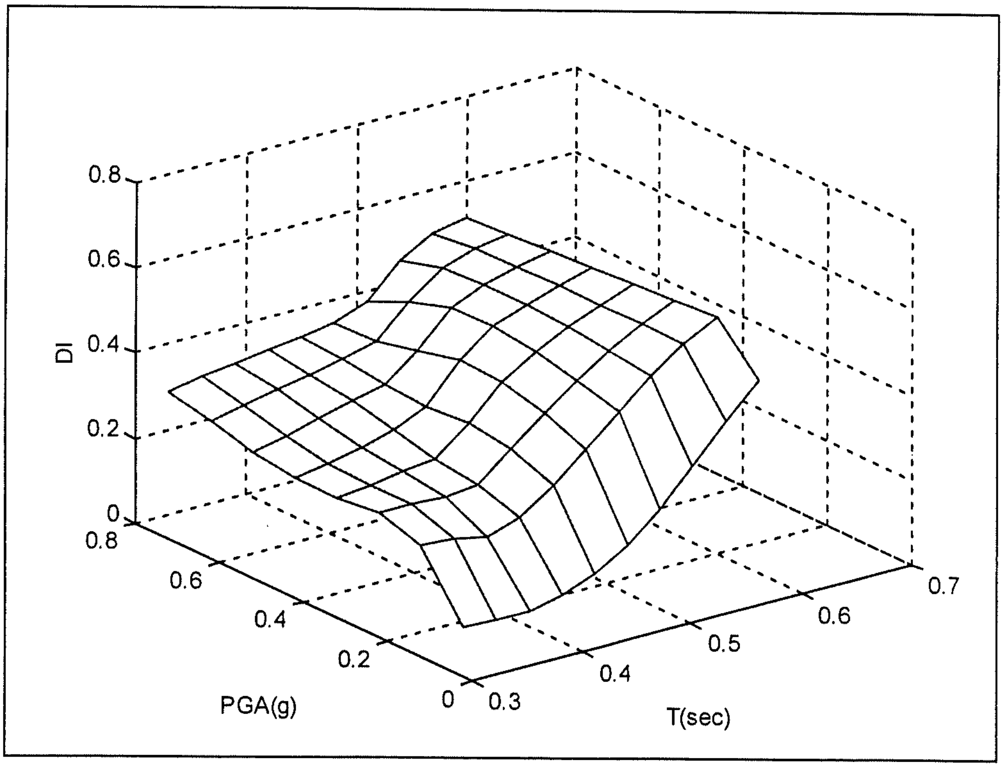

Figure 5.31 T variation of NN-1b (3D)

