



Centrale
Nantes

Second year

ECOLE CENTRALE DE NANTES



UNIVERSITÀ DEGLI STUDI
DI GENOVA

First year

UNIVERSITY OF GENOVA

MASTER ERASMUS MUNDUS

EMARO “EUROPEAN MASTERS IN ADVANCED ROBOTICS”

2014-2015

Master’s Thesis

Presented by

ABHILASH UDAY NAYAK

on 20/07/2015

OPTIMAL AND ROBUST DESIGN OF A WOODEN LINK OF AN
INDUSTRIAL ROBOT

Jury

President:

Philippe MARTINET, Professor

Evaluators:

Sébastien BRIOT, CNRS Researcher

Stéphane CARO, CNRS Researcher

Mark IRLE, Research Director

Supervisor:

Sébastien BRIOT

Laboratory: Institut de Recherche en Communications et
Cybernétique de Nantes

Contents

List of Figures	3
List of Tables	4
1 Introduction	5
1.1 Objective	6
1.2 Dissertation plan	6
2 Literature review	7
2.1 Wood properties	7
2.1.1 Advantages	7
2.1.2 Drawbacks of wood	9
2.1.3 Choice of wood	9
2.1.4 Conclusion	10
2.2 Optimal Design	11
2.2.1 Optimal design of a link	11
2.2.2 Topology optimization	12
2.3 Sensitivity analysis and Robust design	16
2.4 Optimization Methodologies	18
2.4.1 Introduction	18
2.4.2 Categories of optimization problems	19
2.4.3 Algorithms for finding the solution	21
2.5 Steps to be carried out	24
3 FEM	25
3.1 Finite Element Formulation	25
3.1.1 Selecting the element type and finding the interpolation functions	26
3.1.2 Defining the strain-displacement relationships	29
3.1.3 Stress strain relationships for a wooden element	29
3.1.4 Deriving the elemental stiffness matrix	32
3.1.5 Finding the global stiffness matrix and finding the unknown dis- placements	34
3.2 Stiffness modelling in ANSYS workbench	34
3.3 Conclusion	36
4 Shape optimization of a wooden robot link	37
4.1 Design domain	37
4.2 Optimization problem	38
4.3 The BESO procedure adopted and flowchart	39

4.4	Sensitivity analysis	42
4.5	Filter scheme for BESO method	43
4.6	Stabilizing the Evolutionary Process	44
4.7	Determination of the structural volume to satisfy the constraints	45
4.8	Element Removal/Addition and Convergence Criterion	46
4.9	Implementation	47
4.10	Validation of deformation for the link with two holes	48
4.11	Robust design	49
4.12	Conclusion	50
5	Conclusions and future work	51
	Appendices	53
A	MATLAB code for shape optimization of the wooden link	54
6	Bibliography	60

List of Figures

1.1	CAD modeling of the IRSBot2 with a wooden link [1].	6
2.1	Comparison of mechanical properties of several materials [2].	9
2.2	Material Selection Map [3].	10
2.3	Sensor based control [4].	11
2.4	Homogenization approach to topology optimization [5].	13
2.5	Robust vs optimal design [6]	17
2.6	flowchart of post-optimality sensitivity analysis [7]	18
3.1	4 node square element [8].	26
3.2	Transition to natural coordinates [8].	28
3.3	8 noded cubic element in cartesian and natural co-ordinates with origin at the centroid [8].	28
3.4	Components of stress in three dimensions [9].	30
3.5	Wood modelled as an orthotropic material [10].	30
3.6	Cantilever beam.	35
3.7	FE analysis of a cantilever beam of yellow poplar in ANSYS	35
4.1	Initial design environment to find an optimum topology	38
4.2	Calculating mass moment of inertia of each finite element	39
4.3	Flowchart for the shape optimization of the wooden link.	41
4.4	Nodes located inside the circular sub-domain Ω_i are used in the filter scheme for the i th element [11].	43
4.5	Design domain of a short cantilever [11].	45
4.6	Comparison of evolution histories [11]	45
4.7	Optimal design for the two cases based on direction of force applied	48
4.8	FE analysis of the optimized designs in figure 4.7 in ANSYS	49
4.9	Variation of mechanical properties of several materials [2].	49

