SUPPORT OPTIMIZATION TOOL FOR AN AERO ENGINE CONFIGURATION SYSTEMS

A Project Report

Submitted by

Haran Pragalath D C

In partial fulfillment of the requirements

for the award of the Degree of

MASTER OF TECHNOLOGY

in

STRUCTURAL ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

MAY 2011

SUPPORT OPTIMIZATION TOOL FOR AN AERO ENGINE CONFIGURATION SYSTEMS

A Project Report

Submitted by

Haran Pragalath D C

In partial fulfillment of the requirements
for the award of the Degree of

MASTER OF TECHNOLOGY
In
STRUCTURAL ENGINEERING

Under the guidance of

Prof. Pradip Sarkar (Associate Professor, NITR)
Mr. Naleen Kumar Verma (Lead Engineer, GE)

DEPARTMENT OF CIVIL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA-769008,
2010

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

CERTIFICATE

*This is to certify that the thesis entitled, "**SUPPORT OPTIMIZATION TOOL FOR AERO ENGINE CONFIGURATION SYSTEMS**" submitted by Haran Pragalath D.C. in partial fulfillment of the requirements for the award of Master of Technology Degree in Civil Engineering with specialization in "**Structural Engineering**" at National Institute of Technology, Rourkela is an authentic work carried out by him under our supervision and guidance. To the best of my knowledge, the matter embodied in this Project review report has not been submitted to any other university/ institute for award of any Degree or Diploma.*

***Date**: 24th MAY 2010*
***Place**: Bangalore*

(Prof. Pradip Sarkar)
Dept. of Civil Engineering,
National Institute of Technology,
Rourkela-769008.

(Naleen Kumar Verma)
GE India,
Bangalore.

GE

John F Welch Technology Centre
122, EPIP, Whitefield Road
Bangalore 560 066
India

T 91 80 2503 1000
F 91 80 2841 2111

May 30, 2011

To Whomsoever It May Concern

This is to certify that D. C. HARAN PRAGALATH worked as an intern at GE India Technology Centre, Bangalore, from 13-Sep-2010 to 25-Mar-2011. He worked with the Product Integration Centre team of GE Aviation and worked on a project titled ""SUPPORT OPTIMIZATION TOOL FOR AERO ENGINE CONFIGURATION SYSTEMS"

Thanking You.

For GE India Technology Center Private Limited.

Shriram Barve
Team Manager, Product Integration Centre,
GE Aviation Bangalore.

ACKNOWLEDGEMENT

I express my sincere gratitude and thankful to Mr. Shriram Govind Barve, Mr. Naleen Kumar Verma, Mr. Muthu Krishanan Muniyandi, Mr. Siva Prasad and Aditya Botla, for their guidance and constant encouragement and support during the course of my work in the last one year. I am also very thankful towards all my team members for their help and encouragement during the project. I truly appreciate and value their esteemed guidance and encouragement from the beginning to the end of this thesis, their knowledge and company at the time of crisis remembered lifelong.

I express my sincere thanks to Mr. Vasanthkumar Isaac, who given an opportunity to do my project in reputed company and My sincere thanks to Prof M. Panda, Head of the Civil Engineering Department, and Dr. Pradip Sarkar, Associate Professor, National Institute of Technology Rourkela, for allowing me outside and providing necessary facility for my work. I am also thankful to all faculties of the Civil Department for their support and involvement in my project work.

I also thank all my batch mates who have directly or indirectly helped me in my project work.

Finally yet importantly, I would like to thank my parents, who taught me the value of

Hard work by their own example. I would like to share this moment of happiness with my father, mother & brother. They rendered me enormous support during the whole tenure of my stay in NIT Rourkela and in Bangalore.

Date: 24-May-2011 D.C. Haran Pragalath

Place: Rourkela Roll No: 209Ce2039

ABSTRACT

In an aircraft engine, basic functionality of configuration system is to form the network of supply lines for air, fuel & oil. Configuration hardware mainly consists of tubes, ducts, valves & support brackets. Many of the Aircraft engines are failed due to failure of fuel pipes and caused big hazards. Hence to avoid accidents, hard wares are designed properly against the failures. In order to use the resources properly optimum design of hard wares is necessary.

The two basic design requirements for configuration systems are: (a) high cycle fatigue (vibratory stresses) and (b) low cycle fatigue (thermally induced stresses). Under these design requirements optimum design can be done through the selection of critical design parameters. Critical design parameters in a configuration system considered in the present study are: (i) Support location for the tube layout and (ii) Bracket support type at this location. The judicious selections of these two design parameters greatly influence the high cycle fatigue and low cycle fatigue life. This also helps in reducing the weight and cost of the system.

Design optimisation is an iterative process and computer software can help to obtain an optimum design. In the present study, user friendly software is developed using TCL/TK environment. This is linked with a commercial software ANSYS and PEZ tool. ANSYS is used to carry out the structural analysis considering preliminary support locations. Analysis results from ANSYS are the inputs for the optimization analysis. PEZ tool modifies the support locations based on the optimization results. ANSYS will analyse the structure once again based on the modified support locations. This loop will continue till target reaches.

CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The need for improvements in engineering designs especially for aerospace and automotive structures is nowadays becoming a major industry request. Deterministic approaches are unable to take into account all the variabilities that characterize design input properties without leading to oversized structures. This report intends to present case studies of design optimization for various hard wares in aero-engine.

Currently, aerospace and automotive industries are dedicating a lot of attention to improve product quality and reliability already in a virtual simulation environment. The new design process that is needed to build these structures requires also a shift from the traditional design approach to a new approach that integrates all the variabilities and uncertainties. Also the industry is facing new challenges, especially in the field of durability and reliability of structural components, in order to achieve better performances together with improved safety.

1.2 OBJECTIVE

To develop an Auto Optimization Tool for an Aero Engine configuration Systems having Pipes, ducts and Brackets during Preliminary design Stage.

Reducing the number of supports and the reducing the thickness of brackets can achieve it.

1.3 SCOPE

Design the Aero Engine Hardware components at optimum level.

Reduce the work time of Analysis and Designing, for an Aero Engine Hardware.

1.4 METHODOLOGY

Development of software tools required clear understanding of programming language and the domain knowledge. Work on both the fronts was initiated in parallel. Software tool on Optimization concept was undertaken after studying various Optimization algorithms. The programming language used was ‘TCL/TK’ and ‘PEZ’ and the platform used was ‘ANSYS’. The tool developed will be extensively use by team members and GE employers. All models and formulations have been coded for ease of parametric study and for future use by researchers. Therefore it is evident that, lot of effort has gone in understanding of computer programming, development of logics, and testing of these computer programs.

1.5 VARIOUS FORCES ACTING ON AEROPLANE

The various forces acting on an airplane are Lift from Wings, Thrust from Engines Drag from Friction and Weight. Where, the Thrust is a change in momentum divided by time. Fig. 1 presents all the forces pictorially.