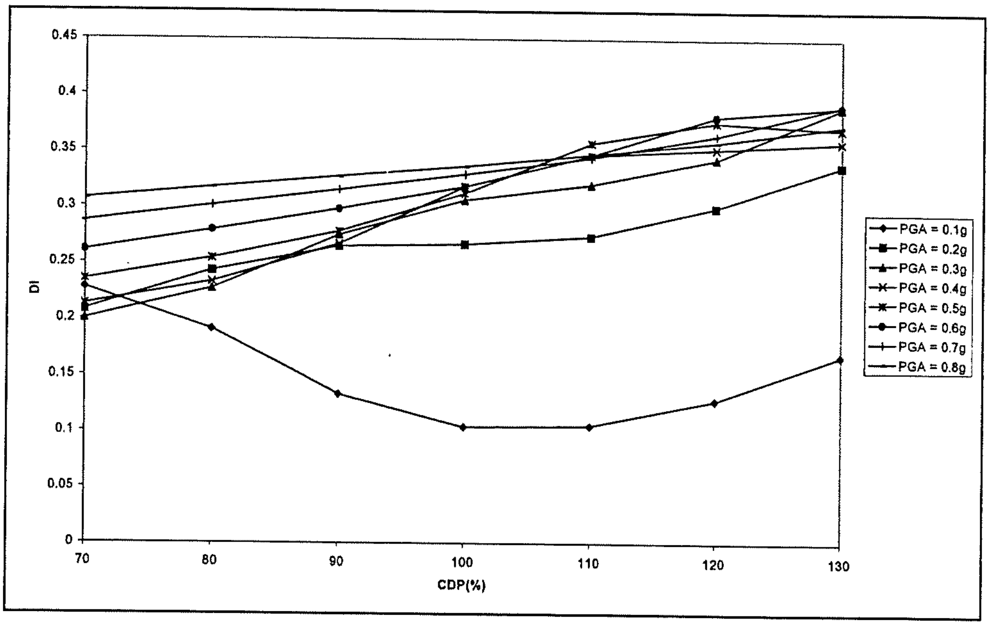

Figure 5.32 CDP variation of NN-1b

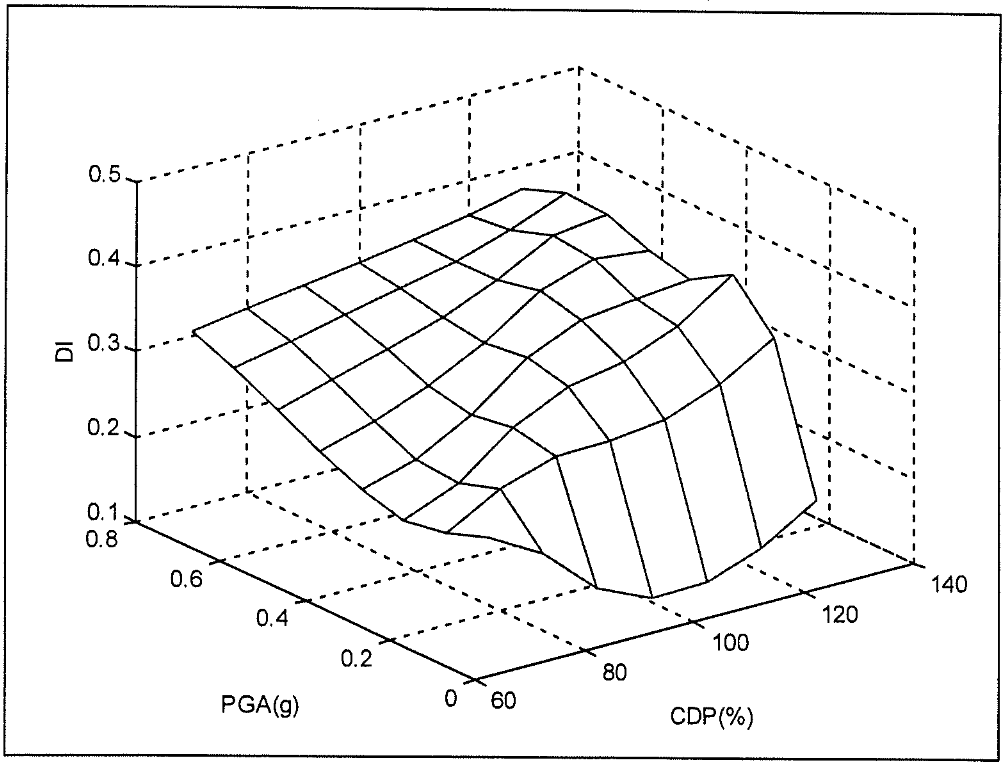

Figure 5.33 CDP variation of NN-1b

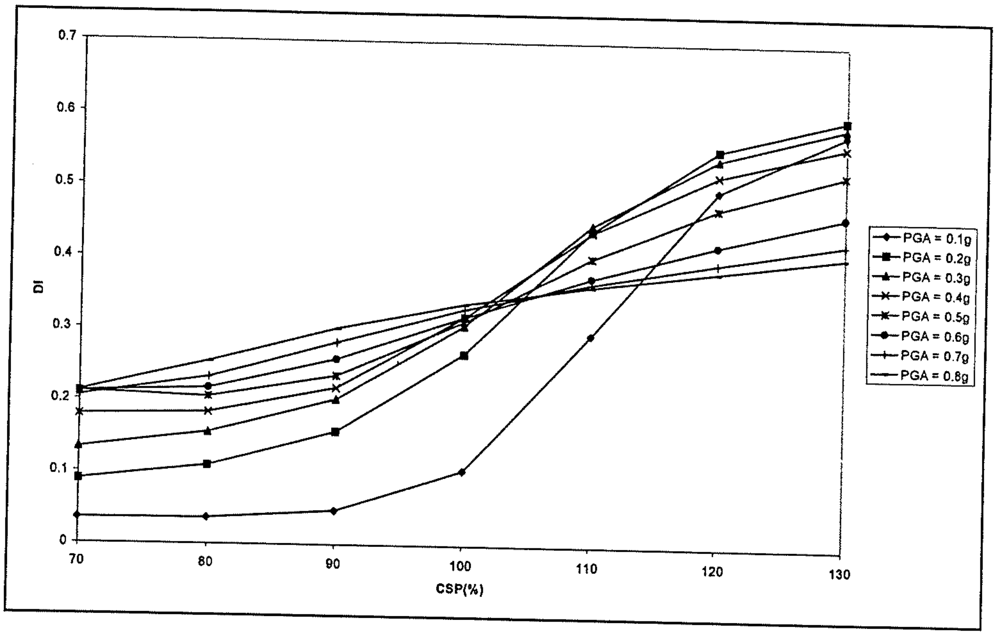

Figure 5.34 CSP variation of NN-1b