List of Tables

2.1	Net carbon emissions in producing a tonne of various materials [12]. . . .	8
2.2	Comparison between different materials for the construction of a square section beam of 2m supporting a load of 10 t/m [2].	8
3.1	List of elastic constants for some wood species [13].	31
3.2	Poisson's ratios for various species at approximately 12% moisture content [12].	32
3.3	Deformation of the cantilever beam.	36
4.1	Deformation of the optimized topology of a cantilever beam.	49

Chapter 1

Introduction

The quest to build better robots capable of completing more tasks is an important one, that could lead to humanity as a whole having a much easier time of it. Recently, the use of parallel robots in various fields has become apparent especially in precise manufacturing, medical sciences and space exploration equipments [14, 15]. They are intrinsically stiffer and faster than serial robots. An important merit of parallel robots is that there are hundreds of possible architectures, each with very specific advantages and disadvantages. Some parallel robots are perfect for machining (they are called Parallel Kinematic Machines or PKMs) or for motion simulation, because they are rigid. Others are great for pick-and-place operations, because their mobile part is light. Cable robots, for example, are ideal for covering a very large work area, such as a stadium.

However, one aspect that few people seem to be considering as we approach a time when robots are a mainstream part of our lives is the environmental impact such a scenario may have. At the moment, robots are primarily made from metals and plastics, both of which offer rigidity and a long life. In fact, these materials are generally toxic and non-biodegradable. Furthermore, any robot must be tracked, and once it reaches the end of its usable life, it should be dismantled, recovered and made safe. This adds complexity to the robotic projects and there is an ever present risk that the robot is irrecoverable with consequent damage to the environment. Additionally, these characteristics limit the number of robots that can be employed since each must be tracked and recovered. Hence, the overall impact on the environment of an army of robots doing our menial tasks could be massively harmful – unless an alternative is sought. Constructing robots from materials with low environmental impact, however, would mean they eventually cease to be, merely decomposing into the earth. Wood, with environment-friendly properties like low embodied energy, low carbon impact and sustainability could be used to design the robots. The replacement of a carbon link with a wooden link in the design of a robot arm presented by Laurent et al [2] proves that it is possible to reconcile the parallel robot architecture and the wood elements to reduce the environmental impact of these machines.

In the design of parallel robots, it is important to have the links with the least moment of inertia about the joints which helps reduce the power required by a motor attached at those joints to run these robots. Also, it is necessary to keep a check on the deformation of the links due to the loads applied on it.

1.1 Objective

IRSBot-2 [1] developed in IRCCyN is a two degree-of-freedom translational parallel manipulator dedicated to fast and accurate pick-and-place operations. The objective of this thesis is to replace the IRSBot-2 links by wood.

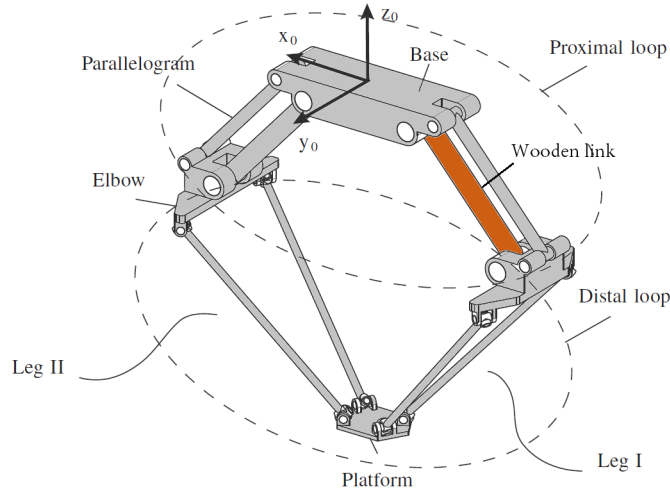


Figure 1.1: CAD modeling of the IRSBot2 with a wooden link [1].

The design of a wooden IRSBot robot can be classified into two main possible research areas that can be conducted sequentially :

1. To perform the optimal and a robust design of a wooden link of the industrial robot shown in Figure 1.1. Structural topology optimization can be adopted in this case.
2. Optimal design of the whole manipulator in regard to its kinematics, dynamics, kinetostatics and elastodynamics.

Since the topic is very wide, only the first part is considered as the goal to be reached. Hence the objective is to come up with a topology for the wooden link of IRSBot-2 having least moment of inertia about the actuated joint with a constraint on deformation. This design should also be robust (least sensitive to the change of properties of the wood considered).