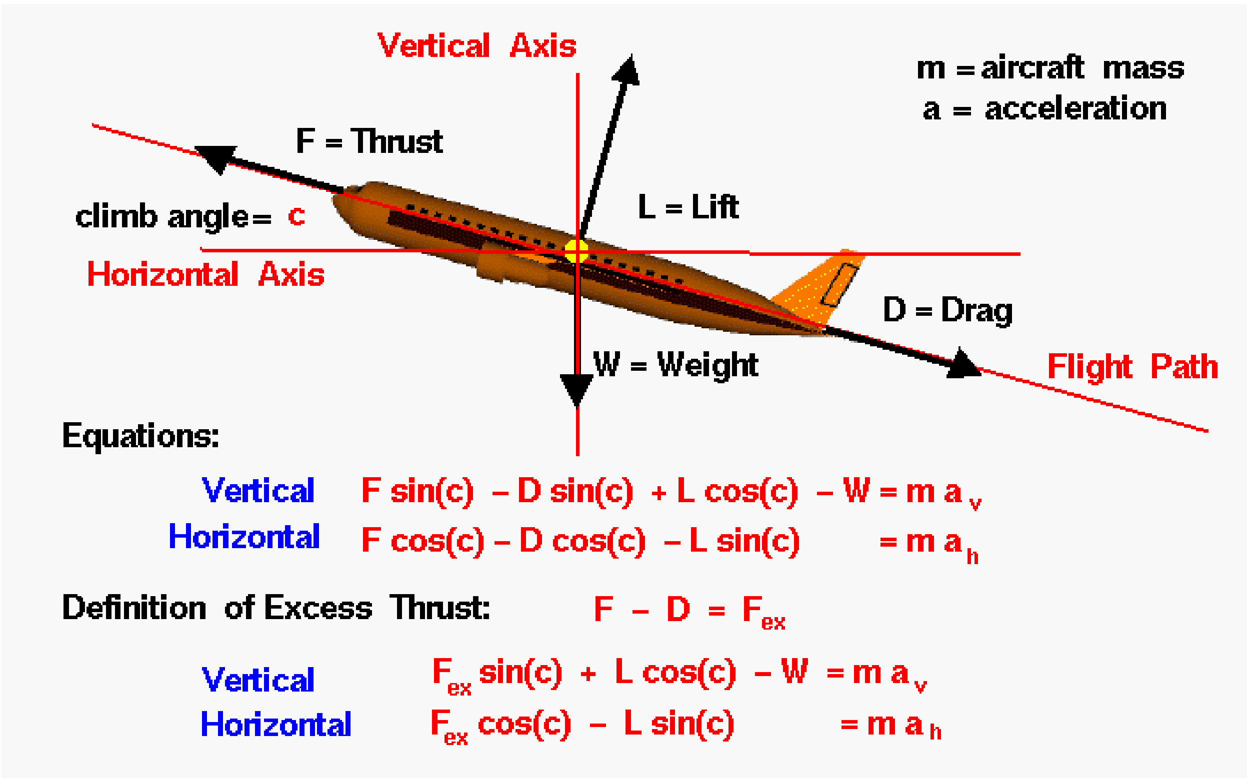


Fig. 1.1: The various forces acting on an aero plane

1.6 AERO ENGINE CONFIGURATION SYSTEM OVERVIEW

Fig. 1.2 shows a cross sectional view of aero engine which consists mainly on (1). AirIntake Section, (2) Compression Section, (3) Combustion Section, (4) Turbine Section, (5) Exhaust Section and (6) Exhaust Diffuser

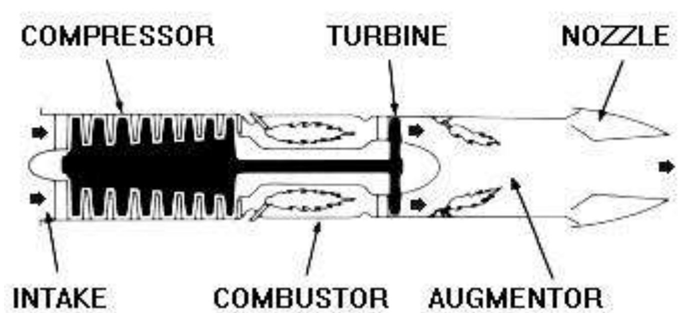


Fig. 1.2: Cross section view of an aero-engine (http://en.wikipedia.org)

1.6.1 Operations of Aero-engine

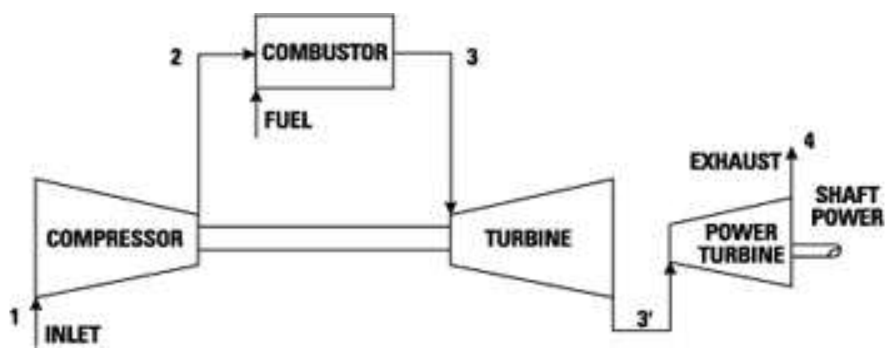


Fig. 1.3: Cross section view of an aero-engine (http://en.wikipedia.org)

A gas turbine has a compressor to draw in and compress air a combustor (or burner) to add fuel to heat the compressed air and a turbine to extract power from the hot air flow. The gas turbine is an internal combustion (IC) engine employing a continuous combustion process. This differs from the intermittent combustion occurring in diesel and automotive IC engines. About 2/3rds of the shaft power produced by the turbine is used to run the compressor, leaving about 1/3rd available to turn a genset to produce electrical power.

Generally, Air is drawn into the engine through the front intake. The compressor squeezes the air to many times normal atmospheric pressure and forces it into the combustor. Here, fuel is sprayed into the compressed air, is ignited and burned continuously like a blowtorch. The burning gases expand rapidly rearward and pass through the turbine. The turbine extracts energy from the expanding gases to drive the compressor, which intakes more air. After leaving the turbine, the hot gases exit at the rear of the engine, giving the aircraft its forward push, action, reaction.

For additional thrust or power, an afterburner can be added. Additional fuel is introduced into the hot exhaust and burned with a resultant increase of up to 50 percent in engine thrust by way of even higher velocity and more push.

CHAPTER II

MODELLING AND ANALYSIS

CHAPTER 2

MODELLING AND ANALYSIS

2.1 VARIOUS HARDWARE IN AERO ENGINE

In an aircraft engine, basic functionality of configuration system is to form the network of supply lines for air, fuel & oil. Configuration hardware mainly consists of tubes, ducts, valves & support brackets. Fig 2.1 show the Various Hardware used in aero engine

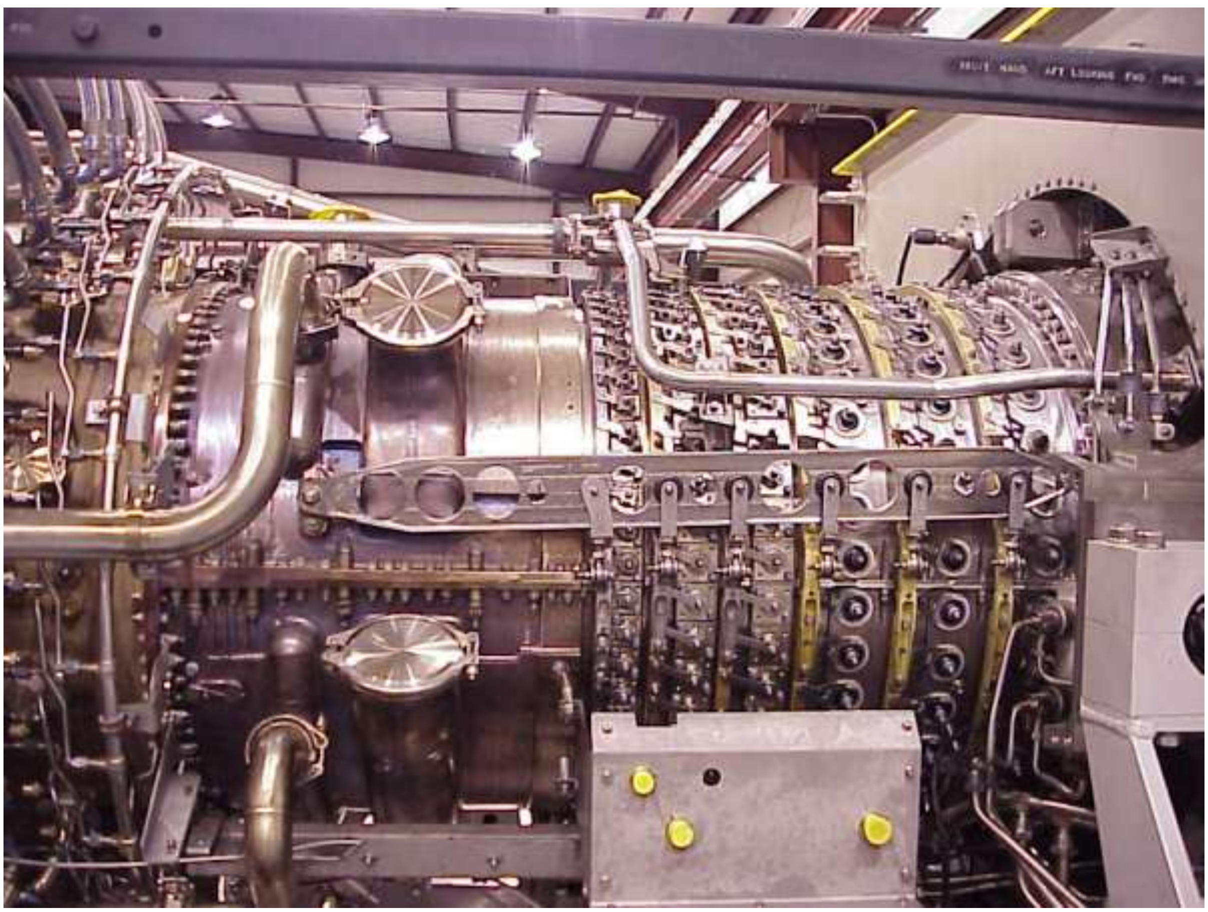


Fig. 2.2: Aero Engine (http://i1.lelong.com.my/Malaysia/-0802-16-asfanb%4017.jpg)

