CHAPTER 3
TOOLS AND TECHNIQUES USED IN THE PRESENT STUDY

3.1 FINITE ELEMENT SIMULATION

The finite element method, sometimes referred to as finite element analysis, is a computational technique used to obtain approximate solutions of boundary value problems in engineering. A boundary value problem is a mathematical problem in which one or more dependent variables must satisfy a differential equation everywhere within a known domain of independent variables and satisfy specific conditions on the boundary of the domain. Boundary value problems are also sometimes called field problems. The field is the domain of interest and most often represents a physical structure. The field variables are the dependent variables of interest governed by the differential equation. The boundary conditions are the specified values of the field variables (or related variables such as derivatives) on the boundaries of the field. Depending on the type of physical problem being analyzed, the field variables may include physical displacement, temperature, heat flux, and fluid velocity to name only a few.

3.1.1 General Procedure for Finite Element Analysis

Certain steps in formulating a finite element analysis of a physical problem are common to all such analyses, whether structural, heat transfer, fluid flow, or some other problem. These steps are embodied in commercial finite element software packages and are described as follows.
3.1.1.1 Preprocessing

The preprocessing step is, quite generally, described as defining the model and includes

- Define the geometric domain of the problem.
- Define the element type(s) to be used.
- Define the material properties of the elements.
- Define the geometric properties of the elements (length, area, and the like).
- Define the element connectivity (mesh the model).
- Define the physical constraints (boundary conditions).
- Define the loadings.

The preprocessing (model definition) step is critical. A perfectly computed finite element solution is of absolutely no value if it corresponds to the wrong problem.

In this research work, first a parametric CAD model of the assemblies was created in Pro/E wildfire 5.0 software. Then ASCIS file (*.sat) type of the model is generated, which is imported into the ANSYS version 11 software. Once a three dimensional model of the assembly is available, the next step is to develop a finite element model of the same to determine deformation of various components. The assembly consists of few components and there exists some contact pairs among the components. The elements used in the analysis are SOLID 92 (Figure 3.1) which has quadratic displacement behavior and is well suited to model irregular meshes such as produced from CAD data.
The element is defined by ten nodes having three degrees of freedom at each node. The element has plasticity, creep, swelling, stress stiffening, large deflection and large strain capabilities. The elements used to define the four contact pairs are CONTA173 (Figure 3.2) and TARGE170 (Figure 3.3).
The CONTA173 is normally used to represent contact and sliding between 3-D "target" surfaces defined by TARGE170 and a deformable surface, defined by this element. The element is applicable to 3-D structural and coupled field contact analyses. This element is located on the surfaces of 3-D solid or shell elements without mid side nodes. It has the same geometric characteristics as the solid or shell element face with which it is connected. Contact occurs when the element surface penetrates one of the target segment elements on a specified target surface.

The material properties required for this analysis are given in Table 3.1.
Table 3.1 List of material properties

<table>
<thead>
<tr>
<th>S.No</th>
<th>Material Property</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Modulus of Elasticity</td>
<td>$E$</td>
</tr>
<tr>
<td>2</td>
<td>Poisson’s ratio</td>
<td>$\nu$</td>
</tr>
<tr>
<td>3</td>
<td>Mass density</td>
<td>$\rho$</td>
</tr>
<tr>
<td>4</td>
<td>Thermal coefficient of expansion</td>
<td>$\alpha$</td>
</tr>
</tbody>
</table>

The types of loads applied in this analysis are gravity and angular velocity in order to account for the inertia effect and temperature in order to account for the effect of change in operating temperature.

3.1.1.2 Solution

During the solution phase, finite element software assembles the governing algebraic equations in matrix form and computes the unknown values of the primary field variable(s). The computed values are then used by back substitution to compute additional, derived variables, such as reaction forces, element stresses, and heat flow. Since a finite element model has to be represented by tens of thousands of equations, special solution techniques are used to reduce data storage requirements and computation time. For static, linear problems, a wave front solver, based on Gauss elimination, is commonly used.

3.1.1.3 Postprocessing

Analysis and evaluation of the solution results are referred to as postprocessing. Postprocessor software contains sophisticated routines used
for sorting, printing, and plotting selected results from a finite element solution. Examples of operations that can be accomplished include

Sort element stresses in order of magnitude.

Check equilibrium.

Calculate factors of safety.

Plot deformed structural shape.

Animate dynamic model behavior.

Produce colour-coded temperature plots.

While solution data can be manipulated in many ways in postprocessing, the most important objective is to apply sound engineering judgment in determining whether the solution results are physically reasonable.