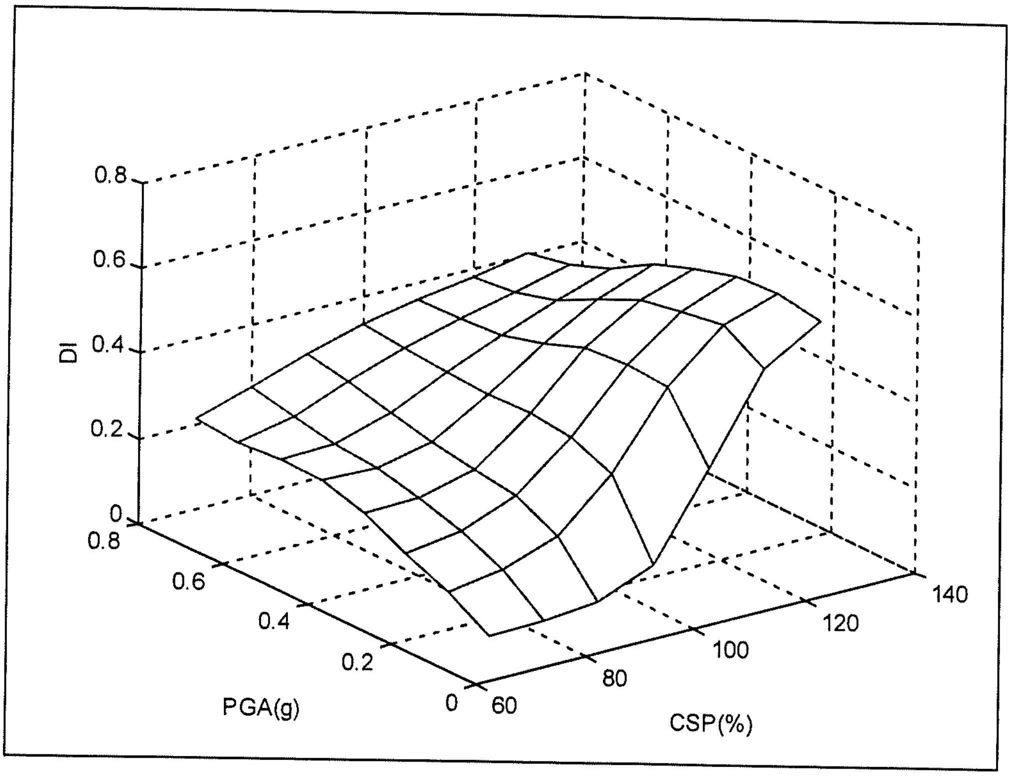

Figure 5.35 CSP variation of NN-1b

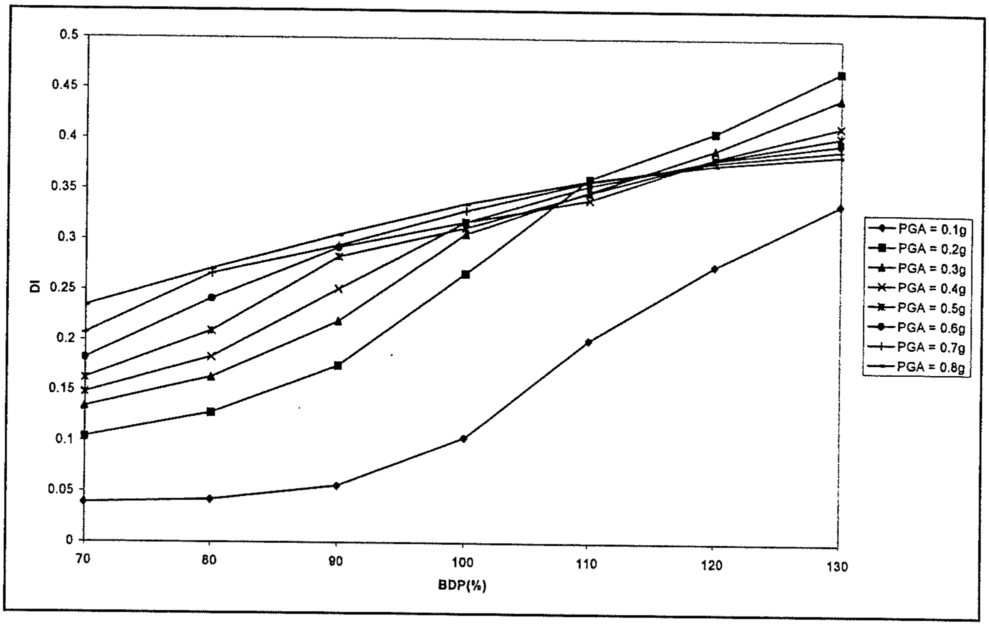

Figure 5.36 BDP variation of NN-1b

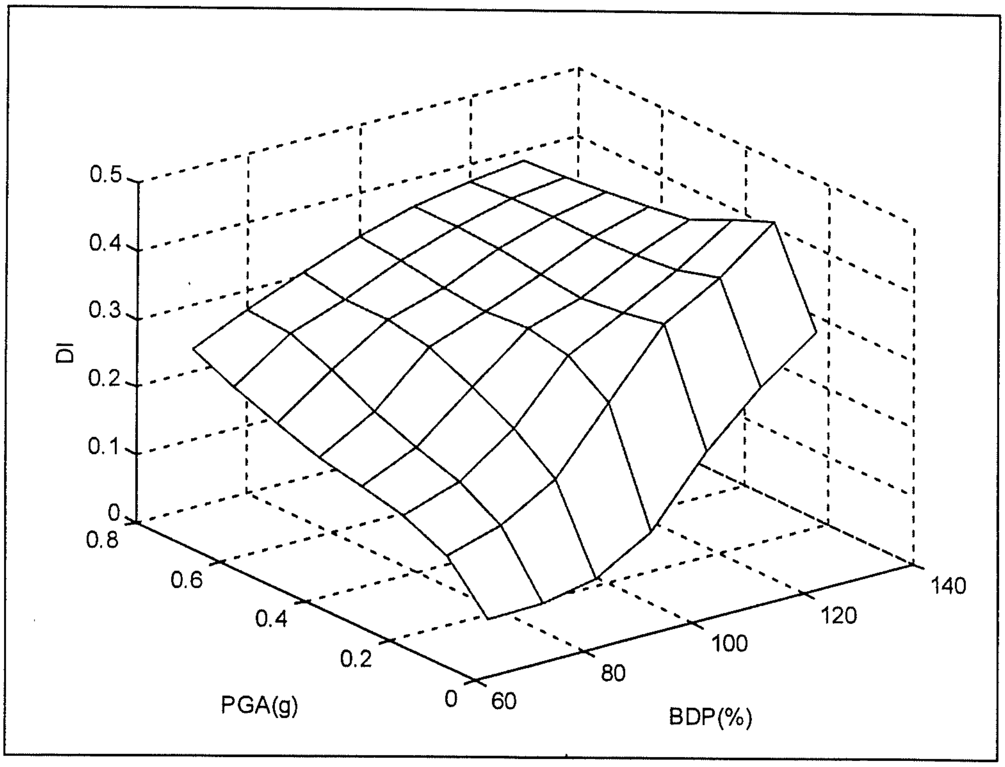

Figure 5.37 BDP variation of NN-1b (3D)