2.2 BASIC ENGINE DESIGN CONSIDERATIONS

The process of developing an engine is one of compromises. Engineers design specific attributes into engines to achieve specific goals. Aircraft are one of the most demanding applications for an engine, presenting multiple design requirements, many of which conflict with each other. An aircraft engine must be:

- *reliable*, as losing power in an airplane is a substantially greater problem than in an automobile. Aircraft engines operate at temperature, pressure, and speed extremes, and therefore need to perform reliably and safely under all reasonable conditions.
- *light weight*, as a heavy engine increases the empty weight of the aircraft and reduces its payload.
- *powerful*, to overcome the weight and drag of the aircraft.
- *small* and *easily streamlined*; large engines with substantial surface area, when installed, create too much drag.
- *field repairable*, to keep the cost of replacement down. Minor repairs should be relatively inexpensive and possible outside of specialized shops.
- *fuel efficient* to give the aircraft the range the design requires.
- capable of operating at sufficient altitude for the aircraft

The two main design requirements for configuration systems are:

1) Design for High cycle fatigue (Vibratory stresses)

2) Design for Low cycle fatigue (Thermally induced stresses)

2.2.1 FATIGUE

Fatigue is the progressive and localized structural damage that occurs when a material is subjected to cyclic loading. The nominal maximum stress values are less than the ultimate tensile stress limit, and may be below the yield stress limit of the material. Fatigue occurs when a material is subjected to repeated loading and unloading. If the loads are above a certain threshold, microscopic cracks will begin to form at the surface. Eventually a crack will reach a critical size, and the structure will suddenly fracture. The shape of the structure will significantly affect the fatigue life; square holes or sharp corners will lead to elevated local stresses where fatigue cracks can initiate. Round holes and smooth transitions or fillets are therefore important to increase the fatigue strength of the structure.

CHARACTERISTICS OF FATIGUE

- In metals and alloys, the process starts with dislocation movements, eventually forming persistent slip bands that nucleate short cracks.
- Fatigue is a stochastic process, often showing considerable scatter even in controlled environments.
- The greater the applied stress range, the shorter the life.
- Fatigue life scatter tends to increase for longer fatigue lives.
- Damage is cumulative. Materials do not recover when rested.
- Fatigue life is influenced by a variety of factors, such as temperature, surface finish, microstructure, presence of oxidizing or inert chemicals, residual stresses, contact (fretting), etc.

- Some materials (e.g., some steel and titanium alloys) exhibit a theoretical fatigue limit below which continued loading does not lead to failure.
- In recent years, researchers have found that failures occur below the theoretical fatigue limit at very high fatigue lives (10^9 to 10^{10} cycles). An ultrasonic resonance technique is used in these experiments with frequencies around 10–20 kHz.
- High cycle fatigue strength (about 10^3 to 10^8 cycles) can be described by stress-based parameters. A load-controlled servo-hydraulic test rig is commonly used in these tests, with frequencies of around 20–50 Hz. Other sorts of machines like resonant magnetic machines can also be used, achieving frequencies up to 250 Hz.
- Low cycle fatigue (typically less than 10^3 cycles) is associated with widespread plasticity in metals; thus, a strain-based parameter should be used for fatigue life prediction in metals and alloys. Testing is conducted with constant strain amplitudes typically at 0.01–5 Hz.

2.2.2 HIGH-CYCLE FATIGUE

Historically, most attention has focused on situations that require more than 10^4 cycles to failure where stress is low and deformation primarily elastic.

The S-N curve

In high-cycle fatigue situations, materials performance is commonly characterized by an *S-N curve*, also known as a *Wöhler curve* . This is a graph of the magnitude of a cyclic stress (*S*) against the logarithmic scale of cycles to failure (*N*).

S-N curves are derived from tests on samples of the material to be characterised (often called *coupons*) where a regular sinusoidal stress is applied by a testing machine, which also counts the number of cycles to failure. This process is sometimes known as *coupon testing*. Each coupon test generates a point on the plot though in some cases there is a *runout* where the time to failure exceeds that available for the test. Analysis of fatigue data requires techniques from statistics, especially survival analysis and linear regression.Fig 2.1 shows the S-N curve

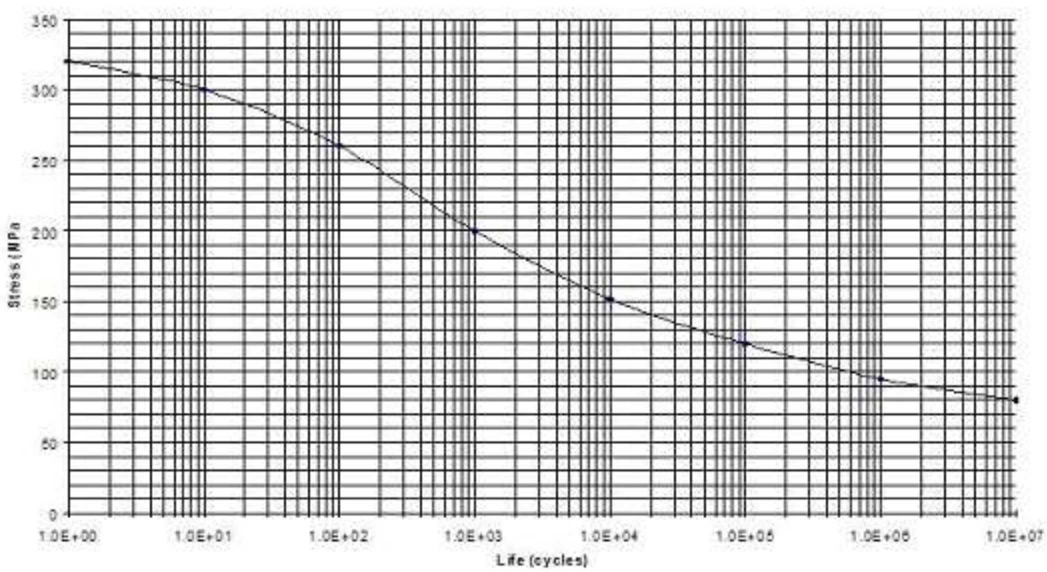


Fig. 2.2: S-N Curve (http://en.wikipedia.org)

2.2.3 LOW-CYCLE FATIGUE

Where the stress is high enough for plastic deformation to occur, the account in terms of stress is less useful and the strain in the material offers a simpler description. Low-cycle fatigue is usually characterised by the *Coffin-Manson relation* .Fig 2.2 shows the graph between Vibratory Stress and Steady State stress

$$\frac{\Delta\epsilon_p}{2} = \epsilon_f'(2N)^c$$

Where

- $\Delta\varepsilon_p$ /2 is the plastic strain amplitude;
- ε_f' is an empirical constant known as the *fatigue ductility coefficient*, the failure strain for a single reversal;
- 2*N* is the number of reversals to failure (*N* cycles);
- *c* is an empirical constant known as the *fatigue ductility exponent*, commonly ranging from -0.5 to -0.7 for metals in time independent fatigue. Slopes can be considerably steeper in the presence of creep or environmental interactions.

