Simulation & Identification of sheet defects using FEM & ANN

A Thesis report

submitted towards the partial fulfillment of the

requirements of the degree of

Master of Engineering

in

Electronics Instrumentation and Control Engineering

submitted by

Deepak Bajaj

Roll No-800851004

Under the supervision of

Dr. Mandeep Singh
Assistant Professor, EIED

DEPARTMENT OF ELECTRICAL AND INSTRUMENTATION ENGINEERING

THAPAR UNIVERSITY, PATIALA – 147004

JULY, 2010

DECLARATION

I hereby declare that the report entitled ***"Simulation & Identification of sheet defects using FEM & ANN"*** *is an authentic record of my own work carried out as requirements for the award of degree of M.E. (Electronic Instrumentation & Control) at Thapar University, Patiala, under the guidance of Dr. Mandeep Singh (Assistant Professor, EIED) during January to July 2010.*

Date: July 13, 2010

Deepak
Deepak Bajaj

Roll No. - 800851004

It is certified that the above statement made by the candidate is correct to best of my knowledge and belief.

13/07/10
Dr. Mandeep Singh
Assistant Professor, EIED,
Thapar University, Patiala

S. Ghosh 13/7/10
Dr. Smarajit Ghosh
Professor & Head, EIED,
Thapar University, Patiala

Dr. R.K. Sharma
Dean of Academic Affairs
Thapar University, Patiala

ACKNOWLEDGEMENT

Words are often too less to reveals one's deep regards. An understanding of the work like this is never the outcome of the efforts of a single person. I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis.

First of all I would like to thank the Supreme Power, one who has always guided me to work on the right path of the life. Without His grace this would never come to be today's reality.

This work would not have been possible without the encouragement and able guidance of my supervisor, Dr. Mandeep Singh. Their enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Their feedback and editorial comments were also invaluable for the writing of this thesis.

I would like to express my deep sense of gratitude towards Dr. Smarajit Ghosh, Professor and Head, EIED, Thapar University, Patiala who has been a constant source of inspiration for me throughout this work.

I would also like thank all the faculty members of the department and my friends who directly or indirectly helped me in completion of my thesis.

No words of thanks are enough for my dear parents whose support and care makes me stay on earth. Thanks to be with me.

Date: July 13, 2010
Place: Thapar University, Patiala

Deepak
Deepak Bajaj

ABSTRACT

In this work simulation and identification of defect in metallic wall is done using the Finite element method (FEM) and Artificial Neural Network (ANN). The problem of defect identification is an inverse problem which has been solved in this work. Recently, ANNs are introduced to solve the inverse problems in most of the research applications in industrial non-destructive testing, mathematical modeling, medical diagnostics, geophysical prospecting for petroleum and minerals, and detection of earthquakes. Here FEM is used for the analysis of change in magnetic induction properties on the account of defect present in the metallic sheet. Now for the simulation of these defects on metallic wall both variations in width and height of the defects are considered. The simulation is done using the Pdetool of Matlab. The obtained results are used to generate a set of vectors for the training of neural network (NN).

The basic problem of training of NN is that it requires some input data for training by simulation using FEM this problem has been solved as this provide an input data for training of ANN. Therefore NN has been trained for the given dataset, finally, the obtained neural network is used to identify a group of new defects, simulated by the finite element method, but not belonging to the original dataset. The network can be embedded in electronic devices in order to identify defects in real metallic walls. The association of FEM and ANN techniques seems to be a useful alternative for identification of defects trough inverse analysis. Future works are intended to be done in this field, such as the use of more realistic FEM, computer parallel programming, in order to get quickly solutions.

TABLE OF CONTENTS

Contents	**Page No.**

List of figures

List of tables

List of Abbreviations

• MLP	Multilayer Perceptron Neural Network
• RBF	Radial Basis Functions
• ANN	Artificial Neural Network
• EM	Electromagnetic
• RNN	Recurrent Neural Networks
• PDE	Partial Differential Equations
• FEM	Finite-Element Method
• FENN	Finite-Element Neural Network
• IP	Inverse Problem
• ODE	Ordinary Differential Equations
• NN	Neural Network
• BEM	Boundary element method
• FDM	Finite difference method

Hacib et al. (2006) in this paper presented an approach which is based on the use of artificial neural networks and finite element analysis to solve the inverse problem of defect identification. The approach is used to identify unknown defects in metallic walls. The methodology used in this study consists in the simulation of a large number of defects in a metallic wall, using the finite element method. Both variations in width and height of the defects are considered. Then, the obtained results are used to generate a set of vectors for the training of two neural network models: multilayer perceptron neural network (MLP) and radial basis functions (RBF). Finally, the obtained neural networks are used to classify a group of new defects, simulated by the finite element method, but not belonging to the original dataset. The reached results demonstrate the efficiency of the proposed approach, and encourage future works on this subject. [44]

Wan et al. (2008) presented an overview of emerging artificial neural network (ANN) techniques and applications for electromagnetic (EM) simulation and design. Accurate time domain EM modelling using recurrent neural networks (RNNs) is reviewed. Advanced robust training algorithm combining particle swarm optimization and quasi-Newton method is described through frequency domain EM modelling, showing its ability to avoid ANN training being trapped in local minima to obtain accurate models. ANN applications in computational electromagnetics are also discussed. Great efficiency can be achieved by using ANNs to approximate the computationally intensive calculations in solving Maxwell equations using method of moments. As illustrated in examples, these ANN-based techniques are capable of fast and accurate EM modelling and method of moments computation, and useful for efficient EM based design. [45]

In the magnetic flux leakage method of non-destructive testing commonly used to inspect ferromagnetic materials, a crucial problem is signal inversion, wherein the defect profiles must be recovered from measured signals. Pradeep Ramuhalli and Lalita Udpa(2002) proposed a neural-network-based inversion algorithm to solve the problem. Neural networks (radial-basis function and wavelet-basis function) are first trained to approximate the mapping from the signal to the defect space. The trained networks are then used iteratively in the algorithm to estimate the profile, given the measurement signal. The paper presents the results of applying the algorithm to simulated magnetic flux leakage data. [46]

Back propagation Algorithm is applied in transformer oil diagnosis. The Algorithm is trained by Levenberg-Maquardt technique, Momentum technique and Momentum & adaptive learning rate technique. A comparison between the three techniques was done by the L.Mokbnache and A.Boubakeur(2002). [47]

In this paper a Multi-Frequency Excitation and Spectrogram Eddy Current System and an inverse neural model were used by Chady et al. (2001) to detect and identify natural flaws in steam generator tubes. It is shown that the applied dynamic neural model of the ECT sensor offers very high speed of operation and guarantees reliability of the recognition results. [48]

Multilayer neural networks, trained via the back-propagation rule, are proved to provide an efficient means for solving electric and/or magnetic inverse problems. The underlying model of the system is learned by the network by means of a dataset defining the relationship between input and output parameters. The merits of the method are illustrated by Coccorese et al. (1994) at the light of three example cases. The first two samples deal with inverse electrostatic problems which are relevant for non-destructive testing applications. In a first problem, a boss on an earthed plane is identified on the basis of the map of potential produced by a point charge. In the second problem, the geometric parameters of an ellipsoid carrying an electric charge are identified. In both cases, database of simulated measurements has been generated thanks to the available analytical solutions. As a sample magnetic inverse problem, the identification of circular plasma in a tokamak device from external flux measurements is carried out. The results achieved show that the method here proposed is promising for technically meaningful applications. [49]

One property of the Hopfield neural network is the monotonous minimization of energy as time proceeds. In this paper this property is applied to minimize the energy functional obtained by ordinary finite element analysis. The mathematical representation and correlation between finite element and neural network calculus are presented by Yamashita et al. (1995). The selection of the sigmoid function arid its influence on the iteration process is discussed. The obtained results using the proposed method show excellent agreement with theoretical solutions. [50]

The solution of partial differential equations (PDE) arises in a wide variety of engineering problems. Solutions to most practical problems use numerical analysis techniques such as finite-element or finite-difference methods. The drawbacks of these approaches include computational costs associated with the modelling of complex geometries. This paper by Ramuhalli et al. (2005) proposes a finite-element neural network (FENN) obtained by embedding a finite-element model in a neural network architecture that enables fast and accurate solution of the forward problem. Results of applying the FENN to several simple electromagnetic forward and inverse problems are presented. Initial results indicate that the FENN performance as a forward model is comparable to that of the conventional finite-element method (FEM). The FENN can also be used in an iterative approach to solve inverse problems associated with the PDE. Results showing the ability of the FENN to solve the inverse problem given the measured signal are also presented. The parallel nature of the FENN also makes it an attractive solution for parallel implementation in hardware and software. [51]

A method that combines a neural network and the finite element method is introduced for solving inverse electromagnetic field problems. This form the basis for design synthesis. A two layered neural network with one pass training is used in this proposed scheme by T. S. Low and Bi Chao (1992). It uses the information from the finite element analysis for training and is very efficient and stable. The one paw training of the neural network leads to a time efficient scheme. The finite element method is used, to produce the training patterns and for analyzing the optimized solution and the neural network is used for optimizing the parameters. With the use of the trained neural network for optimization, the solution time for design optimization is reduced. An example of its use in the optimization of a permanent-magnet rotor configuration is presented in this paper. [52]

This work by Alcantara Jr. et al. (2002) presents an investigation into the use of the finite element method and artificial neural networks in the identification of defects in industrial plants metallic tubes, due to the aggressive actions of the fluids contained by them, and\or atmospheric agents. The methodology used in this study consists of simulating a very large number of defects in a metallic tube, using the finite element method. Both variations in width and height of the defects are considered. Then, the obtained results are used to generate a set of vectors for the training of a perceptron multilayer artificial neural network. Finally, the obtained neural network is used to classify a group of new defects, simulated by the finite element method, but that do not belong to the original dataset. The reached results demonstrate the efficiency of the proposed approach, and encourage future works on this subject. [53]

The Hopfield neural network is applied to the solution of linear systems of equations obtained by the Finite Element Analysis. The selection of the form of the neurons' activation function is very crucial for the success of the proposed method by the I.T. Rekanos and T.D. Tsiboukis (1996). A method for the selection of the parameters describing a sigmoid activation function is proposed. Finally the hopfield neural network is applied to an eddy current problem where the linear system of equations is complex. [54]

In this paper, Zhao et al. (2008) examined the capability of the neural network approach to the defect identification and the electrical impedance imaging. It is shown the approach is suitable for detecting the defects and their positions in a conductive field system. [55]

The eddy current non-destructive testing of conductive materials is well known problem. For some eddy current transducers for flaw detection the mathematical models were constructed and the inverse problem (IP) was formulated in order to solve those problems the artificial neural network approach, relatively new, was adopted. The public domain software was used to teach and test the network. It was proved that, a standard feed-forward multilayered neural network is sufficient to identify the parameters of different kind defects with a satisfying accuracy. Neural network approach could be extinguished to impedance tomography and to eddy current tomography. Several examples illustrated abilities of proposed approach will be presented in the paper. Two artificial neural network reconstruction methods for electrical impedance tomography have been presented in this paper by the Sikora et al. (2008). The problem under study concerns the reconstruction of the conductivity distribution inside the investigated area using the information collected from the boundary. The first approach consists in ANN learning using electrical potential vectors, which were obtained from numerical solution of the forward problems. The second method using a standard feed-forward multilayered neural network applies the circuit representation for the finite element discretisation. Using the quadrilateral finite element, the neural network structure for EIT problem has been proposed. The advantages and disadvantages of both methods with respect to the classical approach have been discussed in details. [56]

Lagaris et al.(1998) presented a method to solve initial and boundary value problems using artificial neural networks. A trial solution of the differential equation is written as a sum of two parts. The first part satisfies the initial/boundary conditions and contains no adjustable parameters. The second part is constructed so as not to affect the initial/boundary conditions. This part involves a feed-forward neural network containing adjustable parameters (the weights). Hence by construction the initial/boundary conditions are satisfied and the network is trained to satisfy the differential equation. The applicability of this approach ranges from single ordinary differential equations (ODE), to systems of coupled ODE and also to partial differential equations (PDE). In this article, we illustrate the method by solving a variety of model problems and present comparisons with solutions obtained using the Galerkin finite element method for several cases of partial differential equations. With the advent of neuro-processors and digital signal processors the method becomes particularly interesting due to the expected essential gains in the execution speed. [57]

Chapter 1

Introduction

1.1 Introduction

Recently, artificial neural networks have been introduced to solve the electromagnetic inverse problems [1] and [2]. Inverse problems arise in a number of areas. An inverse problem is the task that often occurs in many branches of science and mathematics where the values of some model parameter(s) must be obtained from the observed data. The inverse problem can be formulated as follows:

$$\text{Data} \rightarrow \text{Model parameters}$$

Examples of inverse problems include industrial non-destructive testing, medical diagnostics, geophysical prospecting for petroleum and minerals, and detection of earthquakes. Magnetic inverse problems can sometimes be stated as simply as the following: if there is an electromagnetic device, it is easy to calculate the magnetic induction in any region of the device. Inverse problems are typically ill posed, as opposed to the well-posed problems more typical when modelling physical situations where the model parameters or material properties are known. An inverse problem is said to be well-posed in the sense of Jacques Hadamard if the solution satisfies three properties:

(i) Existence

(ii) Uniqueness and

(iii) Continuity: the solution depends continuously on the input/stability of the solution or solutions.[60]

Of the three conditions for a well-posed problem suggested by Jacques Hadamard the condition of stability is most often violated. In the sense of functional analysis, the inverse problem is represented by a mapping between metric spaces. While inverse problems are often formulated in infinite dimensional spaces, limitations to a finite number of measurements, and the practical consideration of recovering only a finite number of unknown parameters, may lead to the problems being recast in discrete form. In this case the inverse problem will typically be ill-conditioned. In these cases, regularization may be used to introduce mild assumptions on the solution and prevent over fitting. Many instances of regularized inverse problems can be interpreted as special cases of Bayesian inference.

An inverse problem is to find m such that (at least approximately)

$$d = G(m)$$

Where G is an operator describing the explicit relationship between data d and model parameters m, and is a representation of the physical system. In various contexts, the operator G is called forward operator, observation operator, or observation function. The forward problem in general is well-poised and is solved analytically or by means of numerical modelling. In contrast, inverse problems in general are ill-posed, lacking both uniqueness and continuous dependence of the measured signals on defects. This has resulted in the development of a variety of solution techniques ranging from simple calibration procedures to other direct and iterative approaches. [3]

What, about taking some values of magnetic induction to predict defects in a region of the electromagnetic device. Since the inverse problem is highly nonlinear and without formulations to follow, it is very difficult to construct an effective inversion algorithms. An artificial neural network, however, has the following properties: nonlinearity, input-output mapping, fault tolerance and most important, learning from examples. The need for learning from examples is closely related to the difficulty of formulating explicit rules. An artificial neural network (ANN), usually called "neural network" (NN), is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modelling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data.

Artificial neural networks are based on abstracting from the complex details of human thought and building a simple model using a network of simple processors. Artificial neural networks consist of a large number of simple processing elements called neurons or nodes. Each neuron is connected to other neurons by means of directed links, each with an associated weight. The weights represent information being used by the network to solve a problem. The artificial neural network essentially determines the relationship between input and output by looking at examples of many input-output pairs. In learning processes, the actual output of the artificial neural network is compared to the desired output. Changes are made by modifying the connection weights of the artificial neural network to produce a closer match. The procedure iterates until the error is small enough. In this work an investigation on the use of finite element method (FEM) and artificial neural network (ANN) in the identifications of defects in metallic walls is presented. A comprehensive description of FEM is given in chapter 2 while that of ANN is given in chapter 3.The methodology consists of the following steps:

1) A large number of defects in a metallic wall are simulated using the finite element method.
2) The obtained results are then used to generate the training vectors for a multilayer perceptron artificial neural network.
3) The trained network is used to identify new defects in the metallic wall, which not belong to the original dataset.
4) The network weights can be embedded in an electronic device, and used to identify defects in real pieces, with similar characteristics to those of the simulated ones

The FEM (its practical application often known as finite element analysis (FEA)) is a numerical technique for finding approximate solutions of partial differential equations (PDE) as well as of integral equations. The solution approach is based either on eliminating the differential equation completely (steady state problems), or rendering the PDE into an approximating system of ordinary differential equations, which are then numerically integrated using standard techniques such as Euler's method, Runge-Kutta, etc.

Boundary conditions for these problems are usually of two types: natural boundary conditions and essential boundary conditions. Essential boundary conditions (also referred to as Dirichlet boundary conditions) In mathematics, the Dirichlet (or first-type) boundary condition is a type of boundary condition, named after Johann Peter Gustav Lejeune Dirichlet (1805-1859)[5]. When imposed on an ordinary or a partial differential equation, it specifies the values a solution needs to take on the boundary of the domain. The question of finding solutions to such equations is known as the Dirichlet problem. Natural boundary conditions (of which Neumann boundary conditions are a special case) in mathematics, the Neumann (or second-type) boundary condition is a type of boundary condition, named after Carl Neumann [5]. When imposed on an ordinary or a partial differential equation, it specifies the values that the derivative of a solution is to take on the boundary of the domain. Imposition of these conditions into the finite element formulation is straightforward. Natural boundary conditions are incorporated into the functional and are satisfied automatically during the solution procedure. Dirichlet boundary conditions, on the other hand, need to be handled separately.

For the methodology presented here, the measured values are independent of the relative motion between the probe and the piece under test. In other words, the movement is necessary only to change the position of the probes, to acquire the field’s values, which are necessary for the identification of new defects. Furthermore, the use of neural network in conjunction with the finite element method permits a very good determination of both, width and height of the defect.

Chapter 2

Finite Element Method

2.1 Introduction

There are generally three types of computational method used in design and manufacture:

i) Boundary element method(BEM)
ii) Finite difference method (FDM)
iii) Finite element method (FEM)

Computational methods imply that the problem will be solved using numerical methods and computer. Of these three methods, finite element is extremely popular because many practical problems can be solved using FEM. This has given impetus for growth of FEM and also nowadays many companies can buy fast computing workstations to solve very complex finite element method. Although finite element method was initially launched by Boeing Aircraft Company for analysis of stress in wings of an aircraft, today there are thousands of papers published in FEM and on its applications.

2.1.1 Boundary element method

In boundary element method transformation of governing differential into integral identities that are applicable on the boundary or surface. Numerical integration of these integral identities is done for the boundary which is divided into smaller boundary segments. As in other two methods a unique solution is obtained from a linear differential equation, provided that boundary conditions are satisfied.

2.1.1.1 Comparison to other methods

- The boundary element method is often more efficient than other methods, in terms of computational resources for problems where there is a small surface/volume ratio.

- In certain problems BEM is less efficient then volume-discretisation methods (FEM, FDM).
- In BEM computation time and storage requirements grows as the square of problem of size compared to FEM in which it grows linearly with the problem size.

2.1.2 Finite difference Method

In this method difference equations is written which contains the terms of derivatives of partial differential equations. Thus, in the case of a two dimensional object or domain inside it a cells grid is placed and at every interior point a differencing approximation is applied. Due to this course of action a linear algebraic equations are obtained with an unique solution but provided that given boundary conditions of the problem are satisfied. It is simplest of the three mentioned methods and its programming is relatively easy. It has one major drawback that it is not suitable for engineering problems which has an irregular shape or geometries. Also, it is not suitable for problems which have fast changing variables.

2.1.2.1 Comparison to other methods

- Simplest of the three mentioned methods.
- Programming is relatively easy
- Not suitable for engineering problems which has an irregular shape or geometries
- Not suitable for problems which have fast changing variables.

2.1.3 Finite element method

In finite element method, the complete solution domain is divided into small finite segments. Every segment characteristic is defined by governing differential equations. After this all elements are assembled together and between neighbouring elements condition of equilibrium and continuity are satisfied. Finite element methods is suitable for engineering problems which has very complex geometries and for obtaining accuracy in the case of rapidly changing variables a huge number of finite elements are used.