1.2 Dissertation plan

Introduction being the first chapter, the second chapter is the literature review on the properties of wood, optimal and robust design in general, techniques for topology optimization and the optimization methodologies that could be used to solve the problem at hand. The third chapter sheds light into the method of Finite Element Analysis which is a major part of the topology optimization. The fourth chapter explains the theory behind the shape optimization of a wooden robot link, how it is implemented in MATLAB and the validation of results in ANSYS. The MATLAB code written to optimize the moment of inertia with constraints on displacement is given in Appendix A.

Chapter 2

Literature review

2.1 Wood properties

2.1.1 Advantages

Wood is experiencing a resurgence of interest due to its environmental qualities and improving knowledge on its scientific properties by scientific research. One of the greatest attributes of wood is that it is a renewable resource. If sustainable forest management and harvesting practices are followed, our wood resource will be available indefinitely. Wood has many positive characteristics [12]:

- **Low embodied energy** : The quantity of energy required to harvest, mine, manufacture, and transport to the point of use a material or product. Wood, a material that requires a minimal amount of energy-based processing, has a low level of embodied energy relative to many other materials such as steel, concrete, aluminum, or plastic.
- **Low carbon impact** : The carbon emitted while producing other materials is way more than that of wood.

As indicated in Table 2.1 [12], carbon emitted to produce a tonne of concrete is about eight times that emitted to produce a tonne of framing lumber. A similar comparison for steel indicates that its production emits about 21 times as much carbon as an equal weight of framing lumber. Wood products also mitigate carbon emissions to the degree that they substitute for steel or concrete, which emit more greenhouse gases in their production.

- **Sustainability** : Unlike metals and fossil-fuel-based products (such as plastics), our forest resource is renewable and with proper management a flow of wood products can be maintained indefinitely. The **forest certification programs** not only ensure that the forest resource is harvested in a sustainable fashion but also that issues of biodiversity, habitat protection, and indigenous peoples' rights are included in land management plans.

Thierry et al. [2] highlight the environmental benefits of wood with statistics. Compared to different kind of materials including metals, alloys, polymers, concrete, glass, glulam, carbon footprint of the wood is almost neutral. In their study, it is evident that wood stands out in properties like energy required for production, CO_2 emission and the fraction recyclable. To make a good comparison, the energy consumption and CO_2 emission

Material	Net carbon emission in kg C/t
Framing lumber	33
Medium-density fiberboard (virgin fiber)	60
Brick	88
Glass	154
Recycled steel (100% from scrap)	220
Concrete	265
Recycled aluminum (100% recycled content)	309
Steel	694
Plastic	2,502
Aluminum	4,532

Table 2.1: Net carbon emissions in producing a tonne of various materials [12].

are compared for the same application. Table 2.2 [2] shows the properties of different materials for construction of a beam of 2m considered to endure a load of 10 tonnes per meter.

Material	Mass volume (kg/m^3)	Elastic limit (MPa)	Height (cm)	Mass (kg)	Energy consumed (MJ)	Emission CO_2 (kg)
Composite Car- bon	1500	200	11	36	9900	730
Alloy aluminium	2700	150	13	85	13900	890
Steel	7800	300	10	154	6500	300
Synthetic Poly- mer	950	25	23	98	6600	140
Glass	2200	25	23	228	4600	230
Concrete	2500	10	31	476	950	80
Glulam against plywood	750	35	20	62	560	-20
Hardwood (spruce)	500	45	19	35	90	-70

Table 2.2: Comparison between different materials for the construction of a square section beam of 2m supporting a load of 10 t/m [2].

Also, the characteristics of some important materials for a given treatment are plotted as shown in Figure 2.1 [2]:

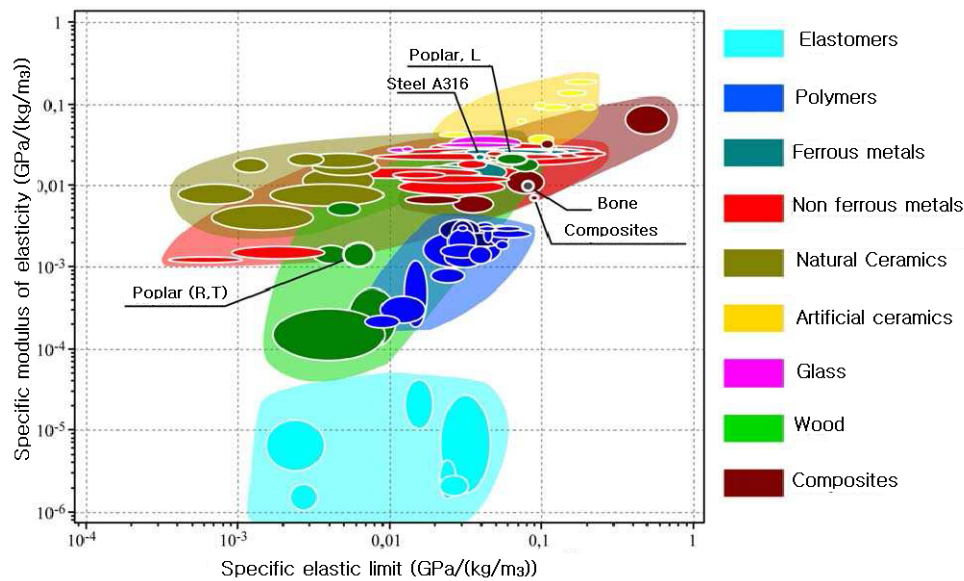


Figure 2.1: Comparison of mechanical properties of several materials [2].

Some conclusions that could be drawn from the picture are [2]: The specific stiffness of the poplar wood in the longitudinal direction is quite comparable to that of steel whereas in the transverse direction, it is 10 times lower than that of steel. Compared to other biological materials, wood has comparable hardness to that of bone. Compared to composite materials, the specific properties of wood are generally lower.

2.1.2 Drawbacks of wood

Wood is an organic product—a structure of infinite variation of detail and design. It is on this account that no two woods are alike: in reality no two specimens from the same log are identical. The important factors that affect the properties are listed below: [12]

- Rate of growth.
- Specific gravity.
- Common defects like cross grains, knots, frost splits, shakes, galls and pitch pockets.
- Season of cutting.
- Water content.
- Temperature.

2.1.3 Choice of wood

There are umpteen varieties of wood in nature. The search for a wood that could satisfy the needs of this project was based on the geographical location (trees in and around France), their applications and material properties.

Raw wood materials like Poplar (peuplier), Oak (Chêne), Beech (Hêtre), Pine (Pin) are the first choice since they are easily available and cost less.

Accoya [16], the acetylated wood has many advantages over raw wood species in terms of durability, strength to weight ratio, machinability, dimensional stability and other properties.

Also, **composite wood materials** [12] created as a mixture of wood, thermoplastics and a bonding material have some advantages over raw wood materials. Panel products, structural timber products or wood-nonwood composites can be some of them.

Thierry et al [3] compare the mechanical properties of different varieties of wood with other composites and alloys in their project aimed at the eco-design of a high speed robot arm made of wood. They choose the material from the selection map shown in figure 2.2 in which three equi-performance lines are shown. These lines correspond to a performance index of the material M which is proportional to the shear strain per unit volume of the material. Higher the M , higher is the stiffness and hence better performance. It is found that the wood used in the L (longitudinal) direction (parallel to the fibers) are more efficient than steels and equivalent to the best alloys of aluminum and titanium. The behaviour of wood is also close to the behaviour of carbon composites in certain fiber directions of the latter.