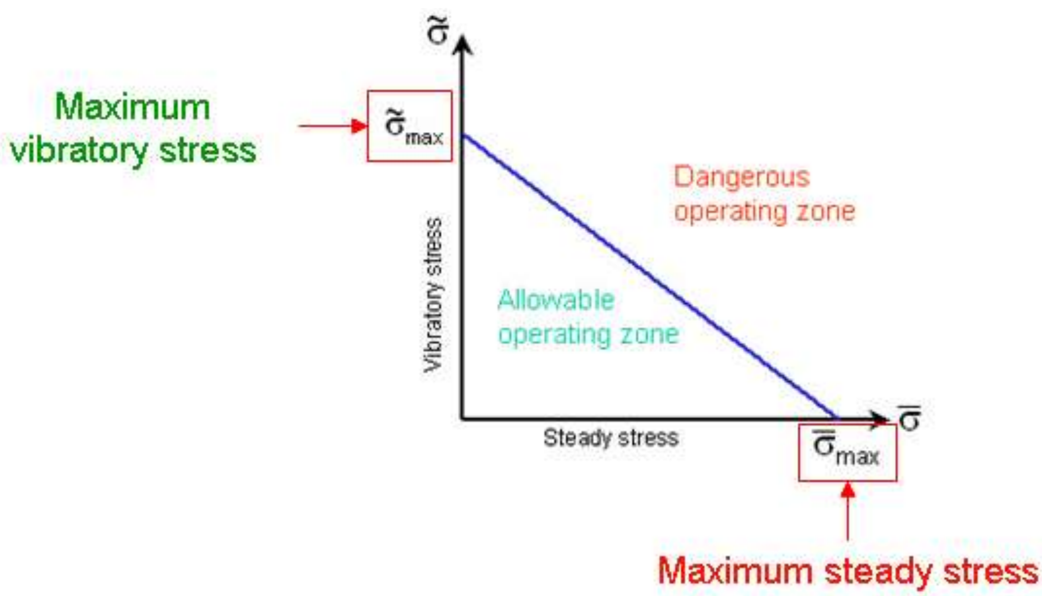


Fig. 2.2: Stress Curve (Source: www.energy.kth.se)

2.2.4 FACTORS THAT AFFECT FATIGUE-LIFE

- **Cyclic stress state:** Depending on the complexity of the geometry and the loading, one or more properties of the stress state need to be considered, such as stress amplitude, mean stress, biaxiality, in-phase or out-of-phase shear stress, and load sequence,
- **Geometry:** Notches and variation in cross section throughout a part lead to stress concentrations where fatigue cracks initiate.
- **Surface quality.** Surface roughness cause microscopic stress concentrations that lower the fatigue strength. Compressive residual stresses can be introduced in the surface by e.g. shot peening to increase fatigue life. Such techniques for producing surface stress are often referred to as *peening*, whatever the mechanism used to produce the stress. Low Plasticity Burnishing, Laser peening, and ultrasonic impact treatment can also produce this surface compressive stress and can increase the fatigue life of the component. This improvement is normally observed only for high-cycle fatigue.

- **Material Type:** Fatigue life, as well as the behaviour during cyclic loading, varies widely for different materials, e.g. composites and polymers differ markedly from metals.

- **Residual stresses:** Welding, cutting, casting, and other manufacturing processes involving heat or deformation can produce high levels of tensile residual stress, which decreases the fatigue strength.

- **Size and distribution of internal defects:** Casting defects such as gas porosity, non-metallic inclusions and shrinkage voids can significantly reduce fatigue strength.

- **Direction of loading:** For non-isotropic materials, fatigue strength depends on the direction of the principal stress.

- **Grain size:** For most metals, smaller grains yield longer fatigue lives, however, the presence of surface defects or scratches will have a greater influence than in a coarse grained alloy.

- **Environment:** Environmental conditions can cause erosion, corrosion, which all affect fatigue life. Corrosion fatigue is a problem encountered in many aggressive environments.

- **Temperature:** Extreme high or low temperatures can decrease fatigue strength.

2.2.5 DESIGN AGAINST FATIGUE

Dependable design against fatigue-failure requires thorough education and supervised experience in structural engineering, mechanical engineering, or materials science. There are three principal approaches to life assurance for mechanical parts that display increasing degrees of sophistication:

1. Design to keep stress below threshold of fatigue limit (infinite lifetime concept);
2. Design (conservatively) for a fixed life after which the user is instructed to replace the part with a new one (a so-called *lifted* part, finite lifetime concept, or "safe-life" design practice);
3. Instruct the user to inspect the part periodically for cracks and to replace the part once a crack exceeds a critical length. This approach usually uses the technologies of non-destructive testing and requires an accurate prediction of the rate of crack-growth between inspections. This is often referred to as damage tolerant design or "retirement-for-cause".

2.3 MODELLING

The entire configuration system is modelled in UniGraphics. Each component is defined by three categories,

a) Geometry- dimensional

b) Properties- Materials, weight

c) Environment- Temperature, Pressure, Fluid type

2.3.1 COMPONENT ATTRIBUTES

	GEOMETRY	PROPERTIES	ENVIRONMENT
Brackets	Thickness	Material	Temperature
Tubes	Thickness and Diameter	Material	Temperature, Fluid Type and Pressure
Ducts	Thickness and Diameter	Material	Temperature, Fluid Type and Pressure

All components faceted bodies are eliminated:

- Faceted bodies are not readily meshed and often cause errors.
- Components occupying the same space make geometry selection for meshing difficult.

The system contains solid, sheet bodies and wire frames. All nut plates are rivet holes from the brackets are removed. Fig. 2.3 Shows the FEM Model of configuration hardware.

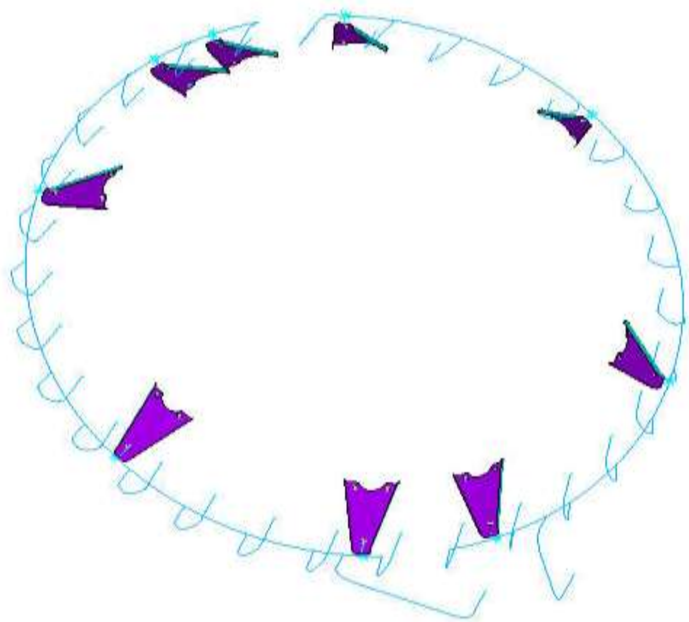


Fig.2.3 FEM Model

2.3.2. FINITE ELEMENT MODELING

The entire configuration system modelled as discussed in previous section is imported to ANSYS through IGES format.

Modelling of entire system is done using Shell –43, Pipe-16 and Pipe-18 elements.

2.3.2.1 Bracket Modelling

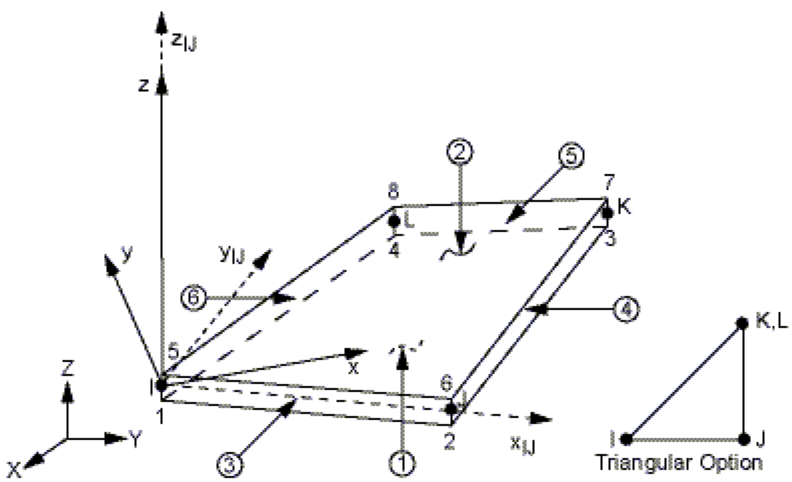
Brackets are modelled using Shell-43 elements in ANSYS. Mid surface of the bracket is meshed with quadrilateral elements. This is simplification for modelling since Shell 43 element is based on the mid surface not the edge. Material Properties and Temperatures are assigned to the bracket elements. Brackets are fixed in all degrees of freedom at the bolthole. Fig.5 Show the Shell-43 element

2.3.2.2 Tube Modelling

When the Diameter of the Cross section is ≤ than 1 inch it is called as Tubes. Tubes are modelled using Pipe-16 and Pipe-18 elements in ANSYS. These tubes are meshed with line elements. Material Properties, Pressure and Temperatures are assigned to the bracket elements. Fig.5 and 6 Show the Pipe-16 and Pipe-18 element.

2.3.2.2 Duct Modelling

When the Diameter of the Cross section is > than 1 inch it is called as Duct. Ducts are modelled using Shell-43 elements in ANSYS. Mid surface of the Ducts are meshed with quadrilateral elements. Material Properties, Pressure and Temperatures are assigned to the Duct elements.



x_{IJ} = Element x-axis if ESYS is not supplied.

x = Element x-axis if ESYS is supplied.