2.1.3.1 Comparison to other methods

Comparison of the Finite element method to the finite difference method

The finite difference method (FDM) is an alternative way of approximating solutions of PDEs. The differences between FEM and FDM are:

- The most attractive feature of the FEM is its ability to handle complicated geometries (and boundaries) with relative ease. While FDM in its basic form is restricted to handle rectangular shapes and simple alterations thereof, the handling of geometries in FEM is theoretically straightforward.
- The most attractive feature of finite differences is that it can be very easy to implement.
- There are several ways one could consider the FDM a special case of the FEM approach. One might choose basis functions as either piecewise constant functions or Dirac delta functions. In both approaches, the approximations are defined on the entire domain, but need not be continuous. Alternatively, one might define the function on a discrete domain, with the result that the continuous differential operator no longer makes sense, however this approach is not FEM.
- There are reasons to consider the mathematical foundation of the finite element approximation more sound, for instance, because the quality of the approximation between grid points is poor in FDM.
- The quality of a FEM approximation is often higher than in the corresponding FDM approach, but this is extremely problem dependent and several examples to the contrary can be provided

2.2 FEM: Overview

The finite element method is a numerical analysis technique for obtaining approximate solutions to a wide variety of engineering problems. Although originally developed to study stresses in complex airframe structures, it has since been extended and applied to the broad field of continuum mechanics. Because of its diversity and flexibility as an analysis tool, it is receiving much attention in engineering schools and in industry. The finite element method (FEM), or finite element analysis (FEA), is based on the idea of building a complicated object with simple blocks, or, dividing a complicated object into small and manageable pieces. Application of this simple idea can be found everywhere in everyday life, as well as in engineering. There are generally three types of computational method as mentioned above the word computational method indicate that the problems will be solved using numerical methods and computer, of these three methods FEM is extremely popular reasons for this is that industry is tuned to the fact that FEM can be used to solve many practical problems that has given an impetus to that kind of explosion in the market for FEM, the other reason is that computer becomes very powerful and affordable lot of companies can buy high end workstations to solve very complex finite element method which will be very useful for them; though FEM started approximately forty years back when Boeing aircraft company launch an project to determine the stresses in their aircraft wings, today we have approximately 20-22000 papers published in this topic

Examples:

- Lego (kids' play)
- Buildings

- Approximation of the area of a circle:

Figure 2.1 Lego and buildings

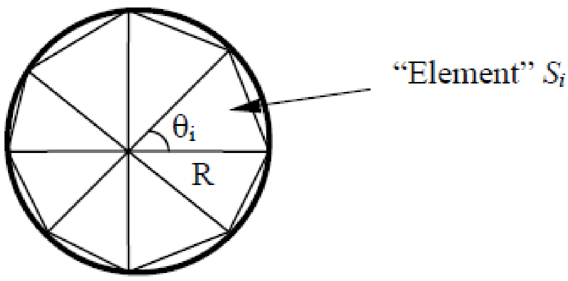


Figure 2.2 A circle

Area of one triangle: $S_i = \frac{1}{2} R^2 \sin\theta_i$

Area of the circle: $S_N = \sum_{i=1}^{N} S_i = \frac{1}{2} R^2 N \sin\left(\frac{2\pi}{N}\right) \longrightarrow \pi R^2$ as $N \longrightarrow \infty$

Where N = total number of triangles (elements).

Observation: Complicated or smooth objects can be represented by geometrically simple pieces (elements).

Consider a simple problem to find the area of complex sheet of figure 2.3 given below, to find out the area of a given sheet will be divided into known geometrical shape, area of this shapes is already known , so first thing is to divide into geometrical shape the second step is to calculate individual areas third step is to assemble it; there are number of ways to calculate the area of sheet ,of course FEM is not used to calculate the area of sheet but it's used in number of applications such as to calculate temperature ,velocity, displacement, vibrations, stresses etc .

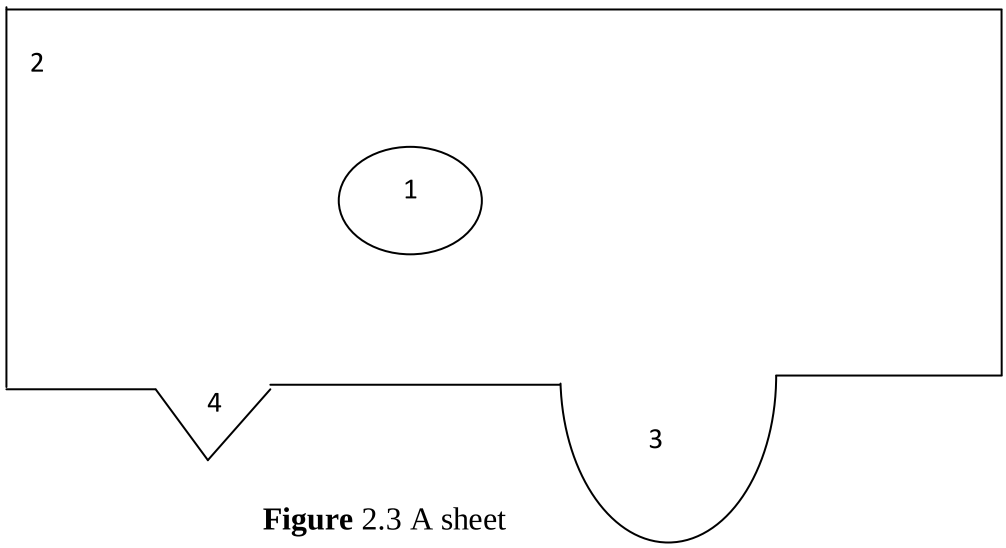


Figure 2.3 A sheet

Whole philosophy of FEM rest on divide and conquer, as to calculate the area of the above sheet the first thing comes in mind is divide the area into known small geometrical shapes, then assemble all these areas to calculate the area of whole sheet, instead of this one can also do one thing also just to use integration to calculate the area. In more and more engineering situations today, it has been found that it is necessary to obtain approximate numerical solutions to problems rather than exact closed-form solutions. For example, one may want to find the load capacity of a plate that has several stiffeners and odd-shaped holes, the concentration of pollutants during non uniform atmospheric conditions, or the rate of fluid flow through a passage of arbitrary shape. Without too much effort, it can be written down; the governing equations and boundary conditions for these problems, but it can be seen immediately that no simple analytical solution can be found Analytical solutions to problems of this type seldom exist; yet these are the kinds of problems that engineers are called upon to solve. The resourcefulness of the analyst usually comes to the rescue and provides several alternatives to overcome this dilemma. One possibility is to make simplifying assumptions to ignore the difficulties and reduce the problem to one that can be handled. Sometimes this procedure works; but, more often than not, it leads to serious inaccuracies or wrong answers. Now that computers are widely available a more viable alternative is to retain the complexities of the problem and find an approximate numerical solution. Several approximate numerical analysis methods have evolved over the years; a commonly used method is the finite difference [6] scheme. The familiar finite difference model of a problem gives a point wise approximation to the governing equations. This model (formed by writing difference equations for an array of grid points) is improved as more points are used. With finite difference techniques we can treat some fairly difficult problems; but, for example, when we encounter irregular geometries or an unusual specification of boundary conditions, we find that finite difference techniques become hard to use. Unlike the finite difference method, which envisions the solution region as an array of grid points, the finite element method envisions the solution region as built up of many small, interconnected sub regions or elements.

A finite element model of a problem gives a piecewise approximation to the governing equations. The basic premise of the finite element method is that a solution region can be analytically modelled or approximated by replacing it with an assemblage of discrete elements. Since these elements can be put together in a variety of ways, they can be used to represent exceedingly complex shapes. Briefly finite element method can be explained as a computational method which is used to obtain approximate solution for a boundary value or field problems in engineering. In this a huge area or volume is divided or meshed into smaller known area then a differential equation is written for this and generalised for the whole system. Certain steps are common to all such analyses done with finite element method. These are:

i) Pre-processing
ii) Solution
iii) Post-processing

2.2.1 Pre-processing

This step consists of defining a model and also includes following

i) Geometric domain of the problem is defined
ii) Element type to be used in the problem is defined
iii) Material properties of the elements is defined
iv) Geometrical details of the elements is defined
v) Connectivity of the element is defined
vi) Physical constraint is defined(Boundary conditions)
vii) Loading is defined

2.2.2 Solution

In this phase, all governing algebraic equations are assembled in matrix form and value of the primary field variables is computed.

2.2.3 Post-processing

The solution results obtained is analysed and evaluated in post-processing. There are certain tasks which are performed in post-processing like

- Sorting of element stresses in order of magnitude
- Equilibrium is checked
- Factors of safety are calculated
- Deformed structural shape is plotted
- Dynamic model behaviour is plotted

As it is a cumbersome task to be done manually, a number of software packages are available to perform the finite element analysis. Some of these are:

- ANSYS
- ABAQUS
- SDRC-ideas
- FEBio

The most important point is to apply engineering domain knowledge to solution result to realize that it is physically realizable or not.

As an example of how a finite difference model and a finite element model might be used to represent a complex geometrical shape, consider the turbine blade cross section in Figure 2.4. For this device we may want to find the distribution of displacements and stresses for a given force loading or the distribution of temperature for a given thermal loading. The interior coolant passage of the blade, along with its exterior shape, gives it a non simple geometry. A uniform finite difference mesh would reasonably cover the blade (the solution region), but the boundaries must be approximated by a series of horizontal and vertical lines (or “stair steps”). On the other hand, the finite element.

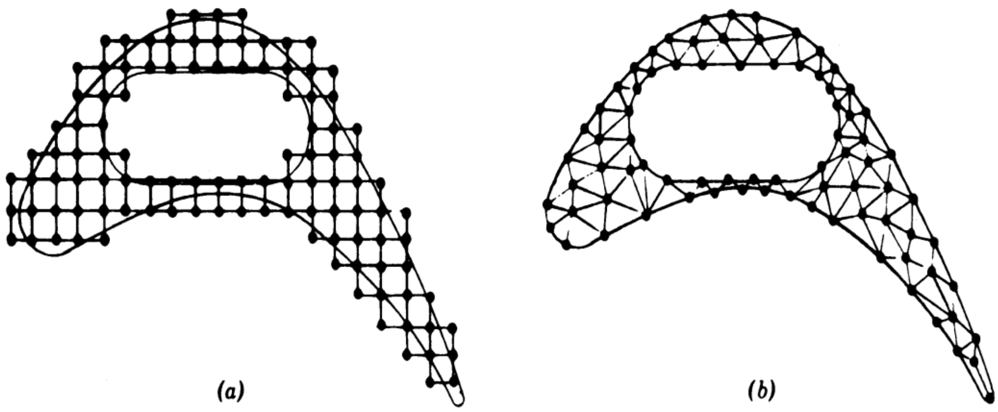


Figure 2.4 (*a*) Finite difference and (*b*) finite element discretizations of a turbine blade profile.

Model (using the simplest two-dimensional element—the triangle) gives a better approximation to the region. Also, a better approximation to the boundary shape results because the curved boundary is represented by straight lines of any inclination. This example is not intended to suggest that finite element models are decidedly better than finite difference models for all problems. The only purpose of the example is to demonstrate that the finite element method is particularly well suited for problems with complex geometries. Still another numerical analysis method is the boundary element method (boundary integral equation method) [7–9]. This method uses Green's theorem to reduce the dimensionality of the problem; a volume problem is reduced to a surface problem, a surface problem is reduced to a line problem. The turbine blade cross section example of Figure 2.4 would have no interior mesh, but rather a mesh of connected points along the exterior boundary and a mesh of connected points along the interior boundary. This method is computationally less efficient than finite elements and is not widely used in industry. It is popular for acoustic problems [10] and is sometimes used as a hybrid method in conjunction with finite elements.

2.3 Applications of FEM in Engineering

- Mechanical/Aerospace/Civil/Automobile Engineering
- Structure analysis (static/dynamic, linear/nonlinear)
- Thermal/fluid flows
- Electromagnetics
- Geomechanics
- Biomechanics
- ...

2.4 How the FEM work

In a continuum problem of any dimension the field variable (whether it is pressure, temperature, displacement, stress, or some other quantity) possesses infinitely many values because it is a function of each generic point in the body or solution region. Consequently, the problem is one with an infinite number of unknowns. The finite element discretisation procedures reduce the problem to one of a finite number of unknowns by dividing the solution region into elements and by expressing the unknown field variable in terms of assumed approximating functions within each element. The approximating functions (sometimes called interpolation functions) are defined in terms of the values of the field variables at specified points called nodes or nodal points. Nodes usually lay on the element boundaries where adjacent elements are connected. In addition to boundary nodes, an element may also have a few interior nodes. The nodal values of the field variable and the interpolation functions for the elements completely define the behaviour of the field variable within the elements. For the finite element representation of a problem the nodal values of the field variable become the unknowns. Once these unknowns are found, the interpolation functions define the field variable throughout the assemblage of elements. Clearly, the nature of the solution and the degree of approximation depend not only on the size and number of the elements used but also on the interpolation functions selected. As one would expect, functions cannot be chosen arbitrarily, because certain compatibility conditions should be satisfied. Often functions are chosen so that the field variable or its derivatives are continuous across adjoining element boundaries. These are applied to the formulation of different kinds of elements. An important feature of the finite element method that sets it apart from other numerical methods is the ability to formulate solutions for individual elements before putting them together to represent the entire problem. This means, for example, that if we are treating a problem in stress analysis, we find the force–displacement or stiffness characteristics of each individual element and then assemble the elements to find the stiffness of the whole structure. In essence, a complex problem reduces to considering a series of greatly simplified problems. Another advantage of the finite element method is the variety of ways in which one can formulate the properties of individual elements. There are basically three different approaches. The first approach to obtaining element properties is called the direct approach because its origin is traceable to the direct stiffness method of structural analysis. The direct approach suggests the need for matrix algebra in dealing with the finite element equations. Element properties obtained by the direct approach can also be determined by the variation approach. The variational approach relies on the calculus of variations and involves extremizing a functional. For problems in solid mechanics the functional turns out to be the potential energy, the complementary energy, or some variant of these. Knowledge of the variational approach is necessary to work beyond the introductory level and to extend the finite element method to a wide variety of engineering problems. Whereas the direct approach can be used to formulate element properties for only the simplest element shapes, the variational approach can be employed for both simple and sophisticated element shapes. A third and even more versatile approach to deriving element properties has its basis in mathematics and is known as the weighted residuals approach. The weighted residuals approach begins with the governing equations of the problem and proceeds without relying on a variational statement.

This approach is advantageous because it thereby becomes possible to extend the finite element method to problems where no functional is available. The method of weighted residuals is widely used to derive element properties for non-structural applications such as heat transfer and fluid mechanics.

Regardless of the approach used to find the element properties, the solution of a continuum problem by the finite element method always follows an orderly step-by-step process. To summarize in general terms how the finite element method works we will succinctly list these steps now:

1. *Discretise the Continuum*: The first step is to divide the continuum or solution region into elements. In the example of Figure 2.4 the turbine blade has been divided into triangular elements that might be used to find the temperature distribution or stress distribution in the blade. A variety of element shapes may be used, and different element shapes may be employed in the same solution region. Indeed, when analyzing an elastic structure that has different types of components such as plates and beams, it is not only desirable but also necessary to use different elements in the same solution. Although the number and the type of elements in a given problem are matters of engineering judgment, the analyst can rely on the experience of others for guidelines.

2. *Select Interpolation Functions*: The next step is to assign nodes to each element and then choose the interpolation function to represent the variation of the field variable over the element. The field variable may be a scalar, a vector, or a higher-order tensor. Often, polynomials are selected as interpolation functions for the field variable because they are easy to integrate and differentiate. The degree of the polynomial chosen depends on the number of nodes assigned to the element, the nature and number of unknowns at each node, and certain continuity requirements imposed at the nodes and along the

element boundaries. The magnitude of the field variable as well as the magnitude of its derivatives may be the unknowns at the nodes.

3. *Find the Element Properties*: Once the finite element model has been established (that is, once the elements and their interpolation functions have been selected), we are ready to determine the matrix equations expressing the properties of the individual elements. For this task we may use one of the three approaches just mentioned: the direct approach, the variational approach, or the weighted residuals approach.

4. *Assemble the Element Properties to Obtain the System Equations*: To find the properties of the overall system modelled by the network of elements we must "assemble" all the element properties. In other words, we combine the matrix equations expressing the behaviour of the elements and form the matrix equations expressing the behaviour of the entire system. The matrix equations for the system have the same form as the equations for an individual element except that they contain many more terms because they include all nodes. The basis for the assembly procedure stems from the fact that at a node, where elements are interconnected, the value of the field variable is the same for each element sharing that node. A unique feature of the finite element method is that the system equations are generated by assembly of the individual element equations. In contrast, in the finite difference method the system equations are generated by writing nodal equations.

5. *Impose the Boundary Conditions*: Before the system equations are ready for solution they must be modified to account for the boundary conditions of the problem. At this stage we impose known nodal values of the dependent variables or nodal loads.

6. *Solve the System Equations*: The assembly process gives a set of simultaneous equations that we solve to obtain the unknown nodal values of the problem. If the problem describes steady or equilibrium behaviour, then we must solve a set of linear or nonlinear algebraic equations. If the problem is unsteady, the nodal unknowns are a function of time, and we must solve a set of linear or nonlinear ordinary differential equations.

7. *Make Additional Computations if desired*: Many times we use the solution of the system equations to calculate other important parameters. For example, in a structural problem the nodal unknowns are displacement components. From these displacements we calculate element strains and stresses. Similarly, in a heat-conduction problem the nodal unknowns are temperatures, and from these we calculate element heat fluxes.

2.5 A brief history of the method

Although the label finite element method first appeared in 1960, when it was used by Clough [11] in a paper on plane elasticity problems, the ideas of finite element analysis date back much further. In fact, the questions who originated the finite element method and when did it begin have three different answers depending on whether one asks an applied mathematician, a physicist, or an engineer. All of these specialists have some justification for claiming the finite element method as their own, because each developed the essential ideas independently at different times and for different reasons. The applied mathematicians were concerned with boundary value problems of continuum mechanics; in particular, they wanted to find approximate upper and lower bounds for Eigen values. The physicists were also interested in solving continuum problems, but they sought means to obtain piecewise approximate functions to represent their continuous functions. Faced with increasingly complex problems in aerospace structures, engineers were searching for a way in which to find the stiffness influence coefficients of shell-type structures reinforced by ribs and spars. The efforts of these three groups resulted in three sets of papers with distinctly different viewpoints. The first efforts to use piecewise continuous functions defined over triangular domains appear in the applied mathematics literature with the work of Courant [12] in 1943. Courant used an assemblage of triangular elements and the principle of minimum potential energy to study the St. Venant torsion problem.

In 1959 Greenstadt [13], motivated by a discussion in the book by Morse and Feshback [14], outlined a discretisation approach involving “cells” instead of points; that is, he imagined the solution domain to be divided into a set of contiguous sub domains. In his theory he describes a procedure for representing the unknown function by a series of functions, each associated with one cell. After assigning approximating functions and evaluating the appropriate variational principle to each cell, he uses continuity requirements to tie together the equations for all the cells. By this means he reduces a continuous problem to a discrete one. Greenstadt's theory allows for irregularly shaped cell meshes and contains many of the essential and fundamental ideas that serve as the mathematical basis for the finite element method as we know it today. As the popularity of the finite element method began to grow in the engineering and physics communities, more applied mathematicians became interested in giving the method for a firm mathematical foundation. As a result, a number of studies were aimed at estimating discretisation error, rates of convergence, and stability for different types of finite element approximations. These studies most often focused on the special case of linear elliptic boundary value problems. Since the late 1960s the mathematical literature on the finite element method has grown more than in any previous period. We shall not study the rigorous mathematical basis of the finite element method, because such knowledge is unnecessary for most practical applications. Instead we shall call upon pertinent results when they are needed. While the mathematicians were developing and using finite element concepts, the physicists were also busy with similar ideas. The work of Prager and Synge [15] leading to the development of the hyper circle method is a key example. As a concept in function space, the hyper circle method was originally developed in connection with classical elasticity theory to give its minimum principles a geometric interpretation. Outgrowths of the hyper circle method (such as the one suggested by Synge [16]) can be applied to the solution of continuum problems in much the same way as finite element techniques can be applied.

Physical intuition first brought finite element concepts to the engineering community. In the 1930s when a structural engineer encountered a truss problem such as the one shown in Figure 2.5*a*, he immediately knew how to solve for component stresses and deflections as well as the overall strength of the unit. First, he would recognize that the truss was simply an assembly of rods whose force–deflection characteristics he knew well. Then he would combine these individual characteristics according to the laws of equilibrium and solve the resulting system of equations for the unknown forces and deflections for the overall system.