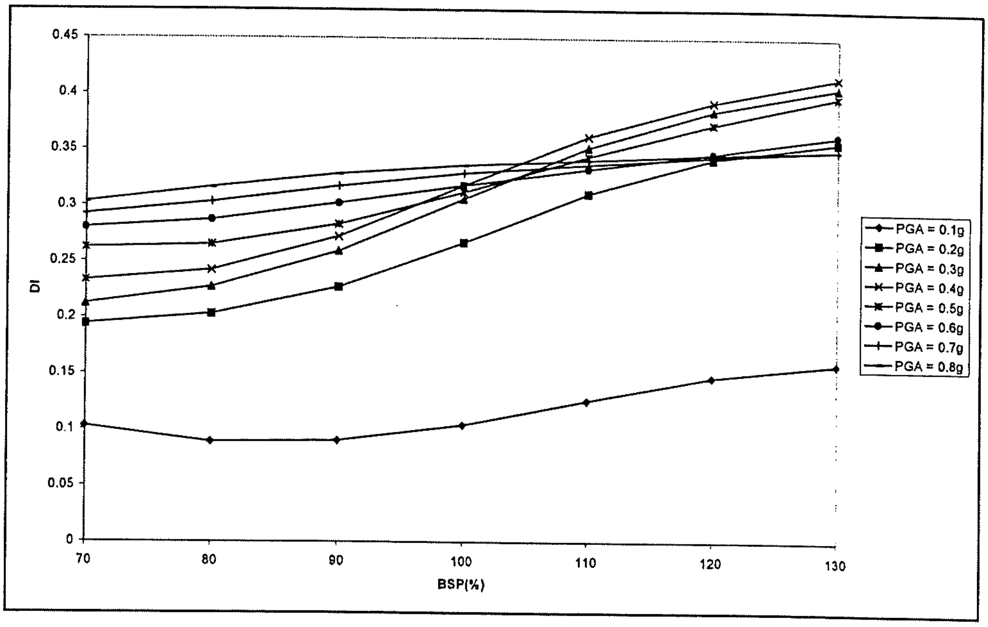

Figure 5.38 BSP variation of NN-1b

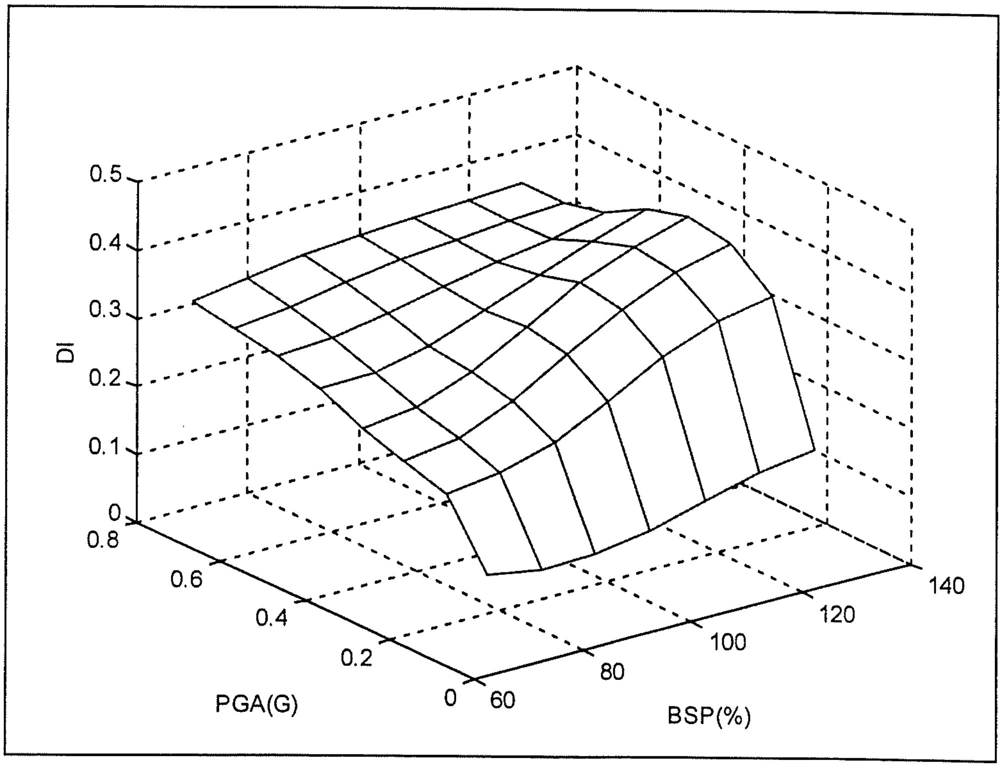

Figure 5.39 BSP variation of NN-1b (3D)

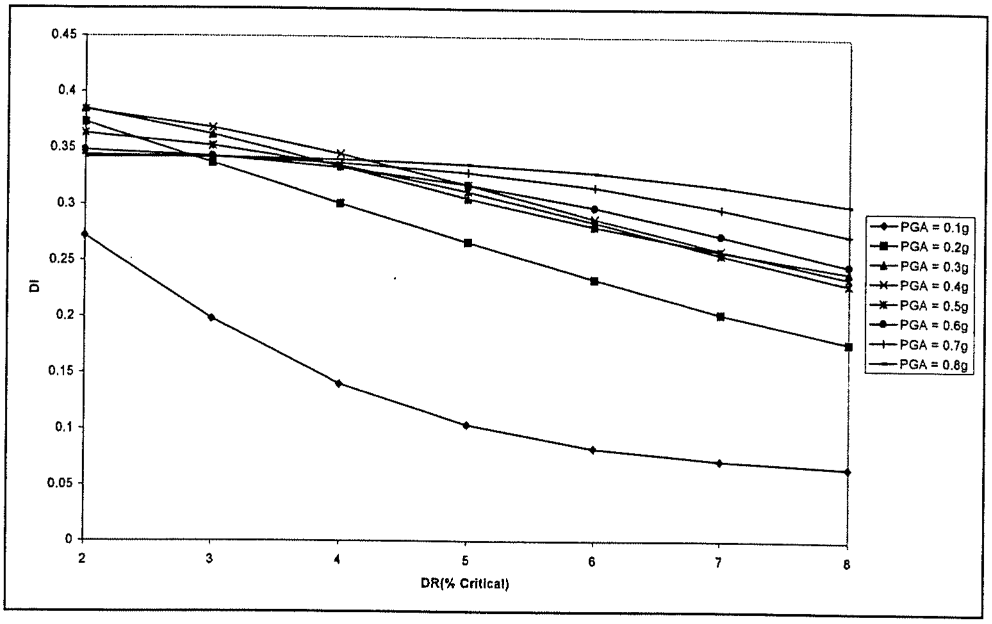

Figure 5.40 DR variation of NN-1b

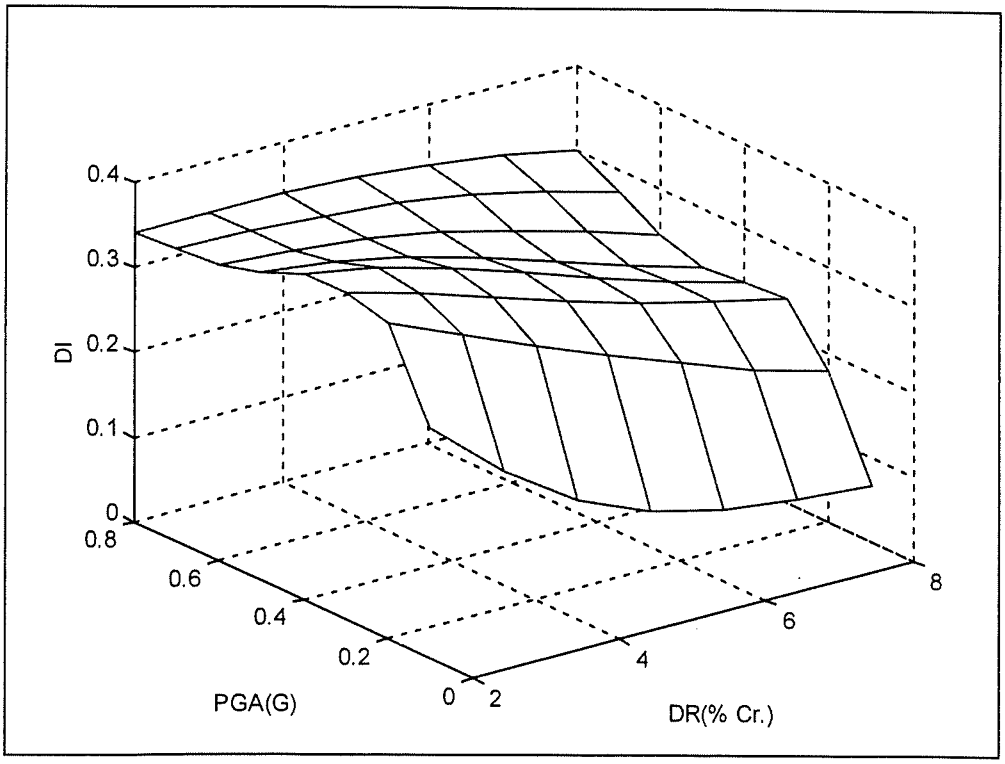

Figure 5.41 DR variation of NN-1b (3D)