Fig.2.4 Shell-43 element (Source: Ansys Help file)

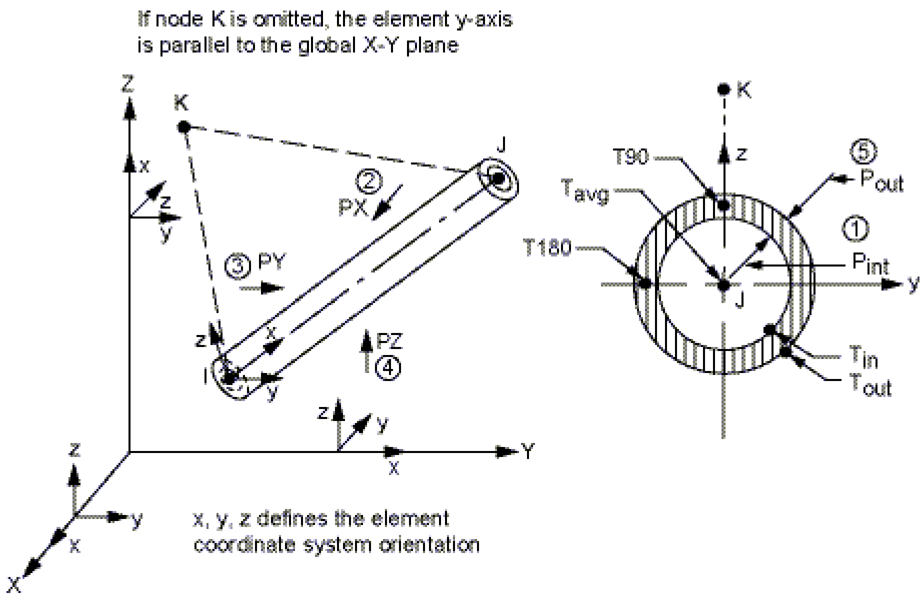


Fig.2.5 Pipe-16 element (Source: Ansys Help file)

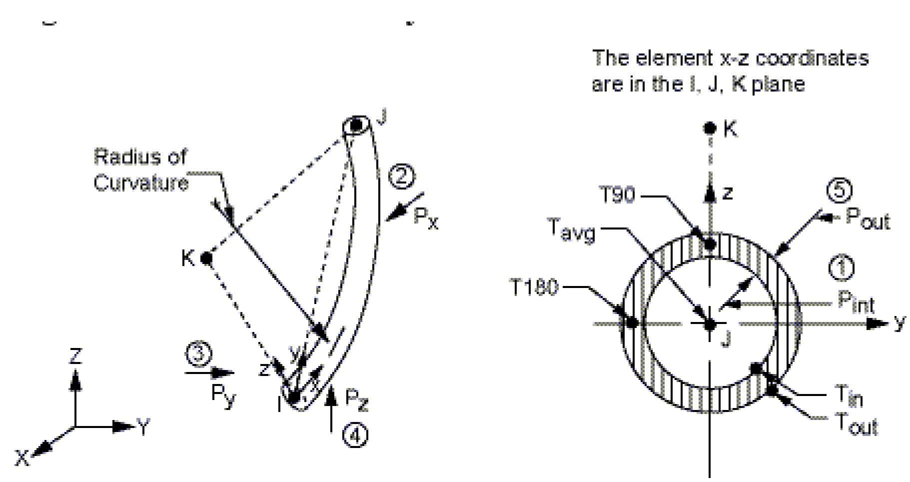


Fig.2.6 Pipe-18 element (Source: Ansys Help file)

1.3 ANALYSIS AND DESIGN PROCEDURE

1.3.1 Pre-Processor

Complete Finite Element Model with Boundary conditions in Ansys environment Apply all the loads such as Pressure, Temperature etc.

1.3.2 Solution

Perform Modal Analysis and the Spectrum Analysis.

1.3.3 Post Processing

Check the frequencies and mode shapes and compare the Stress results with the threshold values.

CHAPTER III

OPTIMIZATION

CHAPTER 3

OPTIMIZATION

In the design of technical systems and components it is of crucial importance to reduce costs, improve performances and system reliability and shorten time to market. The use of rigorous methods of decision-making, such as optimization methods, coupled with modern tools of computer-aided design may be effective in this. Especially for large and complex systems these Structured and advanced methods may become necessary. Furthermore, they lead to better solutions and enhance the creative process of conceptual and detailed design of technical systems.

3.1 STATEMENT OF AN OPTIMIZATION PROBLEM

Find x to minimize f (x) subject to g (x)

Here, x is an n-vector (has n components), f (the *objective function)* is a scalar and ***g*** **(**the *constraints)* is an m-vector.

Problems of this type are called *mathematical programming problems.*

$$\min f(\boldsymbol{x}) \text{ subject to } \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \qquad \text{or} \qquad \min f(\boldsymbol{x}) \mid \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}$$

$$\boldsymbol{g}(\boldsymbol{x}) \leq 0 \quad \Rightarrow \quad \begin{cases} g_1 \leq 0 \\ g_2 \leq 0 \\ \vdots \\ g_m \leq 0 \end{cases}$$

3.2 VARIOUS CLASSIFICATIONS IN OPTIMIZATION

3.2.1 Classification Based on the Existence of Constraints

It is classified as constrained or unconstrained, depending on whether it exists or not in the problem

3.2.2 Classification Based on the Nature of the Design Variables

- ✓ First Category – Prescribed functions, Constraints are constant (which is called *Static Optimization* problems)

$$X = \{b\ d\}^T$$

- ✓ Second Category – Which are continuous functions (which is called as *Dynamic Optimization* problems)

$$X = \{b\ (t)\ d\ (t)\}^T$$

3.3.3 Classification Based on the Nature of the Equations Involved

- Linear Programming - objective or constraint function are *linear*
- Nonlinear Programming - objective or constraint function are nonlinear (most general programming)
- Geometric Programming - objective or constraint function are polynomials
- Quadratic Programming – It is a nonlinear programming problem with a quadratic objective function

3.3.4 **Classification Based on the <u>Permissible Values of the Design Variables</u>**

- When some or all design variables are restricted to take only integer

3.3.5 Classification Based on the <u>Deterministic Nature of the Variables</u>

- Deterministic Programming – all parameters are deterministic
- Stochastic Programming (Probabilistic or nondeterministic) – some or all parameters are probabilistic

3.3.6 Classification Based on the <u>Separability of the Functions</u>

- Separable Programming – Objective function and constraints are separable

$f\ (x) = \Sigma\ f_i(x_i)$ subjected to $g\ (x) = \Sigma\ g_{ij}(x_i) =< b_j$,where j= 1,2,3,4…..m

- Non Separable Programming – Objective function and constraints are not separable

3.3.7 Classification Based on the <u>Number of Objective Functions</u>

- Single Programming Problems – It has only one Objective function
- Multi Programming Problems – It has more than one Objective function

$f_1(X),\ f_2(X),\ f_3(x), \ldots.. f_n(X)$

among so many methods Utility function method is so popular. It is defined for each objective depending on the importance of f_i to the other functions. Then overall utility function

$U = \Sigma U_I (f_i)$

$$= - \Sigma w f_i (X)$$

W_i is the weighing factor . This method is also known as weighting function method

3.3 GENETIC ALGORITHM

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

Genetic algorithms find application in bioinformatics, phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields.

A typical genetic algorithm requires:

1. a genetic representation of the solution domain,
2. a fitness function to evaluate the solution domain.

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem one wants to maximize the total value of objects that can be put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all objects in the knapsack if the representation is valid, or 0 otherwise. In some problems, it is hard or even impossible to define the fitness expression; in these cases, interactive genetic algorithms are used.

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, then improve it through repetitive application of mutation, crossover, inversion and selection operators.

3.2.1 Initialization

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

3.2.2 Selection

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming.

Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. Popular and well-studied selection methods include roulette wheel selection and tournament selection.

3.3.3 Reproduction

Crossover and Mutation

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired", some research suggests more than two "parents" are better to be used to reproduce a good quality chromosome.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

Although Crossover and Mutation are known as the main genetic operators, it is possible to use other operators such as regrouping, colonization-extinction, or migration in genetic algorithms.