This procedure worked well whenever the structure in question had a finite number of interconnection points, but then the following question arose: What can we do when we encounter an elastic continuum structure such as a plate that has an infinite number of interconnection points for example, in Figure 2.5*b*, if a plate replaces the truss, the problem becomes considerably more difficult. Intuitively, Hrenikoff [17] reasoned that this difficulty could be overcome by assuming the continuum structure to be divided into elements or structural sections (beams) interconnected at only a finite number of node points. Under this assumption the problem reduces to that of a conventional structure, which could be handled by the old methods. Attempts to apply Hrenikoff's "framework method" were successful, and thus the seed to finite element techniques began to germinate in the engineering community. Shortly after Hrenikoff, McHenry [18] and Newmark [19] offered further development of these discretization ideas, while Kron [20, 21] studied topological properties of discrete systems. There followed a ten-year spell of inactivity,

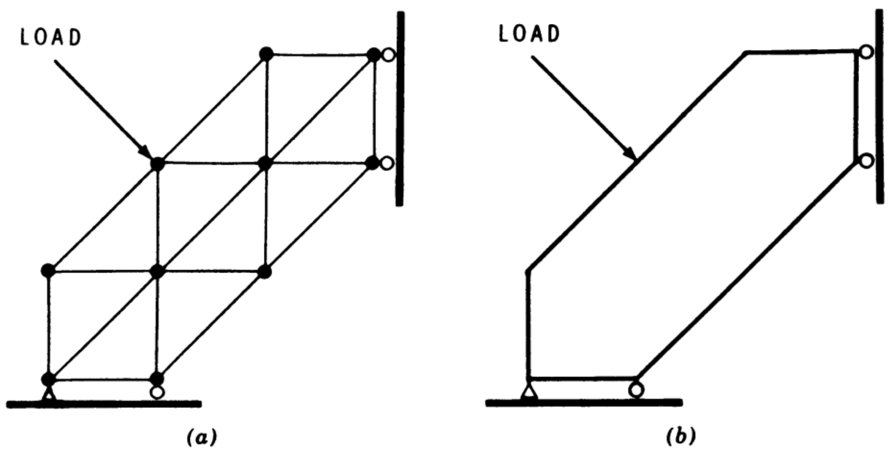


Figure 2.5 A truss problem Example of (*a*) a truss and (*b*) a similarly shaped plate supporting the same load.

which was broken in 1954 when Argyris and his collaborators [22–26] began to publish a series of papers extensively covering linear structural analysis and efficient solution techniques well suited to automatic digital computation? The actual solution of plane stress problems by means of triangular elements whose properties were determined from the equations of elasticity theory was first given in the now classical 1956 paper of Turner, Clough, Martin, and Topp [27]. These investigators were the first to introduce what is now known as the direct stiffness method for determining finite element properties. Their studies along with the advent of the digital computer at that time opened the way to the solution of complex plane elasticity problems. After further treatment of the plane elasticity problem by Clough [11] in 1960, engineers began to recognize the efficacy of the finite element method. In a 1980 paper Clough [28] gives his personal account of the origins of the method, describing the sequence of events from the original efforts at Boeing that produced reference [23] to the paper [11] in which he introduced the label of the finite element method. In 1965 the finite element method received an even broader interpretation when Zienkiewicz and Cheung [29] reported that it is applicable to all field problems that can be cast into variational form. During the late 1960s and early 1970s (while mathematicians were working on establishing errors, bounds, and convergence criteria for finite element approximations) engineers and other practitioners of the finite element method were also studying similar concepts for various problems in the area of solid mechanics. In the years since 1960 the finite element method has received widespread acceptance in engineering. Thousands of papers, hundreds of conferences, and many books appeared on the subject. The number of books published over this period illustrates the exponential growth. A 1991 bibliography [30] lists nearly 400 finite element books in English and other languages. The bibliography also identifies over 200 international finite element symposia, conferences, and short courses that took place between 1964 and 1991.

2.6 Commercial finite element software

The first commercial finite element software made its appearance in 1964. The Control Data Corporation sold it in a time-sharing environment. No preprocessors (mesh generators) were available, so engineers had to prepare data element by element and node by node. A keypunched IBM (Hollerith) card represented each element and each node. Batch-mode line plots were used to check geometry and to post-process results. Turnaround occurred in days for simple problems. Only linear problems could be addressed. Nevertheless it represented a breakthrough in the complexity of the problem that could be handled in a practical time frame. Later, finite element software could be purchased or leased to run on corporate computers. Typically the corporate computer had been purchased to process financial data, so that computer availability to the engineer was restricted, perhaps to nights and weekends. The introduction of workstations circa 1980 brought several breakthrough advantages. Interactive graphics were practical and availability of computer power to solve problems on a dedicated basis was achieved. Finally, the introduction of personal computers (PCs) powerful enough to run finite element software provides extremely cost effective problem solving. Today we have hundreds of commercial software packages to choose from. A small number of these dominate the market. It is difficult to make comparisons purely on a finite element basis, because the software houses are often diversified. Data from Daratech suggest that the companies listed in Table 1.1 are dominant providers of general-purpose finite element software. Choice among these, or other providers, involves a complex set of criteria, usually including: analysis versatility, ease of use, efficiency, cost, technical support, training, and even the labour pool locally available to use particular software. In contrast to the early days, we can now use computer-aided design (CAD) software or solid modellers to generate complex geometries, at either the component or assembly level. We can (with some restrictions) automatically generate elements and nodes, by merely indicating the desired nodal density. Software is available that works in conjunction with finite elements to generate structures of optimum topology, shape, or size. Nonlinear analyses including contact, large deflection, and nonlinear material behaviour are routinely addressed.

Company Name	Product Name	Web Site
Hibbitt, Karlson & Sorenson	ABAQUS	www.hks.com
Ansys, Incorporated	ANSYS	www.ansys.com

Structural Data Research Corp.	SDRC-Ideas	www.sdrc.com
Parametric Technology, Inc.	RASNA	www.ptc.com
MSC Software Corp.	MSC/NASTRAN	www.mscsoftware.com

TABLE 2.1 Leading commercial finite element software companies

2.7 The Future of the FEM

Our brief look at the history of the finite element method shows us that its early development was sporadic. The applied mathematicians, physicists, and engineers all dabbled with finite element concepts, but they did not recognize at first the diversity and the multitude of potential applications. After 1960 this situation changed and the tempo of development increased markedly. By 1972 the finite element method had become the most active field of interest in the numerical solution of continuum problems. It remains the dominant method today. Part of its strength is that it can be used in conjunction with other methods. Software components such as solvers can be used in a modular fashion, so that improvements in diverse areas can be rapidly assimilated. Having said that, we can still remark that major innovations in technology Therefore, academic publications are the best leading indicator of what's to come in commercial finite elements.

Certainly, improved iterative solvers, meshless formulations, better error indicators, and special-purpose elements are on the list of things to come. Although the finite element method can be used to solve a very large number of complex problems, there are still some practical engineering problems that are difficult to address because we lack an adequate theory of failure, or because we lack appropriate material data. This is not a finite element problem per se, but is of serious concern to any engineer who wants to supplant testing with analysis. (The use of analysis usually permits faster design turnaround, the exploration of widely varying environments, and the use of optimization tools. Furthermore, analysis is usually significantly cheaper than building prototypes and testing them.) The mechanical and thermal properties of many non metallic materials are difficult to acquire, especially over a range of temperatures. Fatigue data is often lacking. Fatigue failure theory often lags our ability to calculate changing complex stress states. Data on friction is often difficult to obtain. Calculations based on the assumption of Coulomb friction are often unrealistic. There is a general paucity of thermal data, especially regarding absorptivity and emissivity needed for radiation calculations. The World Wide Web should offer a means of placing material properties into accessible databases. From a practitioner's viewpoint, the finite element method, like any other numerical analysis technique, can always be made more efficient and easier to use. As the method is applied to larger and more complex problems, it becomes increasingly important that the solution process remain economical. The rapid growth in engineering usage of computer technology will undoubtedly continue to have a significant effect on the advancement of the finite element method. Improved efficiency achieved by computer technology advancements such as parallel processing will surely occur. Since the mid 1970s interactive finite element programs on small but powerful personal computers and workstations have played a major role in the remarkable growth of computer-aided design.

With continuing economic pressures to improve engineering productivity, this decade will see an accelerated role of the finite element method in the design process. This methodology is still exciting and an important part of an engineer's tool kit.

2.8 Theoretical Background

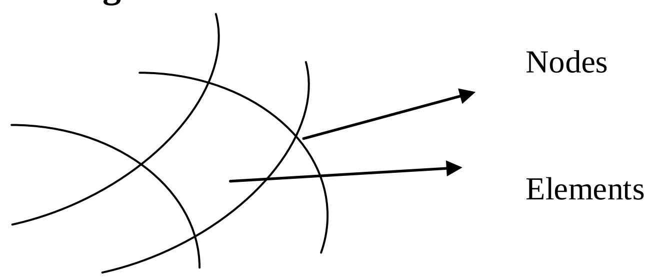
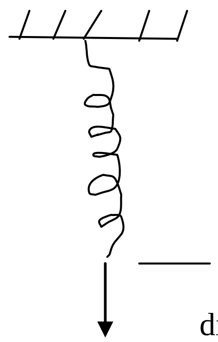


Figure 2.6 Explanations of node & element

Consider a figure above we have lines meet at certain point in other words bodies which we consider for analysis have been discretised what we called an element and they are bounded by nodes which sit an intersection of elements, nodes and element are important in finite element method analysis, what we have done is that a complex structure has been divided into elements, essentially what we are looking in a design is that we have a force, body and deformation what we need is an relationship between force and deformation let us say for time being this deformation as displacement. When there is force at one end and displacement at other end stiffness will come into picture, suppose we have spring for example

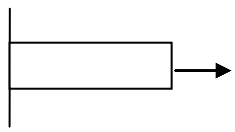


P-Force applied, u

displacement

Figure 2.7 Spring element

To find out the displacement of the spring it undergoes on the application of the force p, the stiffness of the spring must be known, then P=Ku, where k is stiffness of the spring



p

Figure 2.8 Bar element

As we know that

Σ = E€ or (2.1)

P/A=E. (Change in length/ Original Length) (2.2)

P/A=E. (u/L)

P = (AE/L). U (2.3)

After comparing equation of force of spring we can say that stiffness of a bar is (AE/L), so stiffness is not a concept restricted to spring as it is extended in this case to bar.

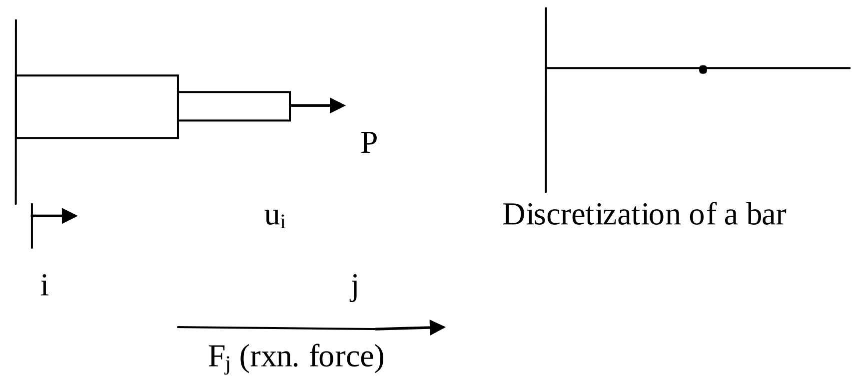


Figure 2.9 Discretization of a bar

Stiffness matrix of a particular bar is defined by area A, length L, Young Modulus E as we know that F_i= (AE/L).u_i and F_j= -F_i , now fix i and move node j by amount u_j, this will result in F_j= (AE/L). u_j and as we know that now F_i will be the reaction force i.e. F_i=-F_j= -(AE/L).u_j

Now by superposition theorem, the result we will get

$$F_i = (AE/L)\,[u_i - u_j] \qquad (2.4)$$

$$F_j = (AE/L)\,[-u_i + u_j] \qquad (2.5)$$

Now we will solve this problem for spring with two nodes as shown below

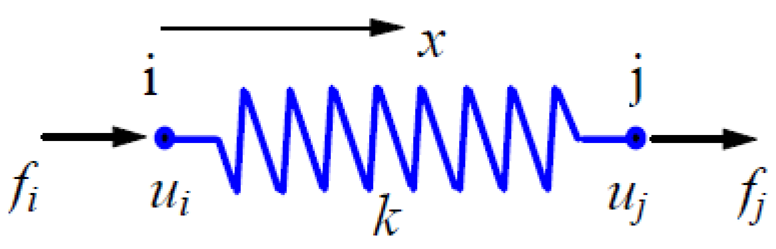


Figure 2.10 Spring element

Two nodes: i, j

Nodal displacements: u_i, u_j (in, m, mm)

Nodal forces: f_i, f_j (lb, Newton)

Spring constant (stiffness): k (lb/in, N/m, N/mm)

Spring force-displacement relationship:

$F = k\,\Delta$ with $\Delta = u_j - u_i$

$K = F/\Delta$ (> 0) is the force needed to produce a unit stretch.

Consider the equilibrium of forces for the spring. At node i, we have

$$f_i = -F = -k(u_j - u_i) = k\,u_i - ku_j \quad (2.6)$$

$$f_j = -F = k(u_j - u_i) = -ku_i + ku_j \quad (2.7)$$

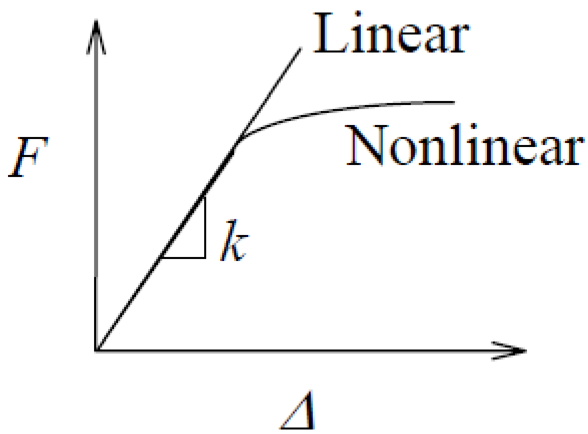


Figure 2.11 Force vs. elongation graph

$K = F/\Delta$ (> 0) is the force needed to produce a unit stretch.

In matrix form,

$$\begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{pmatrix} u_i \\ u_j \end{pmatrix} = \begin{pmatrix} f_i \\ f_j \end{pmatrix} \quad (2.8)$$

Or,

$$k.u = f \quad (2.9)$$

k = (element) stiffness matrix

u = (element nodal) displacement vector

f = (element nodal) force vector

Bar Element

Consider a uniform prismatic bar:

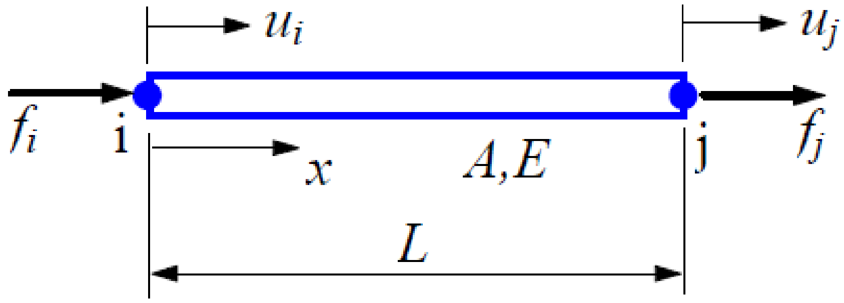


Figure 2.12 Uniform prismatic bar

L	length
A	cross-sectional area
E	elastic modulus
u=u(x)	displacement
ε=ε(x)	strain
σ=σ(x)	stress

Strain-displacement relation:

$$\varepsilon = du/dx \quad (2.10)$$

Stress-strain relation:

$$\sigma = \varepsilon E \quad (2.11)$$

Assuming that the displacement u is varying linearly along the axis of the bar i.e.

$$u(x) = \left\{1 - \frac{x}{L}\right\}u_i + \frac{x}{L}\,u_j \quad (2.12)$$

We have

$$\in = \frac{u_j - u_i}{L} = \frac{\Delta}{L} \quad (2.13)$$

(Δ = elongation)

We also have

$$\sigma = E \in = \frac{E\Delta}{L} \quad (2.14)$$

$$\sigma = \frac{F}{A} \quad (2.15)$$

(F= force in bar)

$$F = \frac{EA}{L}\Delta = k\Delta \quad (2.16)$$

Where $k = \frac{EA}{L}$ is the stiffness of the bar.

The bar is acting like a spring in this case and we conclude that element stiffness matrix is

$$\mathbf{k} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix}$$

Or

$$k = \frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (2.17)$$

This can be verified by considering the equilibrium of the forces at the two nodes.

Element equilibrium equation is

$$k = \frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{pmatrix} u_i \\ u_j \end{pmatrix} = \begin{pmatrix} f_i \\ f_j \end{pmatrix} \quad (2.18)$$

Degree of Freedom (dof)

Number of components of the displacement vector at a node.

For 1-D bar element: one dof at each node.

Physical Meaning of the Coefficients in k

The j^{th} column of k (here j = 1 or 2) represents the forces applied to the bar to maintain a deformed shape with unit displacement at node j and zero displacement at the other node. Now we will implement boundary condition as u_i =constant, it will be constant for fixed end in this case it will be zero, by putting these values we can calculate the unknown variables in this case it will be displacement, which will further used for calculation of stresses, in case of complex problems we will use stress as tensor quantity and resolve into three axes direction, then we will also define tensor vector. Hence it can be said that FEM is an overall very efficient method to obtain the solution of the very complex problem without compromising on quality and safety; the next section discusses the various case studies where FEM has helped to achieve the solution effectively.

2.9 Applications

This section presents a brief overview of FEM in different fields with a brief literature review of FEM in each application.

2.9.1 FEM in Biomechanics

The word biomechanics can be thought of to be consisting of two words; bio and mechanics, where the word bio is a prefix for living organism and mechanics is a study of the action of forces on mechanical systems. Therefore it can be said that biomechanics is a multidisciplinary study which includes:

i) Biological material physical properties
ii) Biomechanics application
iii) Biological signals measurement and analysis
iv) Biomechanical modelling and simulation

2.9.1.1 Muscoskeletal

FEM has found several applications in the field of bio-mechanics. The most prominent ones lie in modelling and analysis of musco-skeletal system. During survey of the literature in this field it was found that a large number of researchers have reported their work in prestigious journals. A few are being reported for the ready reference. Taddei et. al (2006) analyzed that how the modelling uncertainties in the geometry and properties of a material of the human bone affect the predictions of a finite element model which is derived from computed tomography(CT) data. In this paper for sensitivity analysis based on Monte Carlo method, three femur models were generated from CT datasets and each is subjected to two different loading conditions. The bone tissue's density, geometry and the mechanical properties were considered as random input variables. In biomechanics usually finite element results used in research of biomechanics were considered as statistical output variable. The sensitivity of the statistical output variable is assessed to the variability in the input variables. It is not possible to define before the influence of the errors due to the geometry definition process and material assignment process on the result obtained from the finite element analysis. As expected the geometrical representation error of the bone are for the stresses always a dominant variable. The result seems to be dependent on the loading conditions for all the variables and it varies from subject to subject. However the most interesting result is that by using a proposed method to build a femur model by finite element method from a CT dataset, in clinical practice the achievable quality, the output's coefficient of variation never exceeds the 9%.The proposed method is robust enough for investigation of the mechanical behaviour of the bones [31].

Musculoskeltal simulation has very important applications in biomedical engineering, biomechanics, computer graphics and surgery simulation. Muscle, tendon geometry and bone also, the accuracy of dynamic deformation of the tendon and muscle are of prime importance in all applications mentioned above. A framework is presented by Teran et al. for extraction and high resolution simulation from the visible human data set of musculoskeletal geometry. A simulation of upper limb's 30 contact/collision coupled muscle is done also, the geometry of muscle is embedded in a non manifold which help in reducing the computational cost. All this simulation is done using the finite element techniques which can handle both inverted tetrahedral and degenerate [32].