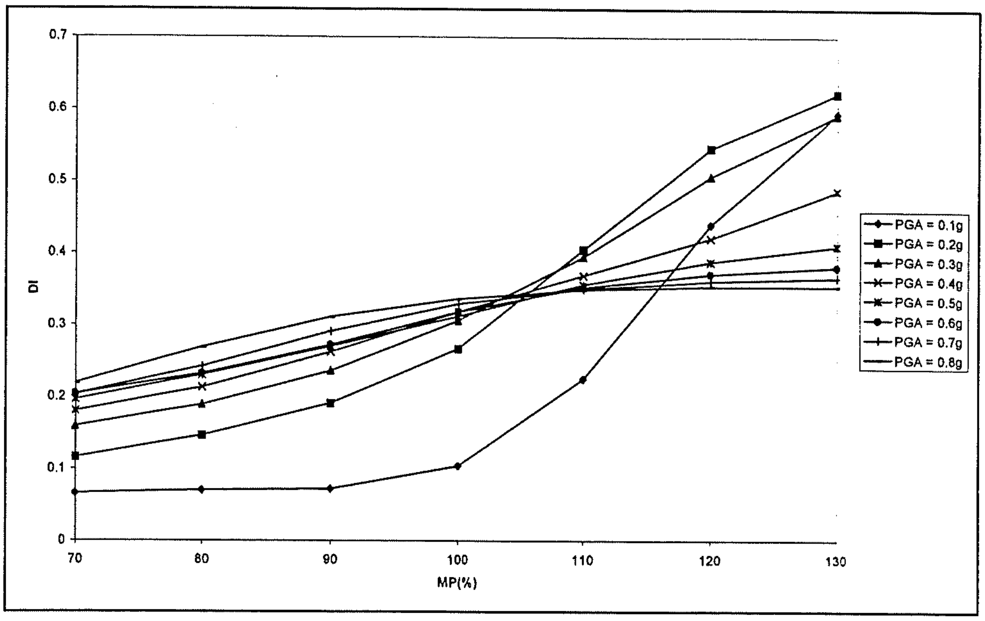

Figure 5.42 MP variation of NN-1b

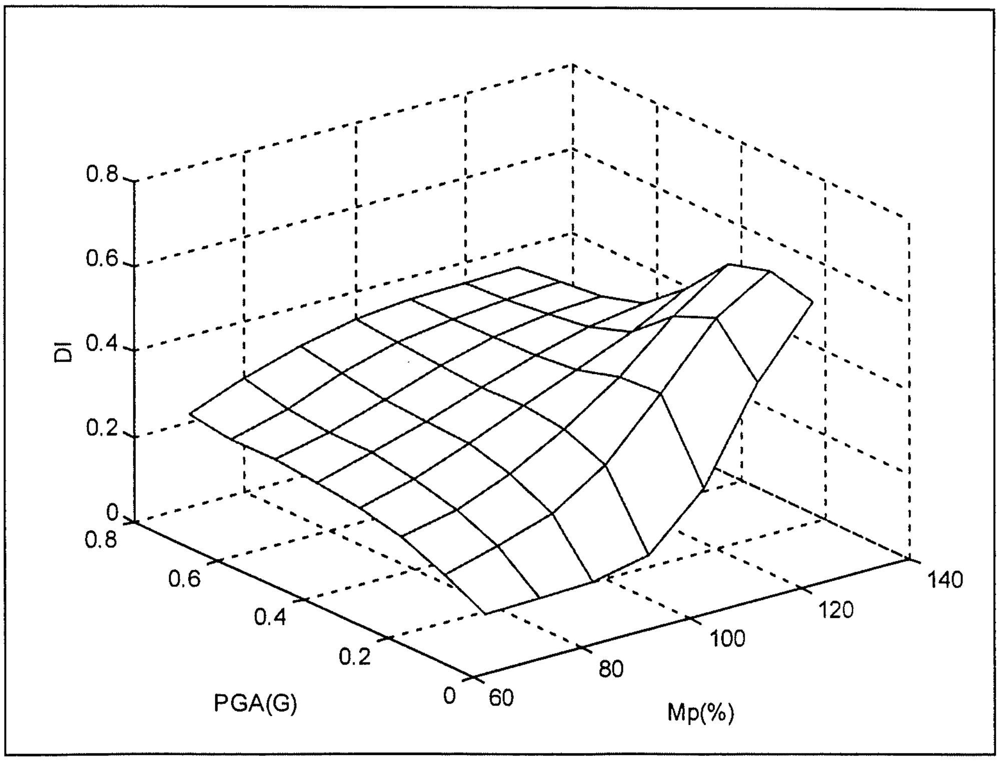

Figure 5.43 MP variation of NN-1b

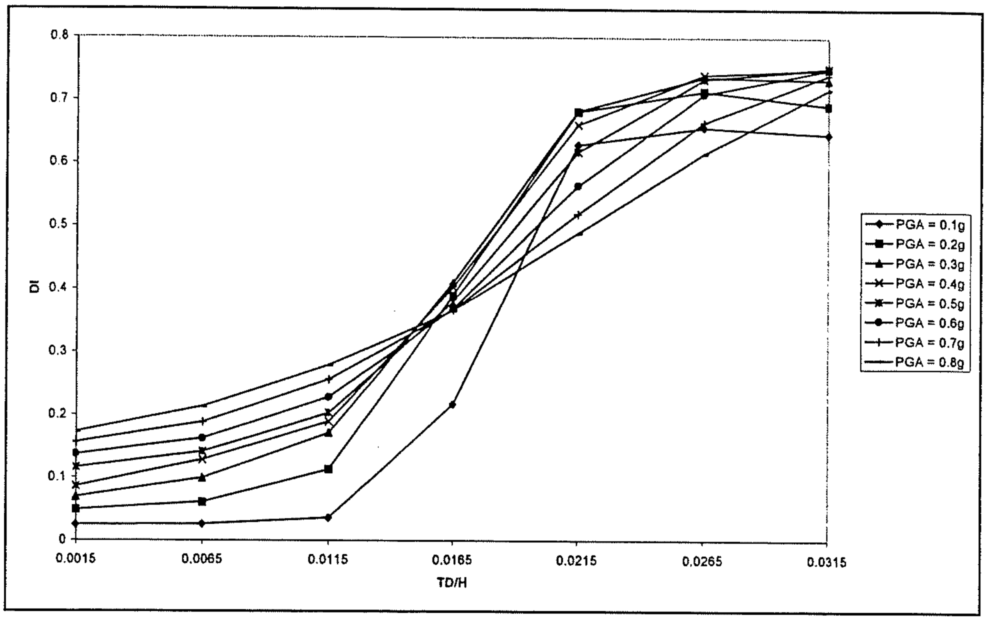

Figure 5.44 TD/H variation of NN-1b

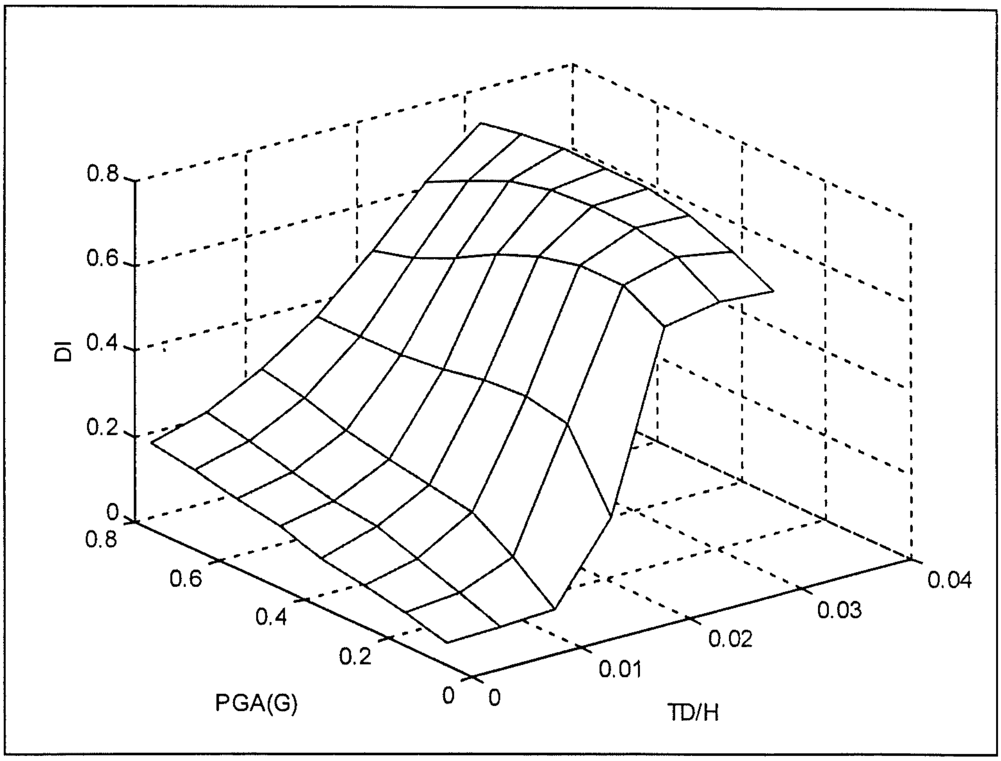

Figure 5.45 TD/H variation of NN-1b

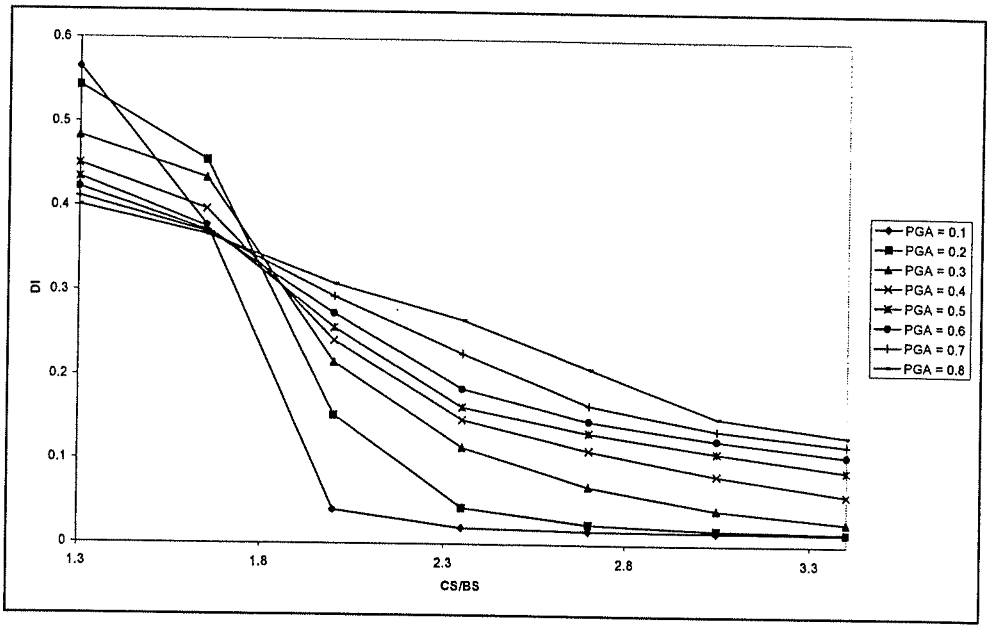

Figure 5.46 CS/BS variation of NN-1b

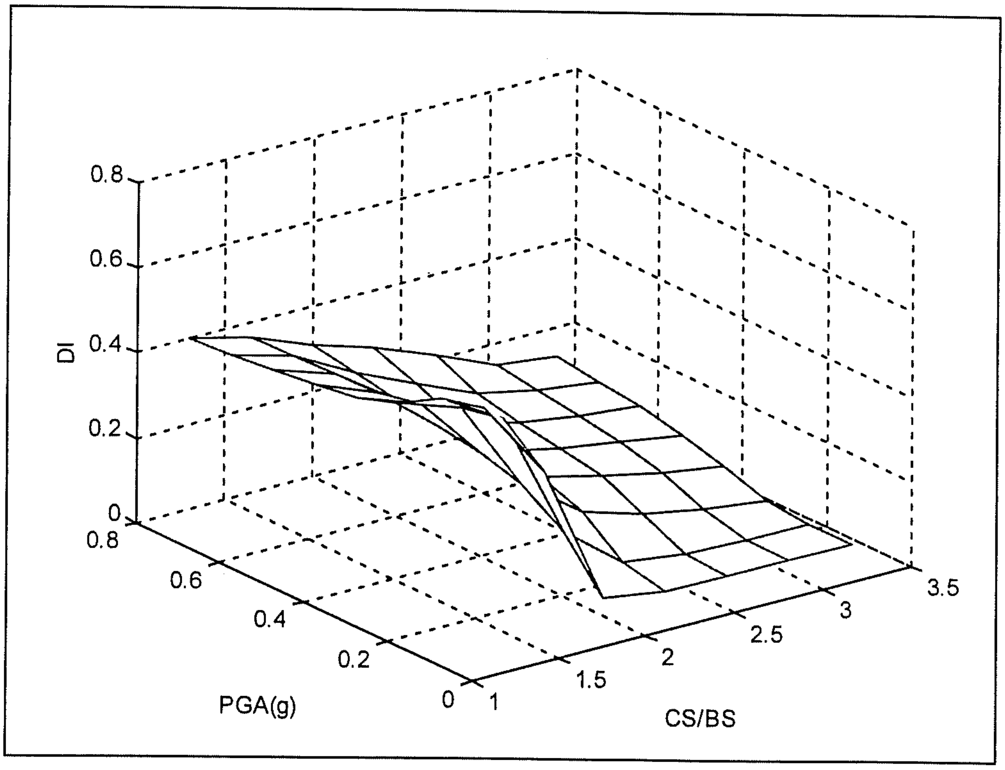

Figure 5.47 CS/BS variation of NN-1b

5.4 Second Neural Network: Damage Classification

In this part of study a new model, is tested (NN-2). This model does not aim to find the damage indices yet just to classify an earthquake experienced structural system. The architecture of this model is 15-17-9-4 with bias of –1 and sigmoid constant of 1. The damage classification proposed by Stone and Taylor [7] (Table 2.2) is used in this part. If the output layer are as shown in Figure 5.46, then the classification will be as shown in Table 5.4.

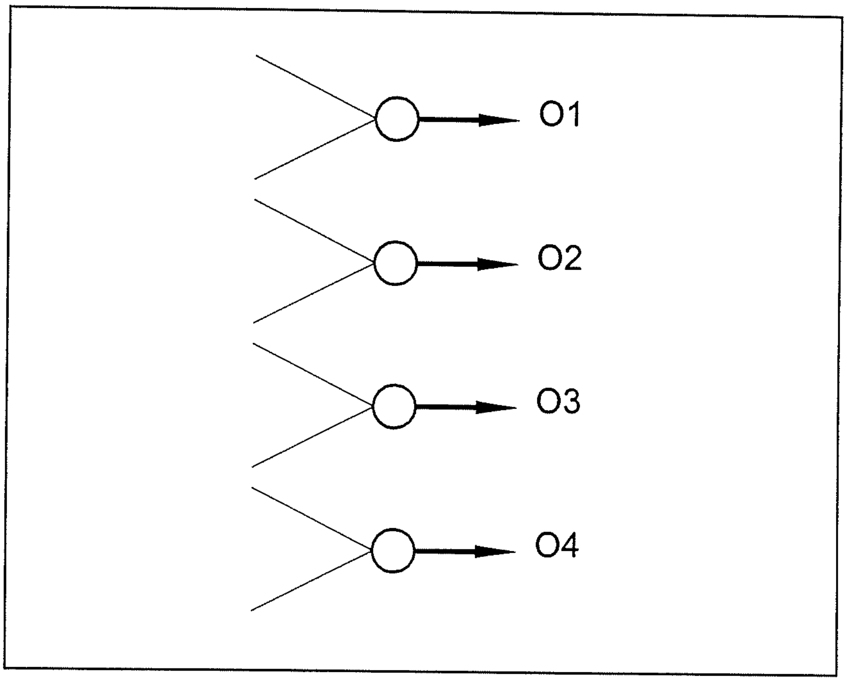

Figure 5.48 Output layer of NN-2

Table 5.4 Coding of NN-2 for damage classification

No	Classification	O1	O2	O3	O4
I	SLIGHT D.	1	0	0	0
II	REPAIRABLE	0	1	0	0
III	IRREPAIRABLE	0	0	1	0
IV	COLLAPSED	0	0	0	1

The training and the test sets for NN-2 is the same as training and test sets of NN-1a, which consists of 84 data and 445 data respectively. The following table summarises the erroneous estimation done by NN-2.