3.3.4 Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria
- Fixed number of generations reached
- Allocated budget (computation time/money) reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Manual inspection
- Combinations of the above

3.3.5 Simple generational genetic algorithm pseudo code:

1. Choose the initial population of individuals
2. Evaluate the fitness of each individual in that population
3. Repeat on this generation until termination: (time limit, sufficient fitness achieved, etc.)
    a. Select the best-fit individuals for reproduction
    b. Breed new individuals through crossover and mutation operations to give birth to offspring
    c. Evaluate the individual fitness of new individuals
    d. Replace least-fit population with new individuals

CHAPTER IV

DEVELOPMENT OF TOOL

CHAPTER 4

DEVELOPMENT OF TOOL

4.1 PROCDURE INVOLVED TO DEVELOP THE TOOL

1. Development of Graphical User interface using TCL/TK
2. Optimization Function
3. Optimization Algorithm -–(Stochastic Programming)
    a) Genetic Algorithm
4. Validation of Tool.

4.2 DEVELOPMENT OF GUI:

4.2.1 TCL/TK

Tcl stands for **Tool Command Language. Tcl** and **Tk**, its associated graphical user interface toolkit, was created by Professor John Ousterhout of the University of California, Berkeley. Tcl is a scripting language that runs on Windows, UNIX and Macintosh platforms. Tk is a standard add-on to Tcl that provides commands to quickly and easily create user interfaces. Even though Tcl was originally created on UNIX, your Tcl/Tk scripts should run the same on all supported platforms, except for a few differences.

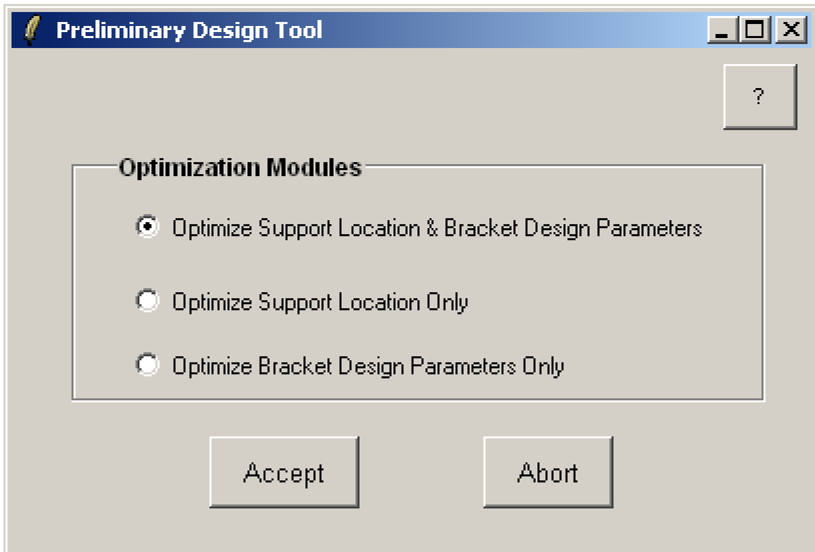
Tcl is a very simple, open-source-licensed programming language and provides basic language features such as variables, procedures, and control, and it runs on almost any modern OS, such as Unix, Macintosh, and Windows 95/98/NT/XP computers.

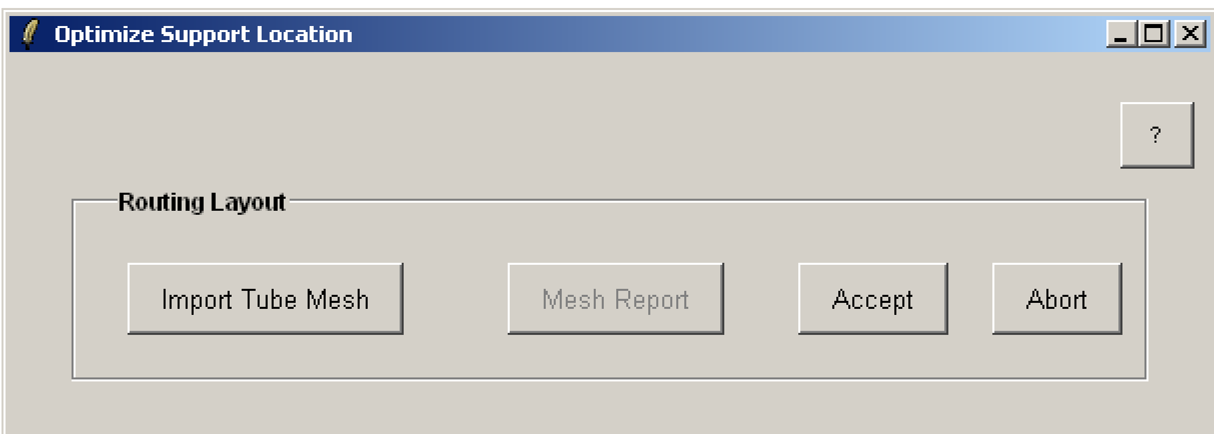
Tcl was originally developed as a reusable command language for experimental computer aided design(CAD) tools. The interpreter was implemented as a C library that could be linked into any application. It is very easy to add new functions to the Tcl interpreter, so it is an ideal reusable "macro language" that can be integrated into many applications.

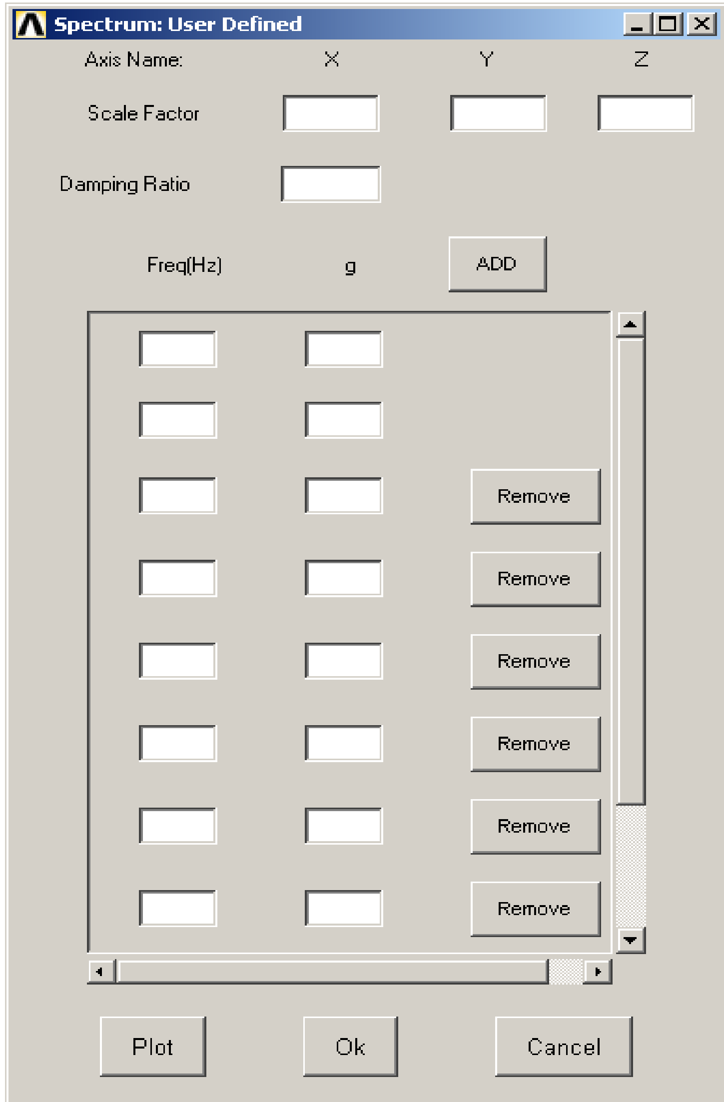
But Tcl is a programming language in its own right, which can be roughly described as a cross-breed between LISP/Scheme and shells (with more powerful structuring). You can write any number of programs in Tcl, just as you can in any other language. Tcl programs are usually called "scripts" because the programs do not need to be compiled into a machine-readable form.