2.9.1.2 Needle insertion

In this paper simulation of virtual needle insertion is presented by DiMaio and Salcudean. These models of simulations are based on insertion forces of needle and measured planar tissue deformations. Along the needle shaft, only interest is in force-displacement relationship, linear simulation models computational complexity has been significantly reduced by the condensation technique. The tissue and needle motion change is determined by boundary conditions which changes when needle penetrates or withdrawn from the model of the tissue. Changes in the local material coordinate and boundary conditions are facilitated by updates in low-rank matrix updates. Tissue models and large strain elastic needle model are coupled together to account for deflection in the needle and simulated insertion bending [33].

Piercing process is studied by Yong and Zhen by simulation and modelling of piezodriven intracytoplasmic sperm injection. This modelling and simulation is done using the material point method (MPM). It is revealed by MPM simulation that in the piercing process a key role is played by the lateral vibration of the injection pipette and the claim is disproved that the zona piercing is done by axial displacement of the pipette. Further analyisation of lateral vibration of the injection pipette lateral vibration is done using the finite element method to investigate the role of mercury in process of piercing. Pipette tip amplitude is generally reduced by the use of mercury and smaller vibration amplitude is resulted due to that shorter piezopulse duration. Hence this implies that mercury effect on the piezo-ICSI procedure may cause less damage to the oocyte due to the pipette tip reduced amplitude, and by changing the injection micropipette dimensions and control parameters of piezodrill piezo-ICSI procedure development is possible without using mercury [34].

2.9.1.3 Miscellaneous

Comparison between the uncompressed volumetric data and compressed projection mammographic data offers a challenge to accurately localize anatomy in both data sets. Simulation of mechanical compression of volumetric breast data is presented in this work by the Kellner et al. A rectilinear-grid finite element mesh approach is used and this method is applied to the volumetric breast data. A good agreement of results with theory is achieved. Also, with clinical results qualitatively agreement is achieved. Presented method provides a simulation method of high quality for mechanical compression of breast data [18].

A description of the technique for the reconstruction of the breast tissue elasticity modulus is given by the Samani et al. which assumes that from a contrast- enhanced magnetic resonance image geometry is available of normal and suspicious tissue. Also, one assumption is taken that throughout each tissue volume modulus is constant. Quasi static strain data is an iterative technique in which by each iteration modulus updating is done followed by calculation of stress. It is assumed that breast mechanical simulation is done by two compressional rigid plates. Based on well defined boundary conditions of the compressional rigid plates finite element method is used for the breast mechanical simulation. Based on Hooke's law updating of modulus is done element- by- element. Modulus reconstruction of breast tissue by simulated data and phantom modulus reconstruction using experimental data indicates that this is a robust technique [35].

2.9.2 Stress analysis of a LP rotor

The objective of above application re as follows:

- To determine the stresses in the existing rotor
- To modify the defective rotor geometry and limit the stress to existing value
- Consider centrifugal loading in analysis

During the manufacture of a rotor there was a small mistake that is done during CNC programming LP rotor was designed improperly so a small hole was created as shown in fig. below

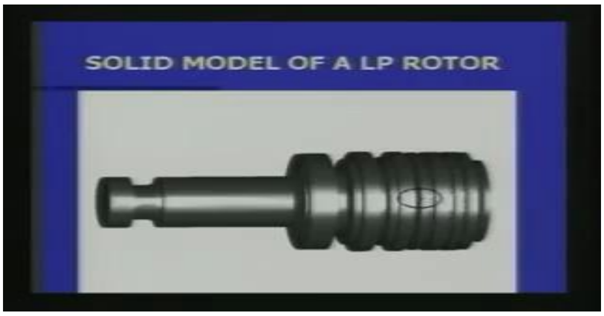


FIGURE 2.13 Solid model of LP ROTOR

The problem is that in this case, the hole which is happen to be at a point where stresses are also maximum but this rotor is very expensive, it is about 10 crore and time need to manufacture this rotor is nearly six months which means a project of thermal power plant comes to standstill for six months, which is of course not acceptable.

The question is this can we use computational method to find out whether this mistake is tolerable or not, if not can we modify it that we don’t compromise on quality and safety of the project. The answer is yes, by using FEM we will first analysis the good LP rotor then by comparison we will modify the design of defective rotor at a point of maximum stress, it has been done in a matter of about month, this a power of computational method.

Connecting rod mechanism is an important consideration in the design of Diesel Engine. But it is difficult to describe the dynamic changes of boundary conditions of the running engine by the traditional method of simply finite element analysis. In order to obtain the vibration characteristics and vibration frequency distributions, structural characteristics of the connecting rod mechanism using modal analysis is investigated. Firstly, a physical model of connecting rod mechanism is built using CAD software. Secondly finite element analysis and simulation of the model is taken by hyperworks and MSC. Nastran softwares. Then its flexible multi-body dynamic model is established by ADAMS/View. And the fatigue stress of connecting rod under the max combustion pressure and Inertia force condition is calculated using the durability Module. The result obtained by Qinghui et al. indicates the stress distribution and deformation instance. The stress is mainly produced on the joint of connecting rod shell and the bottom end or the top end. And the biggest stress acting on the connecting rod is just 34.0613MPa, early smaller than its limited stress 355MPa. The method provides the theoretical evidence for connecting rod structure improvement and optimum design.

A polarizing plate, which is an important part of a liquid crystal display panel (LCD), is made by sandwiching an organic polarizer between protecting films. An organic polarizer is both a hygroscopic and orthotropic material. The hygroscopic swelling and drying shrinkage of the organic polarizer can cause the polarizing plate to crack and the liquid crystal display panel to warp. The diffusion coefficient and Henry‘s law coefficient were measured using a thermo-gravimetric analyzer (TGA) under controlled humidity, while the coefficient of moisture expansion (CME) was measured using a thermo-mechanical analyzer (TMA), also under controlled humidity.

The thermo-mechanical and hygro-mechanical deformation of a polarizing plate was analyzed by Mizutani et al. using the finite element method (FEM). This analysis was performed as follows. The distribution of the moisture concentration was analyzed according to Fick‘s law. The equation of Fick‘s law is similar to that of the transient heat conduction, and the FEM for the transient heat conduction was utilized for the transient diffusion analysis. The hygro-mechanical analysis was then carried out in a way similar to the thermal stress analysis. Thermal stress was analyzed separately using the FEM. Finally, the obtained hygro-mechanical strain and stress were added to the thermal strain and stress, respectively. The analyzed displacement of a polarizing plate using the CMEs of a polarizer and protecting films corresponds to the measured displacement.

The warpage of a liquid crystal display panel sometimes causes light leakage along the frame of the display panel due to contact of the display panel with the bezel of the frame. The warpage was analyzed according to the thermo-mechanical strain and the hygro-mechanical strain.

Based on finite element method and genetic algorithms dynamic mathematical model is established by Meng Lei and Zhang Shu, and the simulation of stress distribution around the defects of single crystal nickel-based superalloys is also established with ANSYS. After the change of stress field with time is analyzed, the result is compared with that achieved through numerical calculation and experimental analysis. The comparison shows that the combination of Finite element method and genetic algorithm is an effective way to micro-simulation. a model is established and simulated for nickel-based superalloys by finite element Method tool, the paper also analysis the stress field of microstructure defects. An Optimization Method is used to the Simulation by Genetic Algorithm, and the results are compared with experimental analysis. The comparison shows the model and Algorithm are correct and can provide basis for the study of creep features and microstructure evolution [36].

This paper presented by Moore, T.D and Jarvis, J.L. examines one of the common modes of structural failure in multichip ball grid arrays (BGAs), determines its locations within the package structure, relates it to the stresses generated in the reliability tests under which it occurs, and by finite element simulations, determines an explanation for the failure, and finally proposes a method to avoid this failure mechanism. Several designs of multichip BGA substrates were manufactured and production silicon assembled into them. These were all 14 mm×22 mm 119 ball PBGAs. These were subjected to a set of package reliability tests, until some units failed electrical test. The failed units were analyzed and the physical location and shape of the failure was determined in many cases. From this information, the mechanical mode of failure for each unit was determined. In addition there was sufficient information in some of the analyses to provide definite suggestions as to the mechanism of failure. Meanwhile, finite element analysis was performed using simplified representations of the multichip BGAs, in order to find the locations of highest stress, and the expected modes of failure. This data was matched to the failure modes found in the physical analysis. Some novel failure analysis techniques were used to expose the damage in the failed units. A particular failure mode occurred frequently in temperature cycle, and the sites of failure were located by failure analysis. The failure was due to open circuit in the copper tracks in the top layer of the substrate caused by cracking in the solder resist directly underneath the edge of the die attach fillet. Finite element analysis was carried out and the location of the actual failures was found to be a local zone of high tensile stress in the solder resists [37].

In this study by the Dong Kil Shin and Jung Ju Lee, the coefficient of thermal expansion (CTE) and the elastic modulus of epoxy moulding compound (EMC) are measured using fabricated specimens and then the measured values are compared with the predicted values by theoretical equations (such as dilute suspension method, self consistent method, Hashin-Shtrikman's bounds, Shapery's bounds and others). The measured values are distributed within the upper and lower bounds of predicted values. The measured elastic modulus and the CTE of EMC approach close to the predicted values by self consistent method and upper bound of Shapery's equation respectively. Two-dimensional (2-D) and three-dimensional (3-D) finite element analysis are performed using the measured and analytically predicted values. Finite element method (FEM) analysis indicates that firstly the EMC with eighty weight percentage of filter shows less thermal stress when package is cooling down and relatively high thermal stress when package is heating up. Secondly the stress concentrations at the edge sections about two times higher than the interfaces and at the vertex parts about 1.4 times higher than the edge sections are observed [37].

The effect of stem size and the type of fixation of the prosthesis on the stress distribution in the femoral part of the knee joint is studied here. In this investigation a two dimensional model of a section in the external condyle of distal femur is considered for stress analysis. This model is solved by using the finite element method and by the aid of SAP90 software. The stress analysis and shape deformation are done in 8 different models and each model under 4 different flexion conditions: namely 0, 30, 60 and 90 degrees. By assuming that the constituting elements are made of isotropic material the following results are obtained. Short length prosthesis causes an increase in the stress level of the cortical bone of in the posterior part of the shaft. Prosthesis with a medium sized stem produces a stress distribution similar to the no prosthesis case. The application of a long size stem decreases the stress level in the cortical bone of the posterior part of the shaft. For prosthesis with no stem and no cement an increase in the stress level in the cortical bone of the posterior part of the shaft is seen. Furthermore, implanting the prosthesis with cement causes a greater reduction in the stress level [38].

The development of modern transmission has leaded to transmission gears scuffing becoming more serious. The finite element method (FEM) is proposed by Jie et al. that can be used as the most economical and effective designing method to predict the gears temperature, determining whether the tooth scuffing or not, and analysis the thermal stress and the deformation of gears. As simulating the heat generation process of the friction in the gear engagement, the two kinds of flash temperature models are developed, which were the Block flash temperature model and the gear flash temperature model. The ISO and predicted result are compared for situations where gears engaged at different points to demonstrate the both two thermal models could calculate the gear transient temperature rise accurately. While through comparing the two models, it is found the gear flash temperature model compared to Block model, which is not only much closer to the actual gear tooth and the form of the heat load, but could superimpose gear bulk temperature model directly, so it could be more comprehensive analysis of the real temperature distribution of the gear, and solve the gear interfacial contact temperature more accurately.

Copper (Cu)/low-dielectric constant (K) structures are desired choices for advanced integrated circuits (ICs) as the IC technology moving towards fine pitch, high speed, increased integration and high performance. Copper interconnects with low-k dielectric material improve the performance of ICs by reducing interconnect the RC delay, the cross talk between the adjacent metal lines and the power loss. However, the packaging of Cu/low-k IC device is a challenge for the packaging industry to integrate these devices without any failure during assembly and reliability. The current work Rao et al. presents, 1) the finite element model (FEM) based parametric study on Cu/low-K wafer level package (WLP) reliability and stresses on Cu/low-K layers, and 2) experimental validation of WLP reliability by fabricating the test chips. FEM modelling and simulation results have shown that high aspect ratio interconnects, thinner die, and thinner printed circuit board can reduce the stress in low-k layer and enhance the board level interconnect reliability. Test chip of 7 mm × 7 mm size is designed with 128 input/output (I/O) off-chip interconnects at 300μm pitch in two depopulated rows using redistribution layers (RDL). Test chips are fabricated on 200-mm-diameter wafer with blanket black diamond (BD) low-K layers structure. Two different Pb free solder interconnects, thick copper column of 100μm height with SnAg solder cap and SnAg solder bump of 150μm height with 5μm-thick copper under bump metallurgy (UBM), are fabricated. The Cu/low-K test chips are assembled onto a two layer high glass transition temperature (Tg) FR-4 substrate using two different types of no-flow underfills (NFU) to build the test vehicles and assembled test vehicles are subjected to various JEDEC standard reliability tests, and related failure analysis is carried out. Cu/low-k WLP with copper column interconnects without no-flow underfill passed 1000 h high-temperature storage (HTS) test, and passed the JEDEC drop test- - with no-flow underfill. Thin die test vehicles of Cu column interconnects with no-flow underfill and extra solder shown better thermal cycling (TC) performance and the board level TC performance can be improved further using thicker RDL [40].

The design of electrical rotating machine is required to have high capacity and electrical loading without comparable increase in size. In this paper, kim et al. looked into the electromagnetic force of end winding part of electrical motor by using 3D finite element method to analyze distribution of end winding forces more exactly. And we also performed stress analysis of end windings by using the force distribution calculated by 3D electromagnetic field FEM. Finally, the design of support ring surrounding end windings and preventing the insulation fault is presented by comparing the stress distribution of supporting ring with the yield stress of ring material. [41]

2.9.3 Non-destructive analysis of metallic tubes

Metallic tubes constitute important components of many kinds of industrial plants, such as gas pipelines, chemical pipelines, fuel vessels, sugar and alcohol plants etc. Generally, these walls are subject to the aggressive (corrosive) actions by fluids contained by them, or even by atmospheric agents. So, these equipments must be periodically evaluated in order to avoid operational interruptions and/or dangerous accidents. Usually, these evaluations arc did using non-destructive techniques. Such techniques may involve the use of electromagnetic fields, which are induced in the metallic walls of the equipment under inspection.

More common techniques used in the inspection of metallic walls are based on eddy current systems. In this kind of analysis, the electromagnetic devices are excited by an alternating current of a given frequency that induces a flow of eddy currents in the material beneath them. As the probe passes over the defect, variations occur in the flow of eddy currents. These variations are then detected by electronic sensors. The change in the flow of eddy currents as the probes pass over the defect is generally proportional to the depth of the defect, and makes possible to estimate the depth of the defect by proper electronic calibration. Relative motion between the test probe and the material being inspected is a requirement for this kind of analysis. Although the probe can be hand held as the piece under test is examined, this method is usually too slow and unreliable.

A very interesting alternative introduced by Low is the use of the Finite Element Method (FEM) in conjunction with Artificial Neural Networks (ANN) in the solution of this kind of inverse problem [44].

The methodology consists of the following steps:

1. A large number of defects in a metallic tube is simulated using the finite element method.
2. The obtained results are then used to generate the training vectors of a multilayer perceptron artificial neural network.
3. The trained network is used to classify new defects in the tube, which do not belong to the original dataset.
4. The network weights can be embedded in an electronic device, and used to identify defects in real pieces, with similar characteristics to those of the simulated ones.

For the methodology presented here, the measured values are independent of the relative motion between the probe and the piece under test. In other words, the movement is necessary only to change the position of the probes, to acquire new field values, which are necessary for the identification of new defects. Hence the association of FEM and ANN techniques seems to be a useful alternative for non-destructive evaluations.

A linear hybrid stepping motors (LHSMs) have been widely used owing to its simple structure and ripple-free holding force at aligned position. Despite its attractive features, the LHSM delivers the significant thrust vibrations during position-to-position movement that are the dominant cause of the positioning error, mechanical stress, and acoustic noise. In order to overcome this defect, Hwang et al. propose an active control scheme to damp the vibration for $\pi/4$-multiple-pitched LHSM by the feed-forward compensation signal. Utilizing a reluctance network and FEM analysis that incorporate the factors of non-uniform air gap, manufacturing tolerance, and nonlinear material properties, the LHSM with force ripple components is modelled as a nonlinear position-dependent function. The damping force signal is estimated from the Jacobian linearization observer and the positioning accuracy is significantly improved through a closed-loop control scheme for restraining the thrust ripple [42].

Wang et al. in this paper purposed the finite element method as an excellent tool for crystal resonator analysis in this paper FEM has been applied to quartz crystal resonator engineers due to its long history in research based on Mindlin plate and three-dimensional approaches for piezoelectric plates with considerations of complication factors such as electrodes, thermal effect, mounting, and packaging, among others. The earlier efforts have been dampened, however, by the fact that the high vibration frequency in the thickness shear mode has caused significant increase of the problem size in terms of number of equations, or total degree of freedom, in the linear system resulted. It is typical that an accurate analysis of quartz crystal resonator vibrating at the fundamental thickness shear mode may require solving a linear system around one million for both free and forced vibrations. What we can get from the analysis are the frequency spectra, which are the relationship between frequencies and geometry, and mode shapes of the resonator structure with mountings. This, not surprisingly, is beyond the computing capabilities for many industrial engineers who do not have access to resources like supercomputers widely available to academic researchers. On the other hand, the finite element method is an excellent tool for crystal resonator analysis we should utilize for quick and precise prototyping process. In order to overcome the challenge of computing power, the finite element program development has been taking steps through employing Mindlin plate theory, efficient Eigen value solvers, and sparse matrix handling algorithms, as ways to aggressively improve the efficiency and reduce stringent requirements on hardware. In this continuing research, we have replaced our earlier version of the finite element program with ARPACK as the Eigen value solver, added sparse matrix handling functions, and implemented parallel computing capability on a cost-effective Linux cluster. The computing capability has been improved with the utilization of multiple processors significantly while the cost of infrastructure and operations is within the reach of the frequency control industry. We shall introduce the core improvements on algorithms, parallel features and implementation, hardware requirements, and computing power gain based on our current program configuration. The program has been developed in partnership with major industry players to demonstrate the effectiveness of supercomputing and finite element method as an important design tool for resonator design and improvement [43].

In this MATLAB Pdetoolbox is used this section briefly describe about this toolbox which is used to solve partial differential equations.

2.10 Brief introduction of Pdetoolbox

The objectives of Partial Differential Equation Toolbox™ software are to provide tools that:

• Define a PDE problem, e.g., define 2-D regions, boundary conditions, and PDE coefficients.

• Numerically solve the PDE problem, e.g., generate unstructured meshes, discretize the equations, and produce an approximation to the solution.

• Visualize the results.

The basic equation of Partial Differential Equation Toolbox is the PDE

$$-\nabla.(c\nabla u) + au = f \qquad (2.19)$$

Expressed in Ω, which shall be referred as the elliptic equation, regardless of whether its coefficients and boundary conditions make the PDE problem elliptic in the mathematical sense. Analogously, the terms used should be parabolic equation and hyperbolic equation for equations with spatial operators like the previous one, and first and second order time derivatives, respectively. Ω is a bounded domain in the plane. *c*, *a*, *f*, and the unknown *u* are scalar, complex valued functions defined on Ω. *c* can be a 2-by-2 matrix function on Ω. The toolbox can also handle the parabolic PDE

$$d\frac{\partial u}{\partial t} - \nabla.(c\nabla u) + au = f \qquad (2.20)$$

the hyperbolic PDE

$$d\frac{\partial^2 u}{\partial t^2} - \nabla.(c\nabla u) + au = f \qquad (2.21)$$

and the Eigen value problem

$$-\nabla.(c\nabla u) + au = \lambda du \qquad (2.22)$$

Where *d* is a complex valued function on Ω, and λ is an unknown Eigen value. For the parabolic and hyperbolic PDE the coefficients *c*, *a*, *f*, and *d* can depend on time. A nonlinear solver is available for the nonlinear elliptic PDE

$$-\nabla.(c(u)\nabla u + a(u)u = f(u) \qquad (2.23)$$

Where *c*, *a*, and *f* are functions of the unknown solution *u*.