Table 5.5 Wrong estimations done by NN-2

Data No.	DI (IDARC)	Class. (IDARC)	Class. NN-2	O1(%)	O2(%)	O3(%)	O3(%)
12	0.422	III	II	0.3	**96.4**	3.2	0.1
56	0.100	I	II	2.3	**96.8**	0.7	0.1
66	0.125	II	I	**55.2**	44.4	0.3	0.1
201	0.099	I	II	1.1	**97.8**	1.0	0.1
202	0.097	I	II	34.3	**65.0**	0.5	0.1
239	0.416	III	II	0.3	**85.2**	14.4	0.1
304	0.414	III	II	2.2	**66.1**	31.7	0.1
305	0.406	III	II	0.3	**92.4**	7.2	0.1
369	0.422	III	II	0.2	**62.2**	37.4	0.1
370	0.401	III	II	0.4	**94.0**	5.5	0.1
403	0.091	I	II	9.3	**89.5**	1.1	0.1
404	0.099	I	II	1.0	**97.8**	1.1	0.1
405	0.106	II	II	4.9	**93.9**	1.2	0.0

The above table shows that NN-2 is more efficient then NN-1a although they have the same training and test sets.

5.5 Discussion of Results

5.5.1 General

Neural networks are gaining popularity in civil engineering area but are not yet suitable for market usage. Although it is reported that some structural analysis software is reported to have neural network module, the above study shows that damage analysis by neural networks is promising field but it is not ready for everyday practice yet.

In this study very small training sets were used. These sets were obtained from nonlinear damage analysis of simple frames. Hence the above network is optimised for only FR3 but not the other frames. Experimental data or real case data will be much more useful to create networks that work for several types of structures. For example most of the buildings in Turkey are frames with shear walls or masonry buildings. Instead of using software, real earthquake damage data of these buildings could be used for training. The aim of the study was not to obtain commercial software but just to examine the performance of neural networks and to create new models for damage analysis.

One of the mistakes done during the design of the above networks is to use a parameter, which is obtained from the nonlinear dynamic analysis namely TD/H. At the beginning this parameter was thought to have effect on damage and hence used in the network. Since the main reason for utilising neural networks is not to make a nonlinear analysis, the choice of TD/H is not entirely logical. The above network may only be utilised for the structures whose analyses have been made before.

Maximum error used as stopping criteria during the training of the networks. It was observed that the LR period and LR range concepts proposed in this study are other subjects of research. In this study one LR range was used for all weights but every weights may have its own LR range and period. Optimum LR ranges and periods may be obtained adaptively using several stochastic algorithms during the training phase.