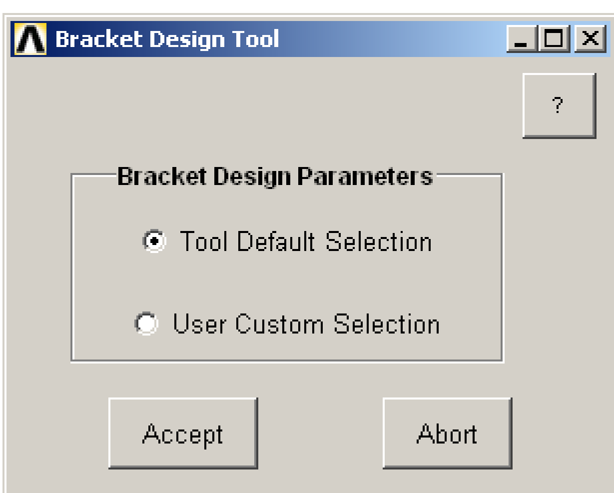
Development of GUI is done using TCL/TK software

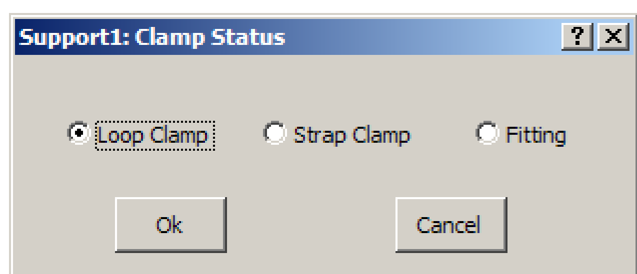
Some GUI windows






Spectrum: User Defined
Axis Name:
X
Y
Z
Scale Factor
Damping Ratio
Freq(Hz)
g
ADD
Remove
Remove
Remove
Remove
Remove
Remove
Plot
Ok
Cancel


Bracket Design Tool
?
Bracket Design Parameters
Tool Default Selection
User Custom Selection
Accept
Abort


Support1: Clamp Status
Loop Clamp
Strap Clamp
Fitting
Ok
Cancel

4.3 OPTIMIZATION FUNCTION

Objective function:

$$\sum_{i=1}^{m}\left(\frac{F_{\text{target1}}-F_i}{F_{\text{target2}}}\right)^2+\sum_{i=n}^{N}\left(\frac{F_{redline}-F_i}{F_{redline}}\right)^2+\prod_{i=m}^{n}1+\left(\frac{F_{redline}-F_o}{F_{redline}}\right)^2+\left(\frac{M_{oper}}{M_{\max}}\right)$$

Where

$F_{redline}$- Redline frequency target

$F_{target1}$- First frequency avoidance

$F_{target2}$- Second frequency avoidance

m-No. of modes within Ftarget1

n-No. of modes within Ftarget12

N- No. of modes within redline frequency

F_0- First frequency above redline frequency

M_{oper}- No. of modes within redline freq

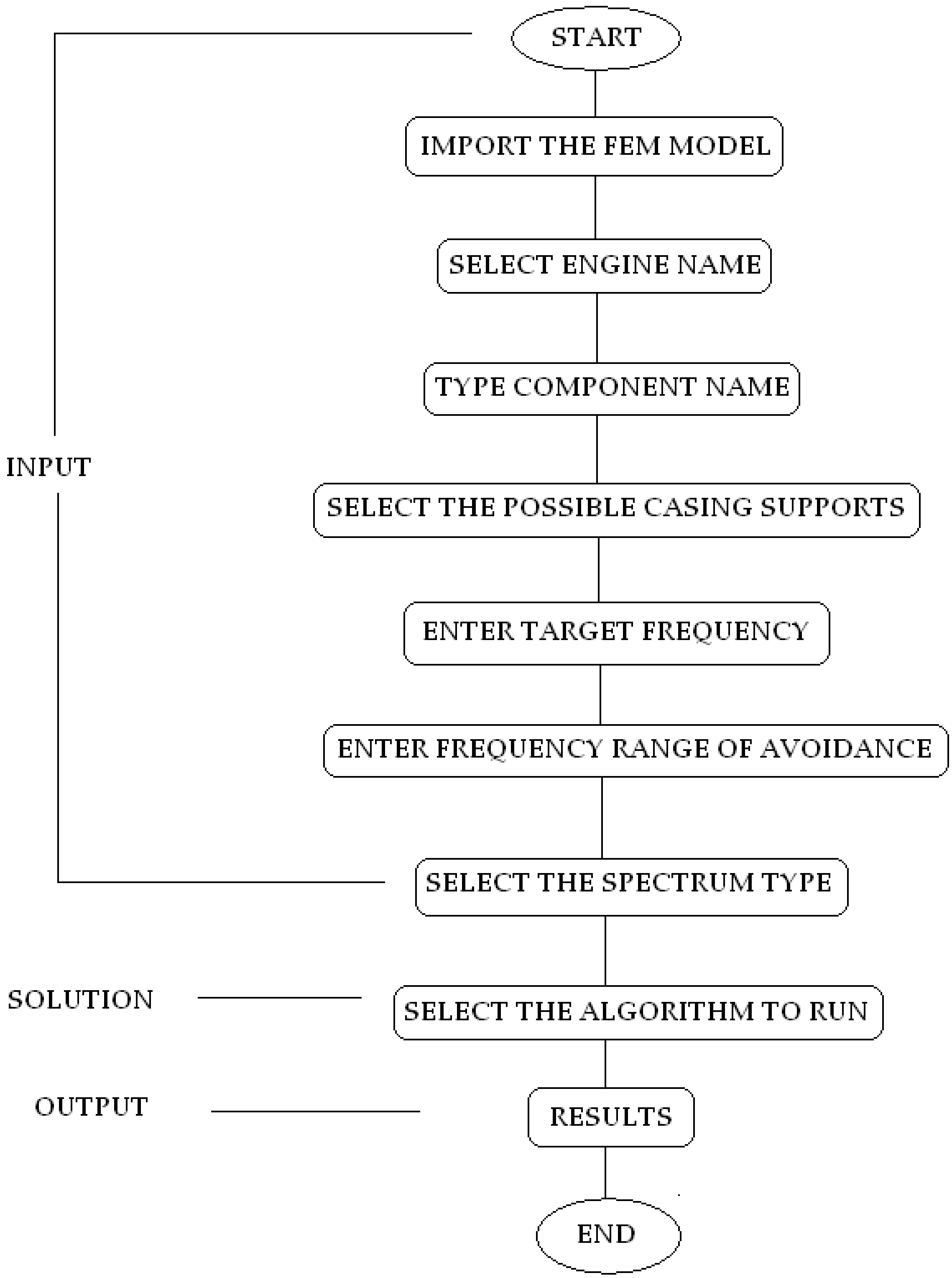
M_{max}- No. of modes within redline freq (with no supports)

Minimizing the above Objective function will give an optimum results.

4.4 OPTIMIZATION ALGORITHM

Genetic algorithm is used , which is readily available with Pez tool. Genetic Algorithm is a Stochastic Programming which gives an better results.

4.5 WORK FLOW OF THE TOOL



CHAPTER V

CASE STUDY

CHAPTER 5

CASE STUDY

5.1 SUMP VENT TUBE SYSTEM

It is a tube system which is used to carry an fluid from one part to other part of the engine. It consists of tubes, brackets and having various possibilites of intermediate supports. Figure 5.1 shows the model of Sump vent tube.

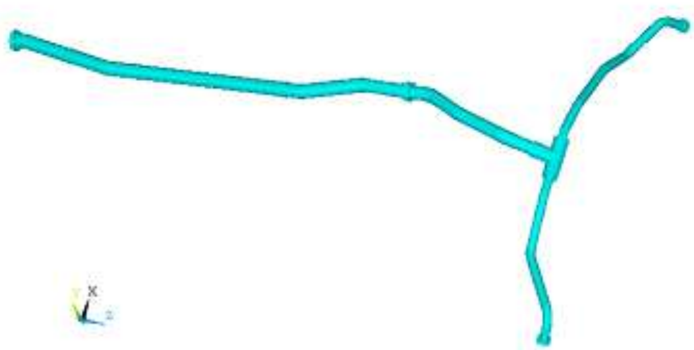


Fig. 5.1: Sump Vent Tube System

Figure 5.2 shows the various possibilities of casing support point and the tube support point and along with the families of tube support points. Families of tube support points are identified based on the availability of casing support points.

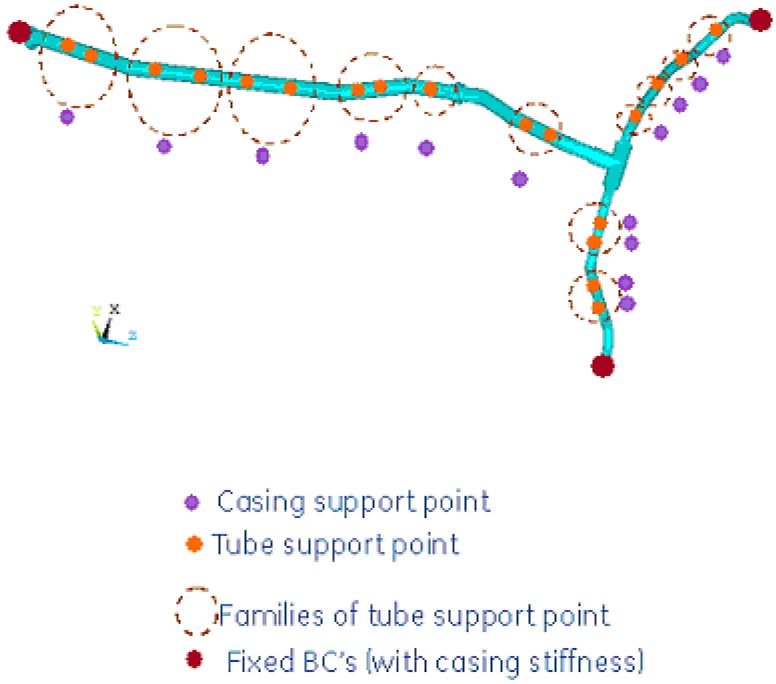


Fig. 5.2: Sump Vent Tube System

Then the tool is allowed to run for an optimization algorithm. From the available of so many results best results is taken. Figure 5.3 shows the outputs from our given objective function having four intermediate support locations having first frequency as 256.2 Hz. It is a preliminary Design output from this a detailed analysis will be carried out.