All solvers can handle the system case

$$-\nabla.(c_{11}\nabla_{u1}) - \nabla.(c_{12}\nabla_{u2}) + a_{11}u_1 + a_{12}u_2 = f_1 \qquad (2.24)$$

$$-\nabla.(c_{21}\nabla_{u1}) - \nabla.(c_{22}\nabla_{u2}) + a_{21}u_1 + a_{22}u_2 = f_2 \qquad (2.25)$$

Systems of arbitrary dimension from the command line can be worked upon. For the elliptic problem, an adaptive mesh refinement algorithm is implemented. It can also be used in conjunction with the nonlinear solver. In addition, a fast solver for Poisson's equation on a rectangular grid is available.

The following boundary conditions are defined for scalar *u*:

- Dirichlet: $hu = r$ on the boundary $\partial\Omega$.
- Generalized Neumann: $\vec{n}.(c\nabla u) + qu = g$ on $\partial\Omega$.

$\vec{n}$ is the outward unit normal. *g*, *q*, *h*, and *r* are complex-valued functions defined on $\partial\Omega$. (The Eigen value problem is a homogeneous problem, i.e., $g = 0$, $r = 0$.) In the nonlinear case, the coefficients *g*, *q*, *h*, and *r* can depend on *u*, and for the hyperbolic and parabolic PDE, the coefficients can depend on time. For the two-dimensional system case, Dirichlet boundary condition is

$$h_{11}u_1 + h_{12}u_2 = r_1 \qquad (2.26)$$

$$h_{21}u_1 + h_{22}u_2 = r_2 \qquad (2.27)$$

the generalized Neumann boundary condition is

$$\vec{n}.(c_{11}\nabla_{u1}) + \vec{n}.(c_{12}\nabla_{u2}) + q_{11}u_1 + q_{12}u_2 = g_1 \qquad (2.28)$$

$$\vec{n}.(c_{21}\nabla_{u1}) + \vec{n}.(c_{22}\nabla_{u2}) + q_{21}u_1 + q_{22}u_2 = g_2 \qquad (2.29)$$

and the mixed boundary condition is

$$h_{11}u_1 + h_{12}u_2 = r_1 \qquad (2.30)$$

$$\vec{n}.(c_{11}\nabla_{u1}) + \vec{n}.(c_{12}\nabla_{u2}) + q_{11}u_1 + q_{12}u_2 = g_1 + h_{11}\mu \qquad (2.31)$$

$$\vec{n}.(c_{21}\nabla_{u1}) + \vec{n}.(c_{22}\nabla_{u2}) + q_{21}u_1 + q_{22}u_2 = g_1 + h_{12}\mu \qquad (2.32)$$

where μ is computed such that the Dirichlet boundary condition is satisfied. Dirichlet boundary conditions are also called essential boundary conditions, and Neumann boundary conditions are also called natural boundary conditions. See Finite Element Method for the general system case.

The PDEs implemented in Partial Differential Equation Toolbox are used as a mathematical model for a wide variety of phenomena in all branches of engineering and science. The following is by no means a complete list of examples.

The elliptic and parabolic equations are used for modelling:

- Steady and unsteady heat transfer in solids
- Flows in porous media and diffusion problems

- Electrostatics of dielectric and conductive media
- Potential flow

The hyperbolic equation is used for:

- Transient and harmonic wave propagation in acoustics and electromagnetics
- Transverse motions of membranes

The Eigen value problems are used for:

- Determining natural vibration states in membranes and structural mechanics problems Partial Differential Equation Toolbox is easy to use in the most common areas due to the application interfaces. Eight application interfaces are available, in addition to the generic scalar and system (vector valued u) cases:

- Structural Mechanics — Plane Stress
- Structural Mechanics — Plane Strain
- Electrostatics
- Magnetostatics
- AC Power Electromagnetics
- Conductive Media DC
- Heat Transfer
- Diffusion

These interfaces have dialog boxes where the PDE coefficients, boundary conditions, and solution are explained in terms of physical entities. The application interfaces enable you to enter specific parameters, such as Young's modulus in the structural mechanics problems. Also, visualization of the relevant physical variables is provided. More detail about Partial differential toolbox can be found from the Matlab pdf help book.

Chapter 3

Artificial Neural Network

3.1 Neural Networks

Research in the field of neural networks has been attracting increasing attention in recent years. Since 1943, when Warren McCulloch and Walter Pitts presented the first model of artificial neurons, new and more sophisticated proposals have been made from decade to decade. Mathematical analysis has solved some of the mysteries posed by the new models but has left many questions open for future investigations. Needless to say, the study of neurons, their interconnections, and their role as the brain's elementary building blocks is one of the most dynamic and important research fields in modern biology. It can be illustrate the relevance of this endeavour by pointing out that between 1901 and 1991 approximately ten percent of the Nobel Prizes for Physiology and Medicine were awarded to scientists who contributed to the understanding of the brain. A neural network is a computational and mathematical model of the brain. In neural network models computation is distributed over several simple units called neurons. Neurons are interconnected and operate parallely that's why neural networks are also called parallel-distributed-processing systems or connectionist systems. An artificial neural network i.e. ANN often called as neural network i.e. NN, is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing. All the computations are based on a connectionist approach. An ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. Practically the neural networks are non-linear statistical data modelling tools. They can be used to model complex relationships between inputs and outputs.

Neural networks i.e. NNs do not perform miracles. But if used sensibly they produce amazing results. Artificial Neural Networks i.e. ANNs are non linear information processing structures in which the elements called as neurons process the information. Signals are transmitted by means of connecting links. The links possess an associated weight, which is multiplied along with the incoming signal i.e. net input for any typical neural network. The output signal is obtained by applying activations to the net input. An Artificial Neural Network is characterized by

1. Its architecture i.e. connection between neurons
2. Its training or learning i.e. to determining weights on connections
3. Its activation functions

The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not make them comfortably to do too much in their inventions. But in late 1980's interest in NN increased with algorithms like Back Propagation and Kohonen. Progress continued during the 1990's also. And it leads to many fold applications in all areas. The neural network field enjoys a resurgence of interest from all. NNs are used in many commercial applications like character recognition, image recognition, fraud detection, credit evaluation, insurance, load forecasting and stock forecasting etc.

In modern software implementations of artificial neural networks the approach inspired by biology has for the most part been abandoned for a more practical approach based on statistics and signal processing. In some of these systems, neural networks or parts of neural networks (such as artificial neurons) are used as components in larger systems that combine both adaptive and non-adaptive elements. While the more general approach of such adaptive systems is more suitable for real-world problem solving. It has less to do with the traditional artificial intelligence connectionist models. But they have in common the principle of non-linear, distributed, parallel and local processing and adaptation.

The most popular neural network is the multi-layer perceptron, which is a feed forward network. For artificial neural network to give any results it must be trained with series of examples and conditions. During the training neural network learns the governing relationships in given data sets. Input vectors to produce right solutions i.e. output vectors. For this purpose, back-propagation training algorithm is used. It is an iterative algorithm for minimizing the mean square error between predicted and desired output values. All signals flow in a single direction from the input to the output of the network. Feed forward networks can perform static mapping between an input space and an output space: the output at a given instant is a function only of the input at that instant.

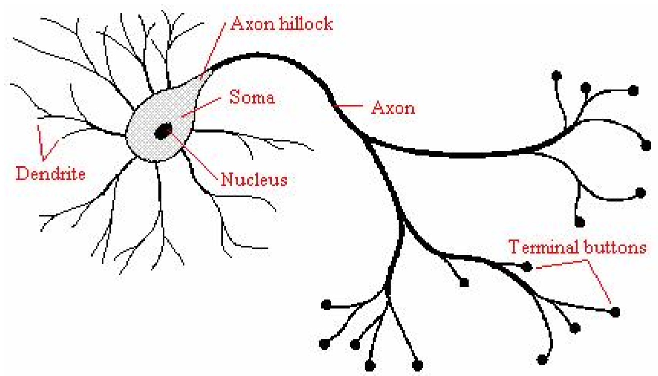


Figure.3.1 Simple Biological Neuron

3.1.2 Types of neural network

Traditionally, the term neural network has been used to refer to a network of biological neurons. In modern usage, the term is often used to refer to artificial neural networks, which are composed of artificial neurons or nodes. Thus the term 'Neural Network' has two distinct connotations:

1. Biological neural networks are made up of real biological neurons that are connected or functionally-related in the peripheral nervous system or the central nervous system. In the field of neuroscience, they are often identified as groups of neurons that perform a specific physiological function in laboratory analysis.

2. Artificial neural networks are made up of interconnecting artificial neurons (usually simplified neurons) designed to model (or mimic) some properties of biological neural networks. Artificial neural networks can be used to model the modes of operation of biological neural networks, whereas

cognitive models are theoretical models that mimic cognitive brain functions without necessarily using neural networks while artificial intelligence are well-crafted algorithms that solve specific intelligent problems (such as chess playing, pattern recognition, etc.) without using neural network as the computational architecture [44,45].

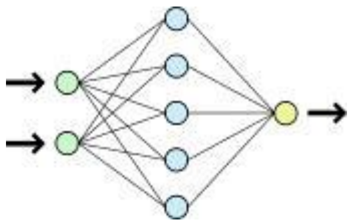


Figure 3.2: Simplified view of an artificial neural network

An (artificial) neural network consists of units, connections, and weights. Inputs and outputs are numeric.

Biological NN	Artificial NN
Soma	Unit
Axon	connection
Dendrite	weight
synapse	weighted sum
potential	bias
threshold	weight
signal	activation

Table 3.1 Biological and artificial neural network

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. This is true of ANNs as well [46].

3.2 Models of neurons

A neuron is a special nervous cell in organisms, which have electric activity. These cells are mainly intended for the operation of the organism. The biological neuron is shown schematically in Figure 3.1. A neuron consists of a cell body, which is surrounded by a membrane. The neuron has dendrites and axons, which are its inputs and outputs of neuron. Axons of neurons join to dendrites of other neurons by forming synaptic contacts (synapses). Input signals of the dendrite tree are weighted and added in the cell body and formed in the axon, where the output signal is generated. The signal's intensity, consequently, is a function of a weighted sum of the input signal. The output is passed through the branches of the axon and reaches the synapses. Through the synapses the signal is transformed into a new input signal for neighbour neurons. The input signal can be either positive or negative (exciting or inhibiting), depending on the synapses (Aliev 2001).In accordance with the biological model, different mathematical models were suggested. The mathematical model of the neuron, which is usually utilized under the simulation of NN, can be shown in Figure 3.3. The neuron receives a set of input signals x_1, x_2… x_n (vector X) which are usually the output signals of other neurons. Each input signal is multiplied to a corresponding connection weight, w, and analogue of the synapse's efficiency. Weighted input signals come to the summation module corresponding to cell body, where their algebraic summation is executed and the excitement level of neuron is determined:

$$n = \sum_{i=1}^{R} W_i X_i \qquad (3.1)$$

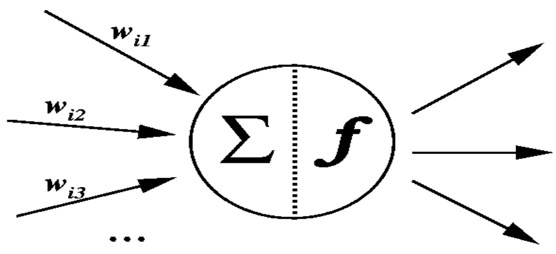


Figure 3.3 Mathematical neuron

The output signal of a neuron is determined by conducting the excitement level through the function f, called activation function as in Equation 3.2.

$$y = f(n) \qquad (3.2)$$

3.3 Basic neuron structure

The structure of biological neuron and artificial neuron is briefly explained here.

3.3.1 Biological neuron

A biological neuron or a nerve cell consists of synapses, dendrites, the cell body and the axon the various building blocks are

- The synapses are elementary signal processing devices
- A synapse is a biochemical device, which converts a pre-synaptic electrical signal into a chemical signal and then back into a post-synaptic electrical signal.
- The input pulse train has its amplitude modified by parameters stored in the synapse. The nature of this modification depends on the type of the synapse, which can be either inhibitory or excitatory.
- The postsynaptic signals are aggregated and transferred along the dendrites to the nerve cell body.
- The cell body generates the output neuronal signal, a spike, which is transferred along the axon to the synaptic terminals of other neurons.

3.3.2 Mathematical model

In accordance with the biological model, different mathematical models were suggested. The mathematical model of the neuron, which is usually utilized under the simulation of NN. The neuron receives a set of input signals x_1, x_2… x_n (vector X) which are usually the output signals of other neurons. Each input signal is multiplied to a corresponding connection weight, w, and analogue of the synapse's efficiency. Weighted input signals come to the summation module corresponding to cell body, where their algebraic summation is executed and the excitement level of neuron is determined:

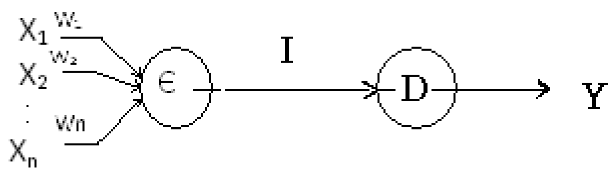


Figure 3.4 Mathematical neuron

The output signal of a neuron is determined by conducting the excitement level through the function f, called activation function as in Equation 2.1

$$y=f\ (I) \qquad (3.3)$$

3.4 Transfer function

The behaviour of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units.

3.4.1 Types of transfer functions

The transfer function of a neuron is chosen to have a number of properties which either enhance or simplify the network containing the neuron. Crucially, for instance, any multilayer perceptron using a linear transfer function has an equivalent single-layer network; a non-linear function is therefore necessary to gain the advantages of a multi-layer network.

Below, *u* refers in all cases to the weighted sum of all the inputs to the neuron, i.e. for n inputs,

$$u = \sum_{i=1}^{n} w_i x_i \qquad (3.4)$$

where w is a vector of synaptic weights and x is a vector of inputs.

3.4.1.1 Step function

The output y of this transfer function is binary, depending on whether the input meets a specified threshold, θ. The "signal" is sent, i.e. the output is set to one, if the activation meets the threshold.

$$y=\begin{cases}1, if \;\; u \geq \theta \\ 0 \; if, \;\; u < \theta\end{cases}$$

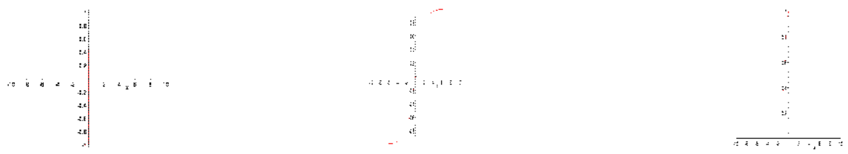
This function is used in perceptrons and often shows up in many other models. It performs a division of the space of inputs by a hyperplane. It is especially useful in the last layer of a network intended to perform binary classification of the inputs. It can be approximated from other sigmoidal functions by assigning large values to the weights.

3.4.1.2 Linear combination

In this case, the output unit is simply the weighted sum of its inputs plus a bias term. A number of such linear neurons perform a linear transformation of the input vector. This is usually more useful in the first layers of a network. A number of analysis tools exist based on linear models, such as harmonic analysis, and they can all be used in neural networks with this linear neuron. The bias term allows us to make affine transformations to the data.

3.4.1.3 Sigmoid

A fairly simple non-linear function, the logistic function also has an easily calculated derivative, which can be important when calculating the weight updates in the network. It thus makes the network more easily manipulable mathematically, and was attractive to early computer scientists who needed to minimize the computational load of their simulations. It is commonly seen in multilayer perceptrons using a backpropagation algorithm



Step Function Symmetric Sigmoid Radial Basis Function

Figure 3.5 Transfer function

3.5 Type of neurons

Neuron can be classified in two types namely, simple neuron and complicated neuron. Following is the basic information of these two given in detail

3.5.1 Simple neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not. Figure 3.6 show an example of simple neuron.

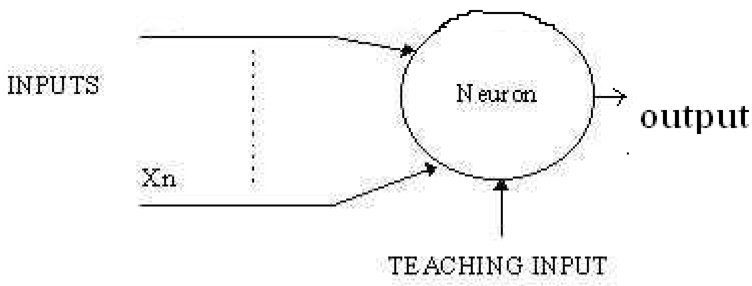


Figure.3.6 Simple neuron

3.5.2 Complicated neuron

The simple neuron doesn't do anything that conventional computers don't do already. The difference from the previous model is that the inputs are ‘weighted’; the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

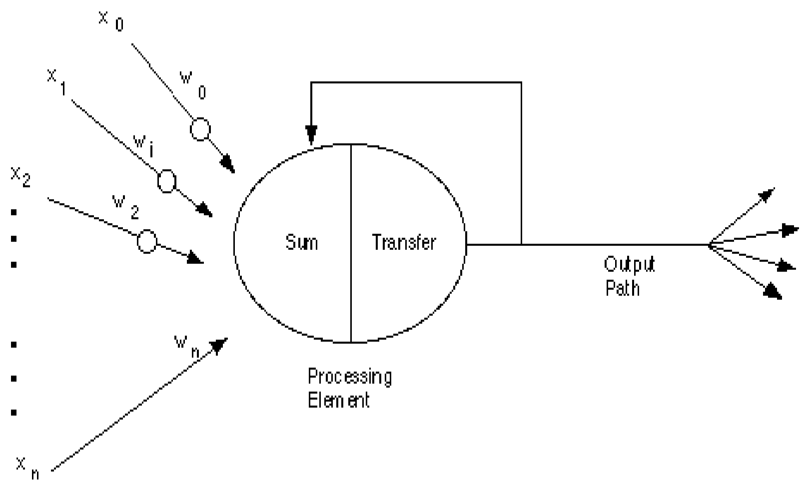


Figure.3.7 A complicated neuron

In mathematical terms, the neuron fires if and only if;

$$XW + XW + XW + ... > T$$

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. The complicated neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to 'adapt'; the most used ones are the Delta rule and the back error propagation.

3.6 Perceptron

The Perceptron, proposed by Rosenblatt in 1959, is somewhat more complex than a single layer network with threshold activation functions. In its simplest form it consist of an N-element input layer which feeds into a layer of M ‘association’, ‘mask’ or ‘predicate, units $Ø_h$ and a single output unit. Rosenblatt created many variations of the perceptron. One of the simplest was a single-layer network whose weights and biases could be trained to produce a correct target vector when presented with the corresponding input vector. The training technique used is called the perceptron learning rule. The perceptron generated great interest due to its ability to generalize from its training vectors and learn from initially randomly distributed connections. Perceptrons are especially suited for simple problems in pattern classification. They are fast and reliable networks for the problems they can solve.

3.6.1 Neuron model

A perceptron neuron, which uses the hard-limit transfer function hardlim, is shown below

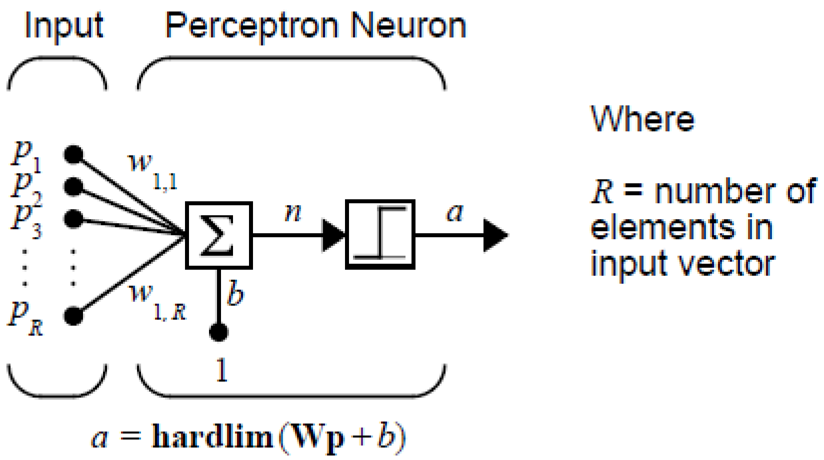
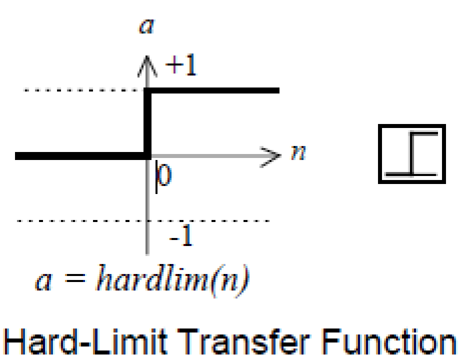


Figure 3.8 Perceptron neuron

Each external input is weighted with an appropriate weight w_{1j}, and the sum of the weighted inputs is sent to the hard-limit transfer function, which also has an input of 1 transmitted to it through the bias. The hard-limit transfer function, which returns a 0 or a 1, is shown below.