The data sets were scaled before the training. The outputs were also scaled. It was observed that when L is not equal to the minimum damage in the training set, the maximum error increases. Hence a value of 0.032 is used for L.

Since the training and testing sets are based on the El Centro earthquake, the above networks are not expected to give satisfactory results for other earthquakes. The training set should include several data for several earthquakes and more parameters that define the earthquake properties should be used. In the later parts discussions specific to network models is given

5.5.2 NN-1a

This network was not successful at the implementations of CDP, BDP, and BSP especially for the larger values of PGA. The possible reason for this is the size of the learning set. The set size should be supported with the additional data at which the network fails. The other implementations were done successfuly. Although the training set is so small the efficiency of the network is good.

5.5.3 NN-1b

Figures from 5.11 to 5.17 show that the network adopted itself according to the learning set. The performance was very good for the modified frames especially for PGA < 0.7g. This result arises from the fact that the efficiency of the network increases with the increasing size of learning rate.

The graphs in Figures from 5.23 to 5.46 show us that the parameters have an effect similar to sigmoid function on the network. But it is impossible to visualise the behaviour of the network in *n*-dimensional space.

5.5.4 NN-2

This network is constructed utilising the main objective behind neural networks: classification. It is the most powerful network when compared with others. The Table 5.5 shows that the network fails when the damage indices are very close to values that define the boundaries of the classifications. Even the probabilities, which were estimated by NN-2, for several damage states are very satisfactory. This way of damage classification can give a decision parameter to the user for the structures that can be classified for two damage states.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 General

This study consists of two main parts. In the first part the effects of structural parameters on damage of generalised yielding systems have been studied. For this purpose the software IDARC was utilised for the nonlinear dynamic and damage analysis of three frames and an SDOF system. The built-in damage model was modified Park and Ang damage model, which incorporates maximum drift and hysteretic energy. The most effective parameters were determined through the use of the resulting graphs. All analyses were done with the El Centro earthquake record. Use of only one earthquake record was a limitation. Since the main aim of this study was to develop an approach for damage analysis and neural network study, additional earthquake records other then listed in Table 4.14 are not used. In the second part, two types of neural networks have been utilised for the implementation of the damage analysis of the structural systems. In the first model the most popular architecture was employed to obtain the damage index while in the second model a new method is employed for the classification of the damage. The performance of the first model was described by several graphs. The results for the second network were presented in tabular form. The conclusions drawn from this study and recommendations for the future research are listed.

6.2 Conclusions for the Damage Analysis

- El Centro earthquake seems to effect the damage when it is scaled such that PGA>0.2g. For the earthquake record calibrated for PGA<0.2g, the contributions of different structural parameters to the damage are more or less the same.

- The effects of shear wall, column, damping and structural mass on structural damage are observed to be larger for the earthquake loadings with large PGA. The limiting value for the El Centro earthquake is 0.2g. Beam properties do not have a clear effect on damage. Damage of frame with shear walls is highly effected by the shear wall area and steel ratio while column parameters, damping and mass effected damage of the frames without shear walls.
- TS-500, EUROCODE-2, ACI code specifications result in the most economical design for wire frames.
- The fundamental period of the structure and the deflection of the top storey are observed to have the most important effect on damage.
- IDARC appears to be a software package that still requires benchmark testing

6.3 Conclusions for the Neural Network Analysis

- Neural networks used for the estimation of damage indices prove to be efficient when a well distributed training set is used for training.
- Neural networks used for the classification of the damage prove to be more efficient than the classical model of neural network used in the previous case.
- A new concept called learning rate range can be introduced alternate to the adaptively changing learning rate.
- Although accelerated algorithms were not employed for the training, pure back-propagation algorithm performed very well.

6.4 Recommendations for future research

- Damage indices should be improved to take account of the effects of the behaviour of multi degree of freedom systems such as the stability of the structural systems.
- Better tested software should be used for shear wall analysis.
- Real earthquake-damage data should be used for training more generalised networks.
- The design of the neural networks should be improved by eliminating less important parameters and incorporating important parameters for the system identification. This implies a wide testing program.
- Neural networks could be used to obtain the most economical and CDP, CSP, BDP, BSP and MP values in a performance based format.
- Modular neural networks, which consist of more than one network, could be designed for the use of not only for one structure but also for several structures. In these networks an initial network could make a system identification that searches for the network reflecting the structural system better.

REFERENCES

[1] Ghobarah, A., Abou-Elfath, H. Biddah, A., 1999. " Response-based Damage Assesment of Structures", Journal of Earthquake Engineering and Structural Mechanics. Vol28, pp. 79-104

[2] Haykin, S., 1994, Neural Networks: A Comprehensive Foundation, Prentice Hall.

[3] Kunnath, S. K., Reinhorn, A. M., Park, Y. J., 1990, "Analytical Modelling of Inelastic Seismic Response of R/C Structures". Journal of Structural Engineering, ASCE, Vol. 116, No. 4, pp996-1017

[4] Molas G. L., Yamazaki, F., 1994, "An Earthquake Damage Model Using Neural Networks", Bulletin of Earthquake Resistant Structure Research Center, No: 27, pp 89-101, Institute of Industrial Science, University of Tokyo.

[5] Nakamura M., Sami F. M., Chassiakos A., G., Caughey T. K., 1998, " A Method for Non-Parametric Damage Detection through Use of Neural Networks", Earthquake Engineering and Structural Dynamics, Vol. 27, pp. 997-1010.

[6] Park, Y. J., and Ang, A. H. S., 1985, "Mechanistic Seismic Damage Model for Reinforced Concrete" Journal of Structural Engineering, ASCE, Vol. 111, No. 4, pp 722-739

[7] Williams, M. S., Sexsmith, R. G., 1995. "Seismic Damage Indices for Concrete Structures: A-state-of-the-Art Review", Earthquake Spectra, Vol. 11, No.2, pp 319 349.

[8] Wu, X., Ghaboussi, J., Garrett, J. H. Jr., 1992, "Use of Neural Networks in Detection of Structural Damage", Computers and Structures, Vol. 42, pp. 649-658.

[9] Valles, R. E., Reinhorn, A. M., Kunnath, S. K., Li, C., Madan, A., 1996, "IDARC2D Version 4.0: A Computer Program for the Inelastic Damage Analysis of Buildings", Technical Report NCEER-96-0010, National Center for Earthquake Engineering Research.