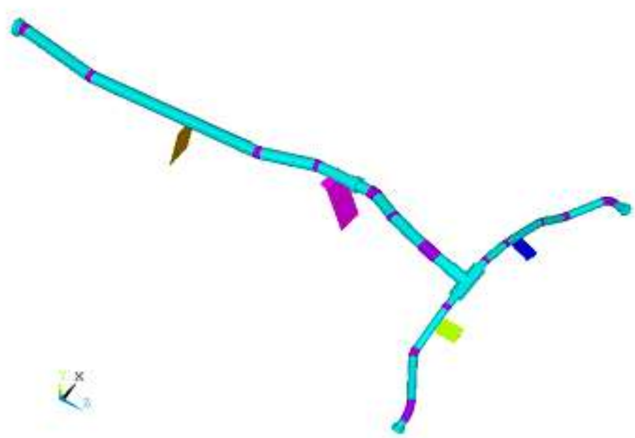


Fig. 5.3: Output: Having four intermediate supports

5.2 FUEL MANIFOLD (PICCOLO TUBE SYSTEM)

It is a tube system which is used to carry an FUEL from one part to other part of the engine. It consists of tubes, more number of brackets. Figure 5.4 shows the model of fuel Manifold.

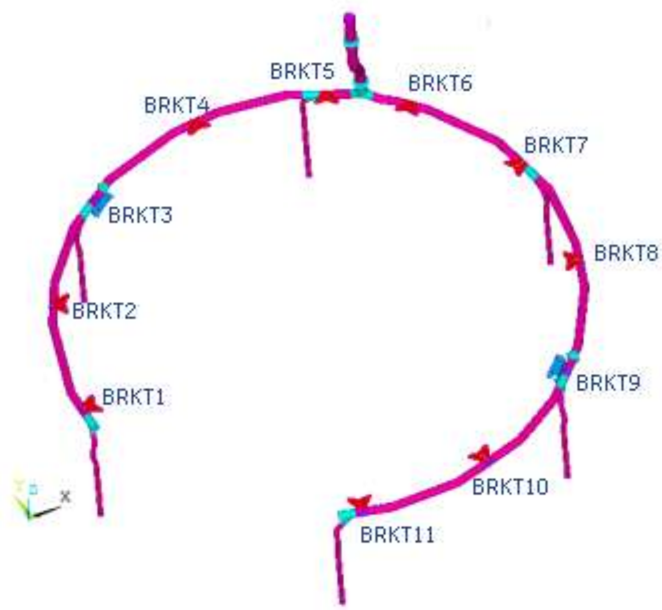


Fig. 5.4: Fuel Manifold System

System is considered as a baseline model of having eleven bracket combinations with a first frequency of 198.1 Hz. Then the tool is allowed to run for an optimization. From the available of so many results best results is taken. Figure 5.5 shows the outputs of the best one having nine bracket locations having first frequency as 261.2 Hz. After this a detailed analysis will be carried out.

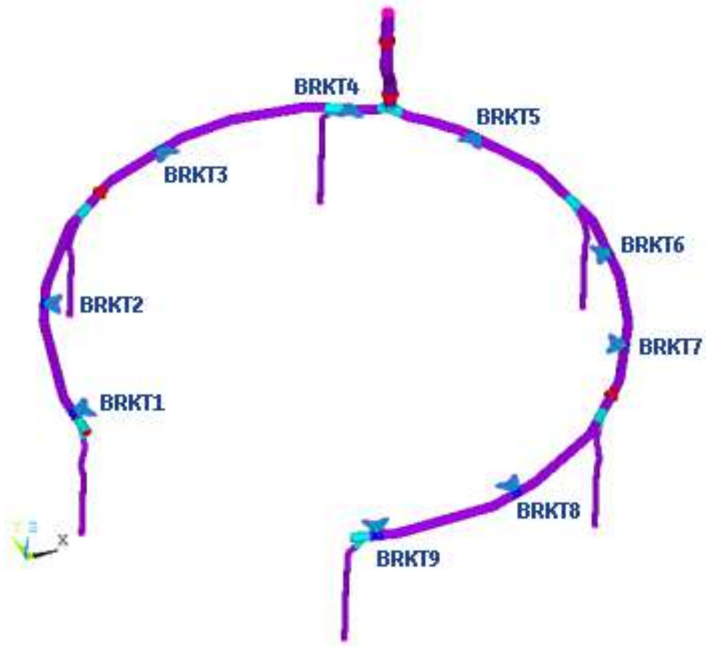


Fig. 5.4: Output: Having Nine Bracket Combinations

CHAPTER VI

SUMMARY

AND CONCLUSIONS

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 GENERAL

In this Chapter, a summary of the present work and the recommendation for future scope of work are presented. The present work gives optimum output when compared with the existing baseline model and tool is developed as a user-friendly computer programs for whole workflow and the output will be viewed through ANSYS.

6.2 CONCLUSIONS

In conventional method it takes number of days to analyse to find the optimum results and more over designers knowledge is required. Where as the present study will gives the optimum results in one day based on our objective function that reduces engineers work time.

6.3 SCOPE FOR FUTURE WORK

- At present the optimization objective function is based only on the frequency criteria to achieve better results Stress criteria, Cost criteria may be included.
- Algorithm used here was Genetic Algorithm, which takes more running time to satisfy the convergence criteria. Different Algorithm may be used.

REFERENCES

REFERENCES

1. **Jaochim Stender, Brain ware GMBH, Berlin, Brainware Limited**, London *" Introduction to Genetic Algorithm"*

2. **Kalyanmoy Deb***,"An efficient constraint handling method for genetic algorithms"* Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering, Indian Institute of Technology Kanpur, kanpur 208 016, India

3. **J. M. Cohen, J. R. Hibshman, W. Proscia, T. Rosfjord, B.E. Wake** *"Longitudinal mode Aeroengine combustion instability model and experiment",* USA 2002.

4. **John Dominy** *"Structural composites in civil gas turbine aero engines"*, Composites Manufacturing, Volume 5, Issue 2, June 1994,

5. **John D. Black and Mark P. Johnson** "*In-situ laser-induced incandescence of soot in an aero-engine exhaust: Comparison with certification style measurements*" Aerospace Science and Technology, Volume 14, Issue 5, July-August 2010.

6. **O. D. Lyantsev, T. V. Breikin, G. G. Kulikov, V. Y. Arkov** "*On-line performance optimisation of aero engine control system*" Automatica, Volume 39, Issue 12, December 2003.

7. **M. Naeem, R. Singh, D. Probert** ,"*Consequences of aero-engine deteriorations for military aircraft*" Applied Energy, Volume 70, Issue 2, October 2001.

8. **Lucjan Witek** ,"*Failure analysis of turbine disc of an aero engine* " Engineering Failure Analysis, Volume 13, Issue 1, January 2006.

9. **Fengjun Lv, Quan Li, Guoru Fu** ,"*Failure analysis of an aero-engine combustor liner* " Engineering Failure Analysis, Volume 17, Issue 5, July 2010

10. **F. Haglind, B. Elmegaard**.,"*Methodologies for predicting the part-load performance of aero-derivative gas turbines*" Energy, Volume 34, Issue 10, October 2009.

11. **M.M.I. Hammouda, R.A. Pasha, A.S. Fayed** ,"*Modelling of cracking sites/development in axial dovetail joints of aero-engine compressor discs*" International Journal of Fatigue, Volume 29, Issue 1, January 2007.

12. **T. V. Breikin, V. Y. Arkov, G. G. Kulikov** ,"*Regularisation approach for real-time modelling of aero gas turbines*" Control Engineering Practice, Volume 12, Issue 4, April 2004.