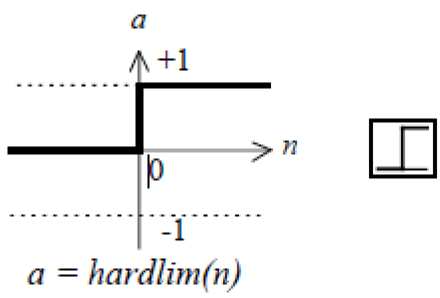


Figure 3.9 Transfer function

The perceptron neuron produces a 1 if the net input into the transfer function f is equal to or greater than 0; otherwise it produces a 0. The hard-limit transfer function gives a perceptron the ability to classify input vectors by dividing the input space into two regions. Specifically, outputs will be 0 if the net input n is less than 0, or 1 if the net input n is 0 or greater. The following figure show the input space of a two-input hard limit neuron with the weights $w_{1,1}$= -1, $w_{1,2}$=1 and a bias b=1.

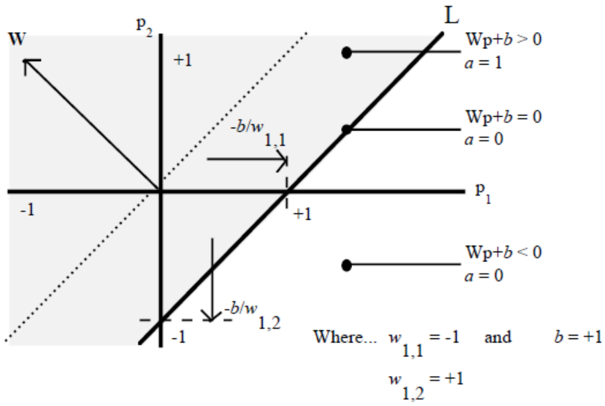


Figure 3.10 Two-input hard limit neuron

Two classification regions are formed by the decision boundary line L at W_p+ b=0. This line is perpendicular to the weight matrix W and shifted according to the bias b. Input vectors above and to the left of the line L will result in a net input greater than 0 and, therefore, cause the hard-limit neuron to output a 1. Input vectors below and to the right of the line L cause the neuron to output 0.

Hard-limit neurons without a bias will always have a classification line going through the origin. Adding a bias allows the neuron to solve problems where the two sets of input vectors are not located on different sides of the origin. The bias allows the decision boundary to be shifted away from the origin, as shown in the plot above.

3.6.2 Perceptron architecture

The perceptron network consists of a single layer of S perceptron neurons connected to R inputs through a set of weights $w_{i,j}$, as shown below in two forms. As before, the network indices i and j indicate that $w_{i,j}$ is the strength of the connection from the j^{th} input to the i^{th} neuron.

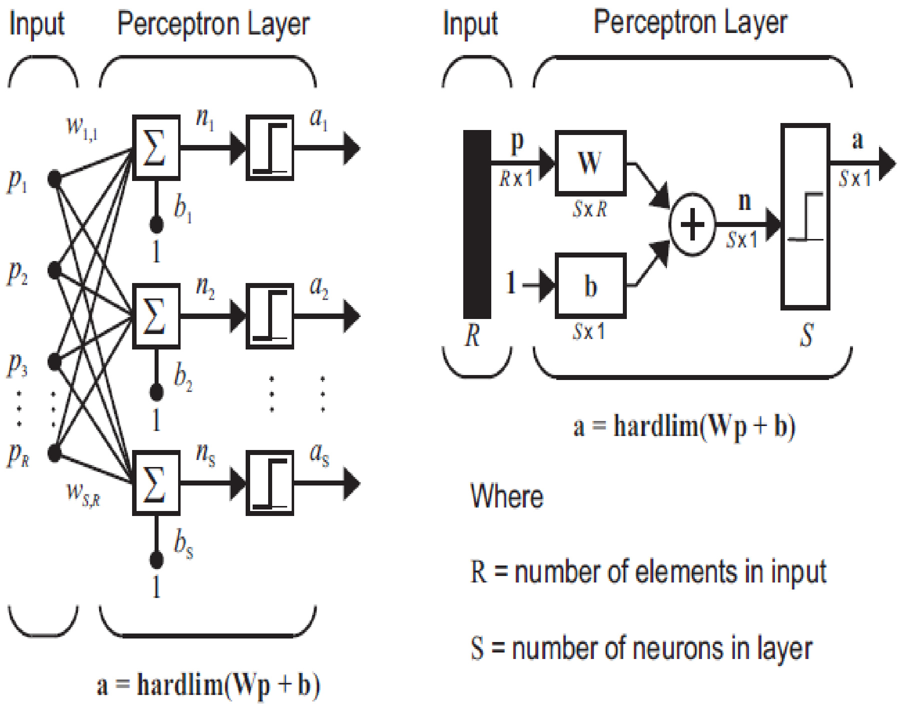


Figure 3.11 Perceptron network

The perceptron learning rule described shortly is capable of training only a single layer. Thus only one-layer networks are considered here.

3.7 Learning rules

A learning rule is defined as a procedure for modifying the weights and biases of a network. (This procedure can also be referred to as a training algorithm.) The learning rule is applied to train the network to perform some particular task. Learning rules in this toolbox fall into two broad categories: supervised learning, and unsupervised learning.

In supervised learning, the learning rule is provided with a set of examples (the training set) $\{p_1,t_1\},\{p_2,t_2\},\ldots,\{p_Q,t_Q\}$, of proper network behaviour where p_q an input to the network is, and is the corresponding correct (target) output. As the inputs are applied to the network, the network outputs are compared to the targets. The learning rule is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets. The perceptron learning rule falls in this supervised learning category. In unsupervised learning, the weights and biases are modified in response to network inputs only. There are no target outputs available. Most of these algorithms perform clustering operations. They categorize the input patterns into a finite number of classes. This is especially useful in such applications as vector quantization.

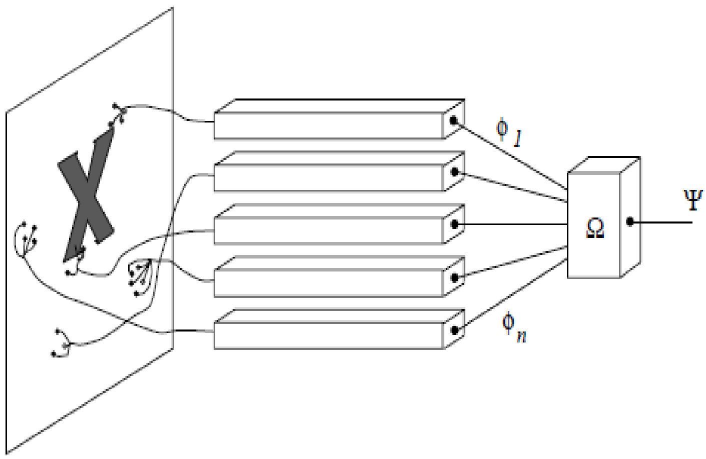


Figure 3.12 The perceptron

3.7.1 Adaline

An important generalisation of the perceptron training algorithm was presented by Widrow and Hoff as the 'least mean square' (LMS) learning procedure, also known as the delta rule. The main functional difference with the perceptron training rule is the way the output of the system is used in the learning rule. The perceptron learning rule uses the output of the threshold function (either -1 or +1) for learning. The delta-rule uses the net output without further mapping into output values-1 or +1.

The learning rule was applied to the 'adaptive linear element' also named Adaline[2] developed by Widrow and Hoff. In a simple physical implementation this device consists of a set of controllable resistors connected to a circuit which can sum up currents caused by the input voltage signals. Usually the central block, the summer, is also followed by a quantiser which outputs either +1 or -1 depending on the polarity of the sum.

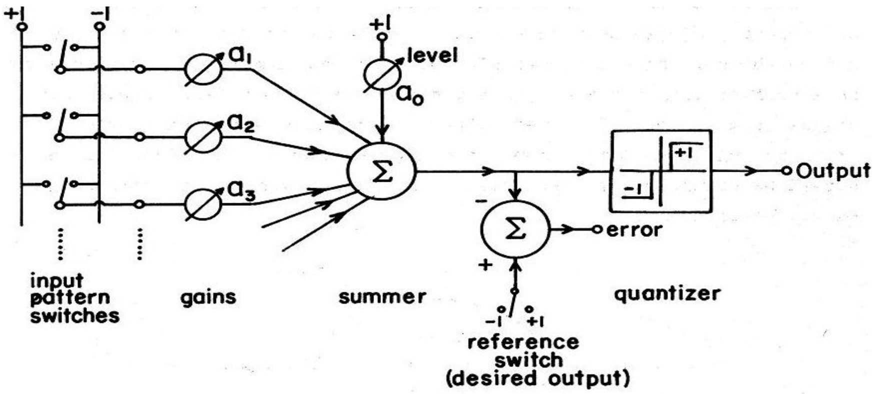


Figure 3.13 The adaline

Although the adaptive process is here exemplified in a case when there is only one output, it may be clear that a system with many parallel outputs is directly implementable by multiple units of the above kind. If the input conductance are denoted by w_i, i= 0, 1, 2… n and the input and output signals by x_i and y, respectively, then the output of the central block is defined to be

$$y = \sum_{i=1}^{n} w_i x_i + \theta_{ii} \qquad (3.5)$$

Output vs. activation of a unit: Since there is no need to do otherwise, we consider the output and the activation value of a unit to be one and the same thing. That is, the output of each neuron equals its activation value. Bias, offset, threshold: These terms all refer to a constant (i.e. independent of the network input but adapted by the learning rule) term which is input to a unit. They may be used interchangeably, although the latter two terms are often envisaged as a property of the activation function. Furthermore, this external input is usually implemented (and can be written) as a weight from a unit with activation value 1.

Number of layers: In a feed-forward network, the inputs perform no computation and their layer is therefore not counted. Thus a network with one input layer, one hidden layer and one output layer is referred to as a network with two layers.

Representation vs. learning: When using a neural network one has to distinguish two issues which influence the performance of the system. The first one is the representational power of the network; the second one is the learning algorithm.

The representational power of a neural network refers to the ability of a neural network to represent a desired function. Because a neural network is built from a set of standard functions, in most cases the network will only approximate the desired function and even for an optimal set of weights the approximation error is not zero.

The second issue is the learning algorithm. Given that there exist a set of optimal weights in the network, is there a procedure to (iteratively) find this set of weights?

The artificial neural network is made up of seven major components,

These components are summarized as:-

- Weighting Factors
- Summation Function
- Transfer Function / Activation Function
- Scaling and Limiting
- Output Function
- Error Function and Back-propagated Value
- Learning Function

Weighting factors: A neuron usually receives many simultaneous inputs. Each input has its own relative weight which gives the input impact that it needs on the processing element's summation function. These weights perform the same type of function as do the varying synaptic strengths of biological neurons. In both cases, some inputs are made more important than others so that they have a greater effect on the processing element as they combine to produce a neural response. These strengths can be modified in response to various training sets and according to a network's specific topology or through its learning rules.

Scaling and Limiting: After the processing element's transfer function, the result can pass through additional processes which scale and limit. This scaling simply multiplies a scale factor times the transfer value, and then adds an offset. Limiting is the mechanism which insures that the scaled result does not exceed an upper or lower bound. This limiting is in addition to the hard limits that the original transfer function may have performed.

Output Function (Competition): Each processing element is allowed one output signal which it may output to hundreds of other neurons. This is just like the biological neuron, where there are many inputs and only one output action. Normally, the output is directly equivalent to the transfer function's result. Some network topologies, however, modify the transfer result to incorporate competition among neighbouring processing elements. Neurons are allowed to compete with each other, inhibiting processing elements unless they have great strength. Competition can occur at one or both of two levels. First, competition determines which artificial neuron will be active, or provides an output. Second, competitive inputs help determine which processing element will participate in the learning or adaptation process.

Error Function and Back-Propagated Value: In most learning networks the difference between the current output and the desired output is calculated. This raw error is then transformed by the error function to match particular network architecture. The most basic architectures use this error directly, but some square the error while retaining its sign, some cube the error, and other paradigms modify the raw error to fit their specific purposes. The artificial neuron's error is then typically propagated into the learning function of another processing element. This error term is sometimes called the current error.

The current error is typically propagated backwards to a previous layer. Yet, this back-propagated value can be either the current error, the current error scaled in some manner (often by the derivative of the transfer function), or some other desired output depending on the network type. Normally, this back-propagated value, after being scaled by the learning function, is multiplied against each of the incoming connection weights to modify them before the next learning cycle.

3.8 Architectures of neural networks

The totality of the neurons, connected with each other and with the environment, forms the NN. Figure 3.14 shows the basic structure of the neural network. The input vector comes to the network by activating the input neurons. A set of input signals of a network's neurons is called the vector of input activeness. Connection weights of neurons are represented in form of matrix W, element w_{ij} of which is the connection weight between i-th and j-th neurons. During the network functioning process, the input vector is transformed into output one, i.e. some information processing is performed. The computational power of the network, thus, solves problems with its connections. Connections link inputs of one neuron with output of others. The connection strengths are given by weight coefficients. NN can also consist a bias term, which acts on a neuron like an offset. The function of the bias is to provide a threshold for the activation of neurons. The bias can be connected all neurons in network.

NNs can be divided into two types of architectures

- Feed forward Networks

- Recurrent NNs.

3.8.1 Feedforward networks

Feed-forward networks have no feedback connections. In this type of network, neurons of the j^{th} layer receive signals from environment (when j=1) or the neurons of previous the $(j-1)^{th}$ layer when (j>1) and pass their outputs to neurons of the next $(j+1)^{th}$ layer to the environment (when j is the last layer).

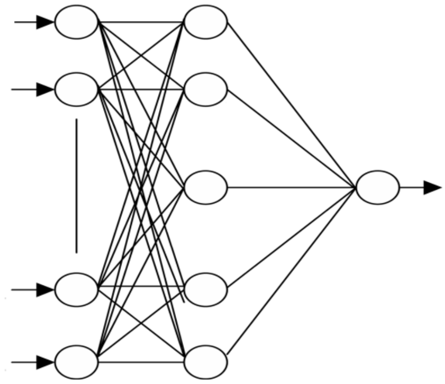


Figure 3.14 Basic structure of neural network

Feedforward networks can be single-layer or multi-layer. Multilayer NNs consist of input, output and hidden layer. The use of hidden layers allows an increase in the computational power of the network. Choosing the optimal structure of a network provides an increase in reliability and computational power, and a decreased processing.

3.8.2 Recurrent neural networks

Recurrent neural networks have structures similar to standard feedforward NN with layers of nodes connected via weighted feed-forward connections, but also include time delayed feedback or recurrent connections in the architecture. The important advantage of the RNN is the ability to approximate a continuous or discrete nonlinear dynamic system by neural dynamics defined by a system of nonlinear differential equations. This offers the opportunities for applications to adaptive control problems.

3.9 Training of ANN

A neural network has to be configured such that the application of a set of inputs produces (either 'direct' or via a relaxation process) the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to 'train' the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule. We can categories the learning situations in three distinct sorts. These are:

3.9.1 Supervised learning

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached. With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, learning can take weeks. This training is considered complete when the neural network reaches a user defined performance level. This level signifies that the network has achieved the desired statistical accuracy as it produces the required outputs for a given sequence of inputs. When no further learning is necessary, the weights are typically frozen for the application. After a supervised network performs well on the training data, then it is important to see what it can do with data it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network has not simply memorized a given set of data but has learned the general patterns involved within an application.

3.9.2 Unsupervised learning or Self organisation

In unsupervised learning we are given some data x and the cost function to be minimized, that can be any function of the data x and the network's output, f. The cost function is dependent on the task and our a priori assumptions (the implicit properties of our model, its parameters and the observed variables). As a trivial example, consider the model $f(x) = a$, where a is a constant and the cost $C = E\,[(x - f(x))^2]$. Minimizing this cost will give us a value of a is equal to the mean of the data. The cost function can be much more complicated. Its form depends on the application: For example in compression it could be related to the mutual information between x and y. In statistical modelling, it could be related to the posterior probability of the model given the data. Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering, the estimation of statistical distributions, compression and filtering.

3.9.3 Reinforcement learning

In reinforcement learning, data x is usually not given, but generated by an agent's interactions with the environment. At each point in time t, the agent performs an action y_t and the environment generates an observation x_t and an instantaneous cost c_t, according to some dynamics. The aim is to discover a *policy* for selecting actions that minimizes some measure of a long-term cost, i.e. the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated. More formally, the environment is modelled as a Markov decision process (MDP) with states $s_1,...,s_n \epsilon S$ and actions $a_1, \dots, a_m \epsilon A$ with the following probability distributions: the instantaneous cost distribution $P(c_t \mid s_t)$, the observation distribution $P(x_t \mid s_t)$ and the transition $P(s_{t+1} \mid s_t,a_t)$, while a policy is defined as conditional distribution over actions given the observations. Taken together, the two define a Markov chain (MC). The aim is to discover the policy that minimizes the cost, i.e., the MC for which the cost is minimal. ANNs are frequently used in reinforcement learning as part of the overall algorithm. Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.

3.10 Learning rates

The rate at which ANNs learn depends upon several controllable factors. In selecting the approach there are many trade-offs to consider. Obviously, a slower rate means a lot more time is spent in accomplishing the off-line learning to produce an adequately trained system. With the faster learning rates, however, the network may not be able to make the fine discriminations possible with a system that learns more slowly.

Network complexity, size, paradigm selection, architecture, type of learning rule or rules employed and desired accuracy must all be considered before training. These factors play a significant role in determining how long it will take to train a network. Changing any one of these factors may either extend the training time to an unreasonable length or even result in an unacceptable accuracy. Most learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one. If the learning rate is greater than one, it is easy for the learning algorithm to overshoot in correcting the weights, and the network will oscillate. Small values of the learning rate will not correct the current error as quickly, but if small steps are taken in correcting errors, there is a good chance of arriving at the best minimum convergence.

3.11 Learning laws

Many learning laws are in common use. Most of these laws are some sort of variation of the best known and oldest learning law, Hebb's Rule. Learning is certainly more complex than the simplifications represented by the learning laws currently developed. A few of the major laws are presented as examples.

3.11.1 Hebb's rule

The first, and undoubtedly the best known, learning rule were introduced by Donald Hebb. The basic Hebb’s rule is: If a neuron receives an input from another neuron and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

3.11.2 The delta rule

This rule is a further variation of Hebb's Rule. It is one of the most commonly used. This rule is based on the simple idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a processing element. This rule changes the synaptic weights in the way that minimizes the mean squared error of the network. This rule is also referred to as the Widrow-Hoff Learning Rule and the Least Mean Square (LMS) Learning Rule. The way that the Delta Rule works is that the delta error in the output layer is transformed by the derivative of the transfer function and is then used in the previous neural layer to adjust input connection weights. In other words, this error is back-propagated into previous layers one layer at a time. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feed-forward; Back-propagation derives its name from this method of computing the error term.

When using the delta rule, it is important to ensure that the input data set is well randomized. Well ordered or structured presentation of the training set can lead to a network which cannot converge to the desired accuracy. If that happens, then the network is incapable of learning the problem.

3.11.3 The gradient descent rule

This rule is similar to the Delta Rule in that the derivative of the transfer function is still used to modify the delta error before it is applied to the connection weights. Here, however, an additional proportional constant tied to the learning rate is appended to the final modifying factor acting upon the weight. This rule is commonly used, even though it converges to a point of stability very slowly. It has been shown that different learning rates for different layers of a network help the learning process converge faster. In these tests, the learning rates for those layers close to the output were set lower than those layers near the input. This is especially important for applications where the input data is not derived from a strong underlying model.