3.1.2 Error Estimation

Finite element analysis is a computational technique used to obtain approximate solutions of boundary value problems in engineering. The solution obtained by finite element analysis can be made to converge to the exact solution by selecting an appropriate higher order element for meshing and by refining mesh density. In this research a higher order element named SOLID 92 is used. In order to improve mesh density, following procedure has to be applied to determine error, because the quality of mesh (mesh density) can be judged by estimating the error. In case of stress analysis, the error is calculated by four methods. They are
3.1.2.1 Percentage error in energy norm (SEPC)

SEPC is a rough estimate of the stress error over the entire set of selected elements. This method can be used to compare models of similar structures subjected to similar loadings. SEPC is shown in legend column of deformed shape displays. They can also be listed manually. As a general rule of thumb, SEPC has to be 10% or less. If it is higher, then check for stress singularities.

A stress singularity is a location in a finite element model where the stress value is unbounded (infinite). Stress singularities exist due to following conditions, they are as follows;

- A point load, such as an applied force or moment
- An isolated constraint point, where the reaction forces behaves like a point load
- A sharp re-entrant corner (with zero fillet radius)
- As the mesh density is refined at a stress singularity, the stress value increases and never converges. Stress singularities are handled by adopting suitable corrective actions. They are as follows;
  - If stress singularities are located far away from the region of interest then the analyst can simply ignore them by deactivating the affected zone while reviewing results.
  - If they are located in the region of interest then they can be corrected by adding a fillet at re-entrant corners and rerunning the analysis.
Stress singularities can also be corrected by replacing a point force with an equivalent pressure load or by ‘spreading out’ displacement constraints over a set of nodes.

3.1.2.2 Element stress deviations (SDSG)

SDSG is a measure of the amount by which an element’s stress disagrees with stress average at its nodes. The finite element solution calculates stresses on a per-element basis, i.e., stresses are individually calculated in each element. While plotting nodal stress contours, the stresses are averaged at nodes resulting in smooth contours. However, while plotting element solution, the discontinuity between elements will be revealed by un-averaged data. The difference between averaged and un-averaged stresses gives an indication of how ‘good’ or how ‘bad’ the mesh is.

3.1.2.3 Element energy error (SERR)

SERR is the energy associated with the stress that mismatches at the nodes of the element. This is the basis error measure from which the other error quantities are derived. Generally, the elements with the highest SERR are areas for mesh refinement. Usually SERR will be highest at stress singularities.

3.1.2.4 Maximum and minimum stress bounds (SMXB, SMNB)

The stress bounds are used to determine the potential effect of mesh discretization error on the maximum stress. They are displayed on stress contour plots in the legend column as SMXB (upper bound) and SMNB (lower bound). The bounds are not estimates of the actual maximum and minimum, but they define a ‘confidence band’.
3.2 NEURAL NETWORK

Neural networks have received a lot of attention in many research studies and application areas. One of the major benefits of neural networks is the adaptive ability of their generalization of data from the real world. Exploiting this advantage, many researchers apply neural networks for nonlinear regression analysis and have reported positive experimental results in their applications. Recently, neural networks have received a great deal of attention in manufacturing areas. Zhang and Huang (1995) presented an extensive review of neural network applications in manufacturing. Neural networks are defined by Rumelhart and McClelland (1989) as massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with objects of the real world in the same way as biological nervous systems do. The neural network approach can be regarded as a statistical method. The feature hidden within the designed experiment can be learnt by the neural network approach based on the collected data. In addition, a neural network can be constructed without requiring any assumptions concerning the functional form of the relationship between predictors and responses (Stern 1996). Therefore, the neural network approach outperforms the conventional statistical modeling approach in terms of analyzing experimental data (Su et al 1998). Among several well-known supervised neural networks, the Back-Propagation model (BP) is the most extensively used and can provide good solutions for many industrial applications (Lippmann, 1987). Back propagation network has been applied in tolerance synthesis problem to fit the cost-tolerance relationship (Mu-Chen chen 2001).

A BP network is a feed-forward network with one or more layers of nodes between the input and output nodes. An imperative item of the BP network is the iterative method that propagates the error terms required
to adopt weights back from nodes in the output layer to nodes in lower layers. The training of a BP network involves three stages: the feed forward of the input training pattern, the calculation and BP of the associated error, and the adjustment of the weights. After the network reaches a satisfactory level of performance, it will learn the relationships between input and output patterns and its weights can be used to recognize new input patterns.

Figure 3.4 depicts a BP network with one hidden layer. The hidden nodes of the hidden layer perform an important role in creating internal representation. The following nomenclatures are used for describing the BP learning rule.

\[ \text{net}_{pi} = \text{net input to processing unit } i \text{ in pattern } p \text{ (a pattern corresponding to a vector of factors)}, \]

Figure 3.4 Architecture of a three-layer BP network

(Mu-Chen Chen 2001)
\( w_{ij} = \) connection weight between processing unit \( i \) and processing unit \( j \),

\( a_{pi} = \) activation value of processing unit \( i \) in pattern \( p \),

\( \delta_{pi} = \) the effect of a change on the output of unit \( i \) in pattern \( p \),

\( g_{pi} = \) target value of processing unit \( i \),

\( \varepsilon = \) learning rate.