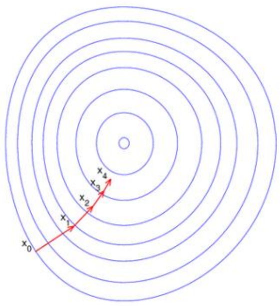


Figure 3.15 Illustration of gradient descent

3.11.4 Kohonen's learning law

This procedure, developed by Teuvo Kohonen, was inspired by learning in biological systems. In this procedure, the processing elements compete for the opportunity to learn, or update their weights. The processing element with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbors. Only the winner is permitted an output, and only the winner plus its neighbors are allowed to adjust their connection weights.

3.12 Types of Neural Networks

3.12.1 Feedforward Neural Network

The feedforward neural network was the first and arguably simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

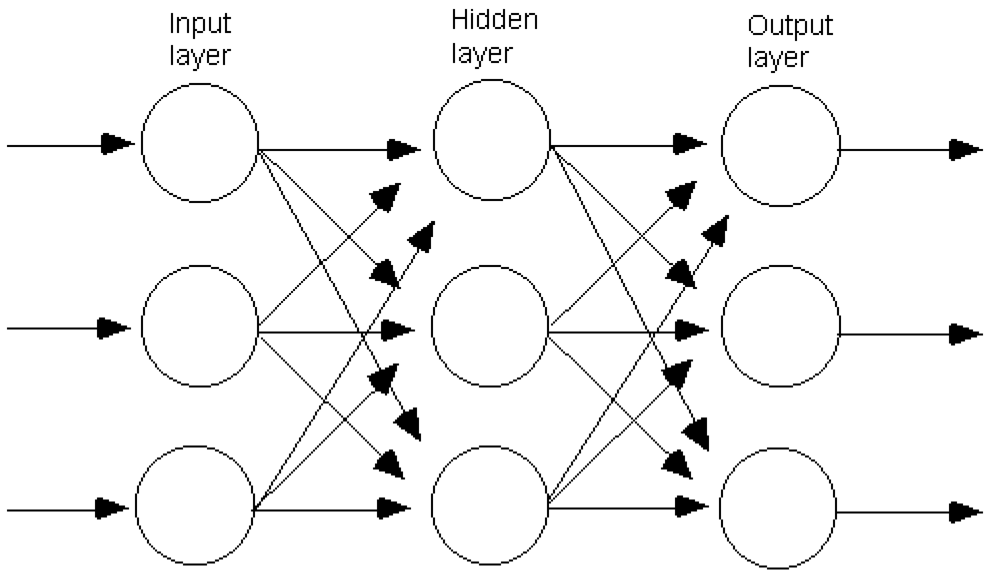


Figure 3.16 Feed forward network

3.12.2 Adaptive network

An adaptive network is an example of multilayer feed-forward NN in which each node performs a particular function (node function) on incoming signals as well as a set of parameters pertaining to this node. The formulas for the node functions may vary from node to node, node function depends on the overall input-output function, which the adaptive network is required to carry out. The links in an adaptive network only show the flow direction signals between nodes. No weights are associated with the links. The basic learning rule of adaptive is the backpropagation learning rule.

However, since it is slow and tends to become trapped in local minima, a hybrid learning rule algorithm was proposed to speed up the learning algorithm.

3.12.3 Radial basis function (RBF) network

Radial Basis Functions are powerful techniques for interpolation in multidimensional space. A RBF is a function which has built into a distance criterion with respect to a centre. Radial basis functions have been applied in the area of neural networks where they may be used as a replacement for the sigmoidal hidden layer transfer characteristic in Multi-Layer Perceptrons. RBF networks have two layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. The RBF chosen is usually a Gaussian. In regression problems the output layer is then a linear combination of hidden layer values representing mean predicted output. The interpretation of this output layer value is the same as a regression model in statistics. In classification problems the output layer is typically a sigmoid function of a linear combination of hidden layer values, representing a posterior probability. Performance in both cases is often improved by shrinkage techniques, known as ridge regression in classical statistics and known to correspond to a prior belief in small parameter values (and therefore smooth output functions) in a Bayesian framework.

RBF networks have the advantage of not suffering from local minima in the same way as Multi-Layer Perceptrons. This is because the only parameters that are adjusted in the learning process are the linear mapping from hidden layer to output layer. Linearity ensures that the error surface is quadratic and therefore has a single easily found minimum. In regression problems this can be found in one matrix operation. In classification problems the fixed non-linearity introduced by the sigmoid output function is most efficiently dealt with using iteratively re-weighted least squares.

RBF networks have the disadvantage of requiring good coverage of the input space by radial basis functions. RBF centres are determined with reference to the distribution of the input data, but without reference to the prediction task. As a result, representational resources may be wasted on areas of the input space that are irrelevant to the learning task. A common solution is to associate each data point with its own centre, although this can make the linear system to be solved in the final layer rather large, and requires shrinkage techniques to avoid overfitting.

Associating each input datum with an RBF leads naturally to kernel methods such as Support Vector Machines and Gaussian Processes (the RBF is the kernel function). All three approaches use a non-linear kernel function to project the input data into a space where the learning problem can be solved using a linear model. Like Gaussian Processes, and unlike SVMs, RBF networks are typically trained in a Maximum Likelihood framework by maximizing the probability (minimizing the error) of the data under the model. SVMs take a different approach to avoiding over fitting by maximizing instead a margin. RBF networks are outperformed in most classification applications by SVMs. In regression applications they can be competitive when the dimensionality of the input space is relatively small.

3.12.4 Kohonen self-organizing network

The self-organizing map (SOM) invented by Teuvo Kohonen performs a form of unsupervised learning. A set of artificial neurons learn to map points in an input space to coordinates in an output space. The input space can have different dimensions and topology from the output space and the SOM will attempt to preserve these.

3.12.5 Recurrent network

Contrary to feed-forward networks, recurrent neural networks (RNs) are models with bi-directional data flow. While a feed-forward network propagates data linearly from input to output, RNs also propagate data from later processing stages to earlier stages.

A recurrent neural network (RNN) is a class of neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behaviour. Recurrent neural networks must be approached differently from feed forward neural networks, both when analyzing their behaviour and training them. Recurrent neural networks can also behave chaotically. Usually, dynamical systems theory is used to model and analyze them. While a feed forward network propagates data linearly from input to output, recurrent networks (RN) also propagate data from later processing stages to earlier stages.

3.13 Advantages / Disadvantages of Artificial Neural Networks

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an expert in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer ‘what if’ questions.

1. *Adaptive learning*: An ability to learn how to do tasks based on the data given for training or initial experience.
2. *Self-Organization*: An ANN can create its own organization or representation of the information it receives during learning time.
3. *Real Time Operation*: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. *Fault Tolerance via Redundant Information Coding*: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

There are also some disadvantages of Neural Networks. These include:

1. The neural network needs training to operate.
2. The architecture of a neural network is different from the architecture of microprocessors therefore needs to be emulated.
3. Requires high processing time for large neural networks.

Applications of neural networks in different fields can be summarized as below

Industry	Business Applications
Aerospace	High-performance aircraft autopilot, flight path simulation, aircraft control systems, autopilot enhancements, aircraft component simulation, and aircraft component fault detection
Automotive	Automobile automatic guidance system, and warranty activity analysis
Banking	Check and other document reading and credit application evaluation
Defence	Weapon steering, target tracking, object discrimination, facial recognition, new kinds of sensors, sonar, radar and image signal processing including data compression, feature extraction and noise suppression, and signal/image identification
Electronics	Code sequence prediction, integrated circuit chip layout, process control, chip failure analysis, machine vision, voice synthesis, and nonlinear modelling
Entertainment	Animation, special effects, and

	market forecasting
Financial	Real estate appraisal, loan advising, mortgage screening, corporate bond rating, credit-line use analysis, credit card activity tracking, portfolio trading program, corporate financial analysis, and currency price prediction
Industrial	Prediction of industrial processes, such as the output gases of furnaces, replacing complex and costly equipment used for this purpose in the past
Insurance	Policy application evaluation and product optimization
Manufacturing	Manufacturing process control, product design and analysis, process and machine diagnosis, real-time particle identification, visual quality inspection systems, beer testing, welding quality analysis, paper quality prediction, computer-chip quality analysis, analysis of grinding operations, chemical product design analysis, machine maintenance analysis, project bidding, planning and

	management, and dynamic modelling of chemical process system
Medical	Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times, hospital expense reduction, hospital quality improvement, and emergency-room test advisement
Oil and gas	Exploration
Robotics	Trajectory control, forklift robot, manipulator controllers, and vision systems
Speech	Speech recognition, speech compression, vowel classification, and text-to-speech synthesis
Securities	Market analysis, automatic bond rating, and stock trading advisory systems
Telecommunications	Image and data compression, automated information service, real-time translation of spoken language and customer payment processing systems
Transportation	Truck brake diagnosis systems, vehicle scheduling,

	and routing systems

Table 3.3 Applications of neural networks in different fields

3.14 Neural Network Toolbox

The Neural Network Toolbox extends the Matlab computing environment to provide tools for the design, implementation, visualization, and simulation of neural networks. Neural networks are very powerful tools that are used in applications where formal analysis would be difficult or impossible, such as pattern recognition and non-linear system identification and control. The Neural Network Toolbox provides a comprehensive support for many proven network paradigms, as well as a graphical user interface that enables the experiment to design and manage the networks. The toolbox's modular, open, and extensible design simplifies the creation of customized functions and networks. More can be read from Matlab pdf help on neural network. The main features of Neural Network toolbox are as follows (Demuth H. And Beale M., 2001):

- Support for the most commonly used supervised and unsupervised network architectures
- A comprehensive set of training and learning functions
- Modular network representation, allowing an unlimited number of input setting layers, and network interconnections
- Pre- and post-processing functions for improving network training and assessing network performance

Chapter 4

Problem Formulation & Proposed Solution

4.1 Problem Formulation

There are many industrial applications where sheets of metal need to be checked for the cracks and other defects. Amongst the available techniques that employ X-rays and ultrasonic, there is a method of detecting these defects using the electromagnetic properties of the sheet. For this sensors that measure electromagnetic induction in the sheet are to be placed at several points on the sheet. Computerised analysis of the output of these sensors is used to predict the type and extent of defect. In case we employ ANN for identification of the defects, a major hurdle that needs to be crossed is the rigorous training of ANN using several known defects. Since, creating the known defects of various types on sheet metal can be daunting task; an alternative route of simulating these defects through FEM may be employed. This problem of simulating the defects in sheet metal using FEM followed by training and validation of ANN for identification of these defects is tackled in this research work.

4.2 Purposed Solution

The purposed solution can simply be stated by a figure below as shown in figure 4.1.Inverse problem of defect identification has been addressed in this work. Inverse problem is highly nonlinear and without formulations to follow, it is very difficult to construct effective inversion algorithms. Inverse problems arise in a number of areas. Examples include industrial non-destructive testing, medical diagnostics, geophysical prospecting for petroleum and minerals, and detection of earthquakes. This work presents an approach which is based on the use of artificial neural networks and finite element analysis to solve the inverse problem of defect identification. An artificial neural network, however, has the following properties: nonlinearity, input-output mapping, fault tolerance and most important, learning from examples. The need for learning from examples is closely related to the difficulty of formulating explicit rules. [44].

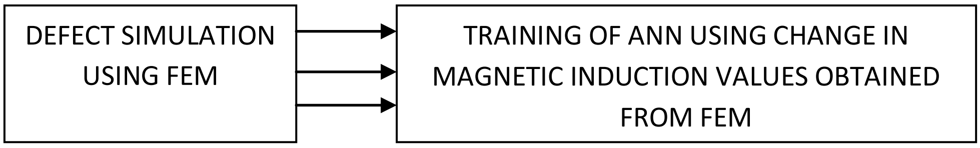


Figure 4.1 Basic methodology used for identification of defects

4.2.1 Simulation of Defect using FEM (PDEtool)

In part of defect simulation using FEM, as it is an electromagnetic inverse problem which can be stated simply as the following: if there is an electromagnetic device, it is easy to calculate the magnetic induction in any region of the device. Some values of this magnetic induction to predict defects in a region of the electromagnetic device are taken. In simulation different defects are created on metallic wall using PDEToolbox, the solution of the obtained analysis of which a vector consists of 11 input values which represent the deviation of magnetic induction and two output values which represent the height and width of defects. The simulations were done for a hypothetic metallic wall with 1mm height and 15mm width. The material of the wall is a magnetic material. The relative permeability of the core is supposed to be 600 and the air gap is 0.2mm. Finite element meshes with 35000 elements and 16000 nodes, approximately, were used in the simulations. During the phase of finite elements simulations, errors can appear, due to it's massively nature. So, the results of the simulations must be carefully analyzed. Some of the values of magnetic inductions that are used for the identification of the defects are given below; the values obtained will be after FEM analysis is complex but a senor can detect only a real signal therefore an absolute value of obtained result is taken in our case as said a vector consists of 11 input values which represent the deviation of magnetic induction in Wb/m^2 is taken which are given below.

Thus the first problem of defect simulation has been solved using the Pdetool a toolbox available for FEM analysis in this a metallic wall is simulated and after simulation of this FEM analysis of this is done which consists of meshing and solution the meshing of sheet is shown in figure 4.2

and that of solution obtained for one of these generated defects is shown in figure 4.3

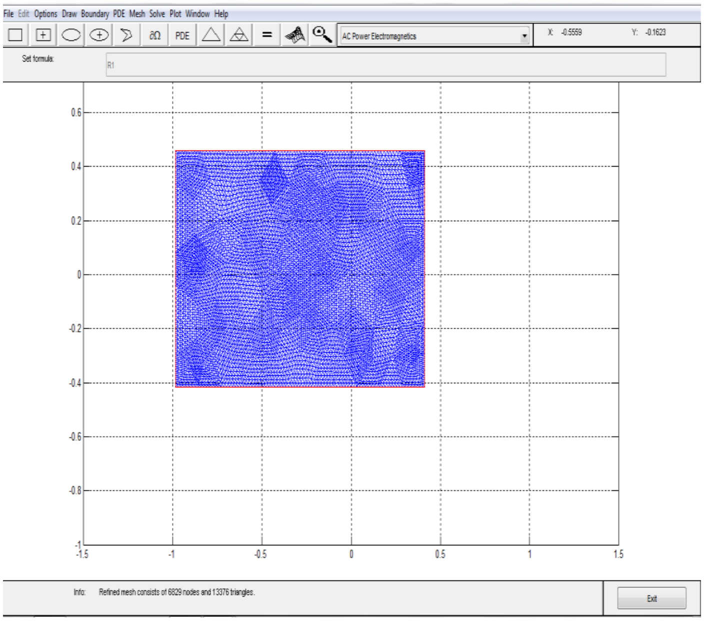


Figure 4.2 Meshing of sheet

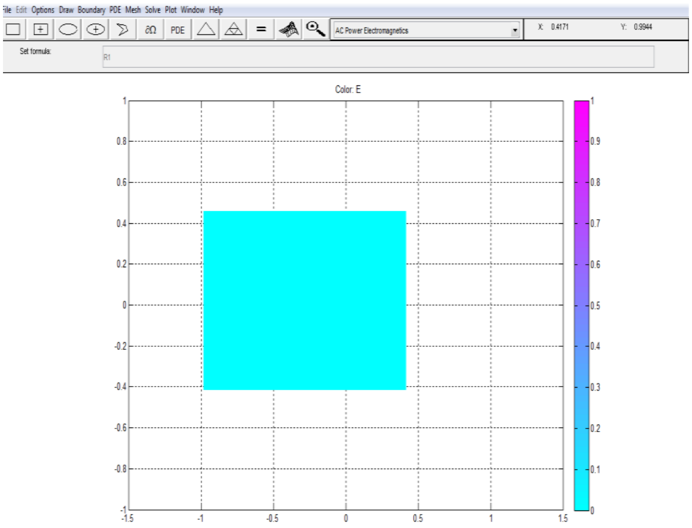


Figure 4.3 Solution of FEA on metallic sheet.

In the finite element analysis change in magnetic induction of the metallic sheet due to the presence of defect has been analysed and the obtained values are used for the training of the neural network, therefore the problem for the requirement of the training of the neural network has been well addressed by this work of simulation.

. The network weights can be embedded in an electronic device and used to identify defects in real pieces with similar characteristics to those of the simulated ones.

C_1	C_2	C_3	C_4	C_5	C_6	C_7
500.00	500.00	498.75	203.06	387.56	479.89	456.87
500.00	203.08	467.76	500.00	476.76	510.44	500.00
481.27	230.08	478.92	500.00	500.00	520.67	510.44
503.33	230.15	538.35	505.54	458.75	489.99	510.56
481.70	387.16	476.26	509.87	501.32	450.00	500.95
350.27	307.91	432.76	410.87	433.94	310.33	475.77
500.00	417.74	453.87	431.22	400.45	510.22	510.22
500.00	443.41	543.76	324.57	432.43	458.89	500.32
500.00	469.43	521.86	276.88	456.76	500.54	532.33
500.00	459.38	500.00	454.99	432.65	500.43	542.33

Table 4.1 Different values of magnetic induction

The different defects have been simulated using the PDEtool of Matlab; the procedure for simulation of defects contains some common steps as like pre-processing, meshing, solution and post processing. As defects are of different height and width will have different values of magnetic induction which are exported to Matlab workspace from where an input vector is formed for the training of the neural network.

4.2.2 Acquisition of data for Training of ANN

Now as simulation of defect is done in above now job left is to identify the defects in metallic wall but this requires training of ANN. The ANN is trained for the identification of defects as due to occurrence of specific defects change in magnetic induction value will take place these values are used for the training of neural network. The change in value of magnetic induction value that are generated using Pdetool are exported to Matlab workspace of which an absolute value are taken of which 11 values are taken as said above a vector is formed which is imported for training of neural network also the target data is given that is respective height and width.

As said the obtained results are used to generate a set of vectors for the training of two neural network models: multilayer perceptron neural network (MLP) and radial basis functions (RBF). Finally, the obtained neural networks are used to classify a group of new defects, simulated by the finite element method, but not belonging to the original dataset.

An electromagnetic device was idealized to be used as exciter for an electromagnetic field (figure 4.4).A direct current in the coil is considered, so the metallic wall material must be ferromagnetic. A high permeability magnetic core is used with a minimum air gap between the core and the metallic wall to increase the sensitivity of the electromagnetic device.

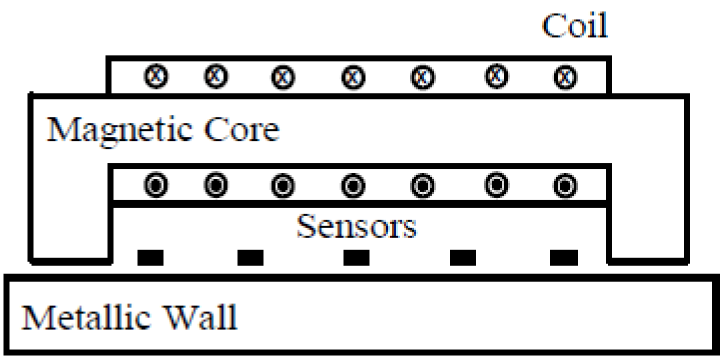


Figure 4.4 Arrangement for the measurements

In this work, using the FEM magnetic field is calculated in the two steps. This method is based on the A representation of the magnetic field [58]. First of all calculation of magnetic field intensity is done by solving the systems of equations

$$\text{rot}\,(H) = J \quad (4.1)$$

$$\text{div}\,(B) = 0 \quad (4.2)$$

H: Magnetic field

B: Magnetic induction

J: Electric current density

The above equation is coupled with relations associated to material property, isotropic material is assumed.

$$B = \mu\,(|H|)\,H \quad (4.3)$$

The magneto-static field analysis for a Cartesian electromagnetic system is carried out by the FEM. [44]

The electromagnetic field equation is expressed by the magnetic vector potential A as

$$\text{div}\left(\frac{1}{\mu}\,grad(A)\right) = -J \quad (4.4)$$

Following algebric matrix equation is obtained when equation (4.4) is discretised using the Galerkin FEM

[K][A]= [F]

with

$$A = \sum_j \alpha_j\,(x, y) A_j$$

$$K_{ij} = \iint_{\Omega} \frac{1}{\mu}\, grad\alpha_i \; grad\alpha_j \, dxdy$$

$$F_i = \iint_{\Omega} J \; \alpha_i dxdy$$

$$\alpha_i: Projection\, function$$

$$\alpha_j: interpolation\, function$$

Field solution is used to calculate the magnetic induction B in the second step. More details about the FEM can be found in [59].