The net inputs and the activation values of the middle processing nodes are calculated as follows:

\[
\text{net}_{pi} = \sum_j w_{ij} a_{pj},
\] (3.1)

\[
a_{pi} = \frac{1}{1 + \exp \left( \text{net}_{pi} \right)}
\] (3.2)

The net input is the weighted sum of activation values of the connected input units plus a bias value. Initially, the connection weights are assigned randomly and are varied continuously. The activation values are in turn used to calculate the net inputs and the activation values of the output processing units using the same Equations (3.1) and (3.2).

Once the activation values of the output units are calculated, we compare the target value with activation value of each output unit. The discrepancy is propagated using.

\[
\delta_{pi} = \left( g_{pi} - a_{pi} \right) f'(\text{net}_{pi})
\] (3.3)
For the hidden processing units in which the target values are unknown, instead of Equation (3.3), the following equation is used to calculate the discrepancy. It takes the form

\[ \delta_{pi} = f_{i}'(\text{net}_{pi})\sum_{k} \delta_{pk} w_{ki}. \] (3.4)

From the results of Equations (3.3) and (3.4), the weights between processing units are adjusted using

\[ \Delta w_{ij} = \varepsilon \hat{\delta}_{pi} a_{pj}. \] (3.5)

BP network architecture and learning rule are presented in this chapter. A cost–tolerance model based on neural network methods is proposed (chapter 7) in order to provide product designers and process planners with an accurate basis for estimating the manufacturing cost.

### 3.3 INTELLIGENT OPTIMIZATION TECHNIQUES

An Evolutionary Algorithm (EA) is a generic population based metaheuristic optimization algorithm. EA uses some mechanisms inspired by biological evolution: reproduction, mutation, recombination and selection. Candidate solution to the optimization problem plays the role of individuals in a population and the cost function determines the environment within which solutions are alive. Evolution of the population then takes place after repeated application of the above operators. The important evolutionary algorithm is Elitist Non–dominated Sorting Genetic Algorithm (NSGA-II).

Intelligent optimization algorithms such as Elitist Non–dominated Sorting Genetic Algorithm (NSGA-II) are very much needed for the tolerance design for a mechanical assembly, since tolerance design for a real world
mechanical assembly is a very complex and tedious task (Singh et al 2008). This is due to the following reasons.

1. The tolerance design algorithm has to consider the simultaneous selection of design and manufacturing tolerances in a concurrent engineering environment.

2. The integrated optimal tolerance design method should consider adjustment of nominal dimensions, Dimensional and Geometrical Tolerances (DGTs) and selection of design and manufacturing tolerances for minimum manufacturing cost and minimum quality loss.

3.3.1 Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II)

Deb et al (2002) proposed the NSGA-II algorithm. Essentially, NSGA-II differs from non-dominated sorting genetic algorithm (NSGA) in implementation in a number of ways. Firstly, NSGA-II uses an elite preserving mechanism, thereby assuring preservation of previously found good solutions. Secondly, NSGA-II uses a faster non-dominated sorting procedure. Thirdly, NSGA-II does not require any tunable parameter, thereby making the algorithm independent of the user.

Solution to any optimization problem is a family of points known as non-dominated solutions or Pareto-optimal set. The working principle of NSGA-II is as follows.

1. Initially a random parent population is created and then the population is sorted based on the non-domination.

2. A special book keeping procedure is used in order to reduce the computational complexity.
3. Each solution is assigned a fitness value equal to its non-domination level. Thus, minimization of fitness is assumed.

4. Binary tournament selection, recombination and mutation operators are used to create a child population. Thereafter the following algorithm is used in every generation.

The algorithm can be explained as follows. First a combined population is formed, thereby allowing parent population to be compared with child population ensuring elitism. Then the population is sorted according to non-domination. The new parent population is formed by adding solutions from the first front and continuing to other fronts successively till the size exceeds that of child population. Thereafter, the solutions of the last accepted front are sorted according to a crowded criterion and first points of child population size are picked. Then they are used for selection, crossover and mutation to create a new population. The above procedure is continued for a specified number of generations. It is clear from the above description that NSGA-II will outperform the other current elitist evolutionary algorithm because it uses a crowded comparison criterion in the tournament selection and in the phase of population reduction thereby preserving diversity. Figure 3.5 shows an iteration of NSGA-II procedure.

![Figure 3.5 An iteration of NSGA-II procedure (Deb et al 2002)]