4.3 Algorithm for defect identification

Step 1: Initial finite element mesh generation.

Step 2: Changing the defect shapes which led to the modifications in the finite element mesh.

Step 3: Magnetic inductions values at the sensor position is obtained from finite element solutions

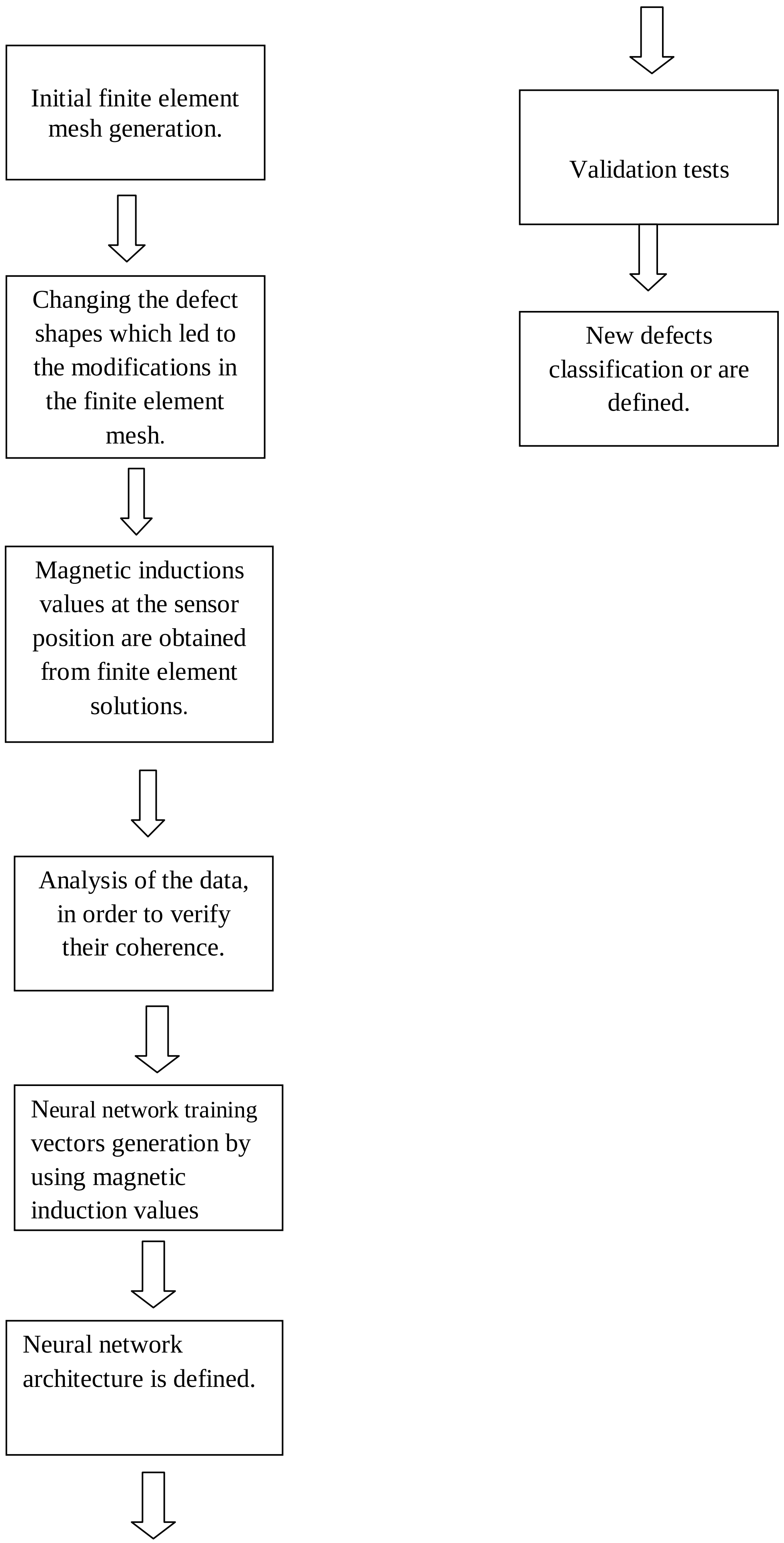
Step 4: Neural network training vectors generation by using magnetic induction values.

Step 5: Definition of the neural network architecture.

Step 6: Training of neural network.

Step 7: Validation tests and classification of new defects.

4.4 FLOW CHART FOR DEFECT IDENTIFICATION



Chapter 5

Results and Discussion

5.1 Results and discussion

The problem was solved under Matlab® workspace using the partial differential equation toolbox and neural network toolbox for the mesh generation and neural networks architecture definition. The network is trained by large number of defects in a metallic wall simulated using FEM. The obtained results are then used to generate the training vector for ANN. The trained network is used to identify new defects in the metallic wall, which not belong to the original dataset.

5.1.2 FORMULATIONOF NETWORK MODELS FOR PARAMETERS IDENTIFICATIONS

The training vectors for neural network have been generated. The RBF neural network architecture considered for this application was a single hidden layer with Gaussian RBF. For MLP network architecture, a single hidden layer with sigmoid activation function, which is optimal for the dichotomous outcome, is chosen. Table 1 show some results for the validation of the networks, for this session. As we can see, the results obtained in the validation are very close to the expected ones. The worse identification defect was obtained with MLP neural network.

Table 5.1 Expected and obtained values during a training session

Height (mm)			Width (mm)		
Expected	MLP	RBF	Expected	MLP	RBF
0.0710	0.0703	0.0719	5.9484	5.9474	5.9478
0.1988	0.1974	0.1983	6.4670	6.4554	6.4652
0.2418	0.2409	0.2433	1.5133	1.5188	1.5391
0.3378	0.3352	0.3365	4.5790	4.5708	4.5760
0.5169	0.5140	0.5152	3.3280	3.3351	3.3195
0.6840	0.6838	0.6843	2.1720	2.1897	2.1745

5.1.3 NEW IDENTIFICATIONS

After the neural networks training and respective validations, new defects were simulated by the finite element method, for posteriori classification by the networks. Table 2 shows the dimensions of the defects (height and width), and the obtained dimensions, by the neural networks.

Table 5.2 Simulation results, for new defects

Defect	Height (mm)			Width (mm)		
	Expected	Obtained		Expected	Obtained	
		MLP	RBF		MLP	RBF
1	0.0750	0.0758	0.0758	3.910	3.903	3.906
2	0.1250	0.1236	0.1244	2.462	2.480	2.480
3	0.2150	0.2118	0.2156	4.850	4.843	4.835
4	0.5750	0.5563	0.5758	1.080	1.090	1.084

Chapter 6

Conclusion and Future Scope

6.1 Conclusion and Future Scope

In this work an investigation is presented on the use of the finite element method and artificial neural networks for the identification of defects in metallic walls, present in industrial plants. For a given metallic wall characteristics, defects can be simulated by the finite element method, and the magnetic fields results are used in the preparation of the training vectors for artificial neural networks. The network can be embedded in electronic devices in order to identify defects in real metallic walls.

The association of FEM and ANN techniques seems to be a useful alternative for identification of defects trough inverse analysis. Future works are intended to be done in this field, such as the use of more realistic finite element problems, computer parallel programming, in order to get quickly solutions.

REFERENCES

[1] T. S. Low and B. Chao, "The use of finite elements and neural networks for the solution of inverse electromagnetic problems", *IEEE Trans. Magnetics*, Vol.28, No.5, 1992.

[2] F. C. Morabito and M. Campolo, "Location of plural defects in conductive plates via neural networks", *IEEE Trans. Magnetics*, Vol.31, No.3, pp. 1765-1768, 1995.

[3] P. Ramuhalli, L. Udpa and S.S. Udpa, "Electromagnetic NDE signal inversion by function approximation neural networks", *IEEE Trans. on Magnetics*, vol. 38, no. 6, pp. 3633-3642, 2002.

[4] S. Haykin, "Neural networks-a comprehensive course", USA, *Prentice Hall*, 1999.

[5] Cheng. A. and D. T. Cheng. "Heritage and early history of the boundary element method, Engineering Analysis with Boundary Elements", 29, 268–302, 2002.

[6] D. A. Anderson, J. C. Tannehill, and R. H. Pletcher. "Computational Fluid Mechanics and Heat Transfer", Hemisphere, Washington, DC, 1984.

[7] F. J. Rizzo, "An Integral Equation Approach to Boundary Value Problems of Classical Electrostatics", *Q. Appl. Math.*, Vol. 25, pp. 83–95, 1967.

[8] F. J. Rizzo and D. J. Shippy, "An Advanced Boundary Integral Equation Method for Three-Dimensional Thermoelasticity", *Int. J. Appl.* Mech., Vol. 11,pp.1753–1790, 1977.

[9] C. Brebbia, J.C.F. Telles, and L. C. Wrobel "Boundary Element Technique", *Springer*-Verlag, Berlin 1984.

[10] R. P. Banaugh and W. Goldsmith, "Diffraction of Steady Acoustic Waves by Surfaces of Arbitrary Shape", *J. Acoust. Soc. Am.*, Vol. 35, No. 10, pp. 1590, 1963.

[11] R. W. Clough, "The Finite Element Method in Plane Stress Analysis", Proceedings of 2nd ASCE *Conference on Electronic*

Computation, Pittsburgh, PA, September 8–9, 1960.

[12] R. Courant, “Variational Methods for the Solutions of Problems of Equilibrium and Vibrations”, *Bull. Am. Math. Soc.*, Vol. 49, pp. 1–23, 1943.

[13] J. Greenstadt, “On the Reduction of Continuous Problems to Discrete Form”, IBM *J.* Res. Dev., Vol. 3, pp. 355–363, 1959.

[14] P. M. Morse and H. Feshback, “Methods of Theoretical Physics”, *McGraw- Hill*, New York, Section 9.4, 1953.

[15] W. Prager and J. L. Synge, “Approximation in Elasticity Based on the Concept of Function Space”, *Q. Appl. Math.*, Vol. 5, pp. 241–269, 1947.

[16] J. L. Synge, “Triangulation in the Hypercircle Method for Plane Problems”, *Proc. R. Irish Acad.*, Vol. 54A21, 1952.

[17] A. Hrenikoff, “Solution of Problems in Elasticity by the Framework Method”, *J. Appl. Mech.*, Vol. 8, pp. 169–175, 1941.

[18] D. McHenry, “A Lattice Analogy for the Solution of Plane Stress Problems”, *J. Inst. Civ. Eng*., Vol. 21, pp. 59–82, 1943.

[19] N. M. Newmark, “Numerical Methods of Analysis in Engineering”, L. E. Grinter (ed.), *Macmillan*, New York, 1949.

[20] G. Kron, “Tensorial Analysis and Equivalent Circuits of Elastic Structures”, *J. Franklin Inst.*, Vol. 238, No. 6, pp. 400–442, December 1944.

[21] G. Kron, “Equivalent Circuits of the Elastic Field”, *J. Appl. Mech.*, Vol. 66, pp. A-149 to A-161, 1994.

[22] J. H. Argyris, “Energy Theorems and Structural Analysis”,Aircraft Eng., Vol. 26, pp. 347–356, 383–387, 394, October-November 1954.

[23] J. H. Argyris, “Energy Theorems and Structural Analysis”, *Aircraft Eng.*, Vol. 27, pp. 42–58, 80–94, 125–134, 145–158, , February-March-April-May 1955.

[24] J. H. Argyris, “The Matrix Theory of Statics” (in German), *Ingenieur Archiv*, Vol. 25, pp. 174–192, 1957.

[25] J. H. Argyris, “The Analysis of Fuselages of Arbitrary Cross-Section and Taper”, *Aircraft Eng.*, Vol. 31, pp. 62–74, 101–112, 133–143, 169–180, 192–203, 244–256, 272–283, 1959.

[26] J. H. Argyris and S. Kelsey, “Energy Theorems and Structural Analysis”, *Butterworth*, London, 1960.

[27] M. J. Turner, R. W. Clough, H. C. Martin, and L. C. Topp, “Stiffness and Deflection Analysis of Complex Structures”, *J. Aeronaut*. Sci., Vol. 23, No. 9, pp. 805–823, 854, 1956.

[28] R. W. Clough, “The Finite Element Method after Twenty-Five Years: A Personal View”, *Comput. Structures*, Vol. 12, pp. 361–370, 1980.

[29] O. C. Zienkiewicz and Y. K. Cheung, “Finite Elements in the Solution of Field Problems”, Engineer, Vol. 220, pp. 507–510, 1965.

[30] A. K. Noor, “Bibliography of Books and Monographs on Finite Element Technology”, *Appl. Mech*. Rev., Vol. 44, No. 8, pp. 307–317, June 1991.

[31] Taddei F., Martelli S., Reggiani B., Cristofolini L. and Viceconti M., “Finite-Element Modeling of Bones From CT Data: Sensitivity to Geometry and Material Uncertainties”, *IEEE Transactions on Biomedical Engineering*, 2194 – 2200, Nov. 2006.

[32] Teran J., Sifakis E., Blemker S.S and Ng-Thow-Hing V., Lau C and Fedkiw R. “Creating and simulating skeletal muscle from the visible human data set”, *IEEE Transactions on Visualization and Computer Graphics*, 317 – 328, May-June 2005.

[33] DiMaio S.P. and Salcudean, “S.E. Interactive simulation of needle insertion models”, *IEEE Transactions on Biomedical*

Engineering, 1167 – 1179, July 2005.

[34] G. Yong and C. Zhen. "A study of the zona piercing process in piezodriven intracytoplasmic sperm injection", *IEEE Journal of Applied Physics*, 044702 - 044702-8, Aug 2008.

[35] Samani A., Bishop J. and Plewes D.B. "A constrained modulus reconstruction technique for breast cancer assessment", *IEEE Transactions on Medical Imaging*, 877 – 885, Sept. 2001.

[36] Meng L. and Zhang S. "Simulation of microstructure defects creep on nickel-based superalloys by Genetic Algorithm", *IEEE International Conference on Intelligent Computing and Intelligent Systems*, 2009. , Page(s): 327 – 330. Volume: 1 2009.

[37] Dong Kil Shin and Jung Ju Lee. "Effective material properties and thermal stress analysis of epoxy molding compound in electronic packaging", *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, Part B: Advanced Packaging. Page(s): 413 – 421, 1998.

[38] Kargarnovin, M.H., Madani, M. and Savadkohi-Goran, D. "An investigation on the effect of stem size and type of prosthesis fixation on stress distribution of the femur using finite element method", *Engineering in Medicine and Biology Society*, 1995 and 14th Conference of the Biomedical Engineering Society of India, Page(s): 3/21 - 3/22, Publication Year: 1995.

[39] Li Jie, Zhang Lei and Zhao Qi. "Comparison and Analysis on Different Finite Element Models of Gear Interfacial Contact Temperature", *International Conference on Computer Modeling and Simulation*, Page(s): 132 – 136, 2010.

[40] Rao, V.S.; Xiaowu Zhang; Ho Soon Wee; Rajoo, R.; Premachandran, C.S.; Kripesh, V. Seung Wook Yoon and Lau, J.H.; "Design and Development of Fine Pitch Copper/Low-K Wafer Level Package", *IEEE Transactions on Advanced Packaging*, Page(s): 377 – 388, 2010.

[41] Ki-Chan Kim; Kab-Jae Lee and Ju Lee. "Analysis of electromagnetic force distribution on end winding for motor reliance", *Magnetics Conference INTERMAG* Asia 2005. Digests of the IEEE International, Page(s): 1777 – 1778, 2005.

[42] Tai-Sik Hwang; Jul-Ki Seok and Dong-Hun Kim. "High performance control of π/4- multiple-pitched linear hybrid stepping motor with ripple force compensator", *Annual Conference of IEEE Industrial Electronics Society*, 2005.

[43] Ji Wang; Wenke Hu; Wenhua Zhao; Jianke Du and Dejin Huang. "An analysis of quartz crystal resonators with mindlin plate theory based parallel finite element method on a Linux cluster", *Frequency Control Symposium*, Page(s): 592 – 595, 2008.

[44] Tarik Hacib, M. Rachid Mekideche and Nassira Ferkha. "Defect Identification Using Artificial Neural Networks and Finite Element Method", *International Conference on E-Learning in Industrial Electronics*1ST IEEE, Page(s): 29 – 33, 2006.

[45] Shan Wan, Lei Zhang and Qijun Zhang. "Application of Artificial Neural Networks for Electromagnetic Modeling and Computational Electromagnetics", *Symposium on Circuits and Systems*, Page(s): 743 – 746, 2008.

[46] Pradeep Ramuhalli and Lalita Udpa. "Electromagnetic NDE Signal Inversion by Function Approximation Neural Networks", I*EEE Transactions on Magnetics*, Page(s): 3633 – 3642, 2002.

[47] L.Mokbnach and A.Boubakeur. "Comparison of different back-propagation algorithms used in the diagnosis of transformer oil". *Annual Report Conference on Electrical Insulation and Dielectric Phenomena*, Page(s): 244 – 247, 2002.

[48] Tomasz Chady, Masato Enokizono, Ryszard Sikora, Takashi Todaka, and Yuji Tsuchida. "Natural Crack Recognition Using Inverse Neural Model and Multi-Frequency Eddy Current

Method”, *IEEE Transactions on Magnetics.* Page(s): 2797 – 2799, 2001.

[49] Enzo Coccorese, Raffaele Martone, and F. Carlo Morabito. “A Neural Network Approach for the Solution of Electric and Magnetic Inverse Problems”, *IEEE Transactions on Magnetics*. Page(s): 2829 – 2839, 1994.

[50] Yamashita, H.; Kowata, N.; Cingoski, Vand Kaneda, K. “Direct Solution Method for Finite Element Analysis Using Hopfield Neural Network”, *IEEE Transactions on Magnetics.* Page(s): 1964 – 1967, 1995.

[51] Pradeep Ramuhalli, Lalita Udpa and Satish S. Udpa. “Finite-Element Neural Networks for Solving Differential Equations”, *IEEE Transactions on Neural Networks*. Page(s): 1381 – 1392, 2005.

[52] Low, T.S. and Chao, B.; “The Use of Finite Elements and Neural Networks for the Solution of Inverse Electromagnetic Problems”, *IEEE Transactions on Magnetics*. Page(s): 2811 – 2813, 1992.

[53] De Alcantara, N.P., Jr.; de Carvalho, A.M. and Ulson, J.A.C.; “Investigations into the use of the finite element method and artificial neural networks in the non-destructive analysis of metallic tubes”, *International Joint Conference on Neural Networks*, IJCNN '02. Page(s): 1450 – 1454, 2002.

[54] I.T. Rekanos and T.D. Tsiboukis. “A parallel method based on the: hopfield neural network for the solution of the finite elements method equations”, *Symposium on Applied Electromagnetism*, 1996. Page(s): FSTE_16 - FSTE_16, 1996.

[55] Zhao Y.X.; Lingyun Chai; Kagawa Y. “Neural Network Approach to the Flaw Identification in a Conductive Plate with Finite Element Modeling”. *International Conference on System Simulation and Scientific Computing,* Asia Simulation Conference. Page(s): 18 – 22, 2008.

[56] Sikora, R.; Sikora, J.; Cardelli, E. and Chady, T.; “Artificial neural network application for material evaluation by electromagnetic methods”, *International Joint Conference on Neural Networks*, Page(s): 4027 - 4032 vol.6, 1999.

[57] Lagaris, I.E.; Likas, A. and Fotiadis, D.I. “Artificial neural networks for solving ordinary and partial differential equations”, IEEE Transactions on Neural Networks. Page(s): 987 – 1000, 1998.

[58] Chari. M. V. K., Salon. S. J., “Numerical methods in electromagnetism”, *CA: Academic*, San Diego, 2000.

[59] Silvester. P. P. and Ferrari. R. L., “Finite elements for electrical engineers”, *Univ Press*, Cambridge, 1996.

[60] http://en.wikipedia.org