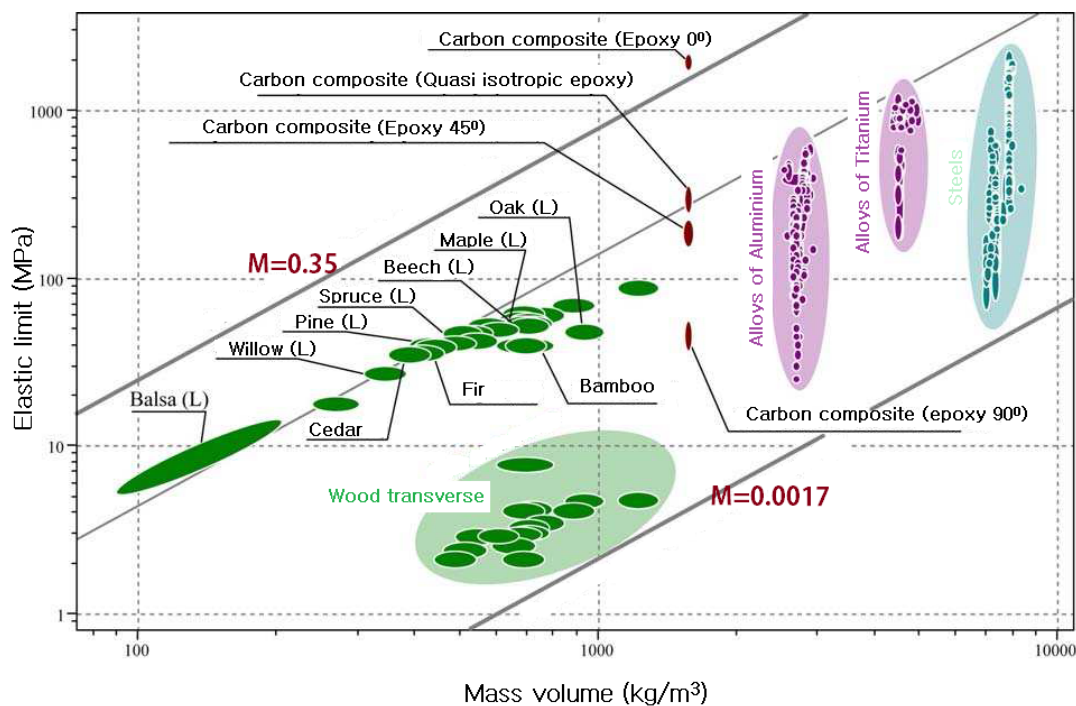


Figure 2.2: Material Selection Map [3].

In this thesis, the type of wood is not chosen in the beginning since the focus was more on the design of the link. Nonetheless, the properties of the wood are the inputs to the design which could be changed according to the type of wood chosen at the end.

2.1.4 Conclusion

Wood is a renewable source of energy with low embodied energy, low carbon impact and sustainability. But it also has some drawbacks like change of properties with time due to the environmental changes in temperature and humidity. Also, the properties of

wood differ from one species to another. Wood can be modelled as an orthotropic material since it has three axes of symmetry-longitudinal, radial and axial. Use of raw wood in the design is easier and costs the least. But it has many disadvantages. The choice of wood with better properties would be to use composite wood materials or accoya. Nonetheless, the problems in variation of properties is inevitable. Therefore, two approaches can be applied to tackle this problem:

- **Optimal robust design** : The objective of optimal robust design is to find the optimal design parameters based on the defined constraints and minimize the variability that results from uncertainty due to variation in the parameters measured. The approaches of optimal robust design are described in the subsequent chapters.
- **Sensor based control** : Exteroceptive sensors measure the current position of the end-effector of the robot and keep updating the error between the current position and the desired one [4]. In figure 2.3, s represents the current measure, s^* , the desired measure and e , the error:

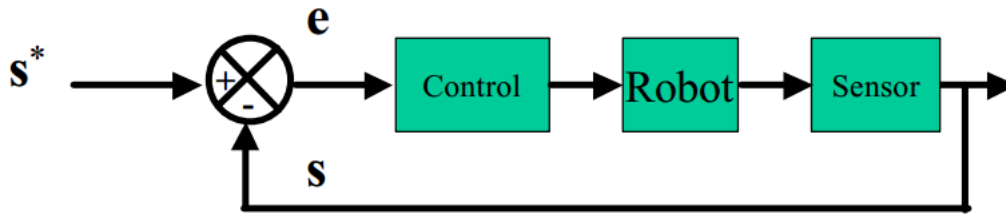


Figure 2.3: Sensor based control [4].

For example if s is the cartesian position of the end effector, the error in the cartesian space, δx can be translated to the error in the joint space δq using the Jacobian, J of the robot ($\delta x = J\delta q$) and hence the control can be performed. Similarly, kinematic control involves velocities and dynamic control involves torques.

2.2 Optimal Design

2.2.1 Optimal design of a link

The optimal design of a link can be considered as a shape optimization problem since we are looking for the best shape of the link satisfying the required properties. One of the options is to come up with some designs by trial and error and to choose one of them that behaves the closest to the metal link that is to be replaced [2, 3].

However, it will not cover the whole range of designs. Hence, the smart approach would be to choose a design through optimization. A problem of shape optimization can be defined by three ingredients : [17, 18]

- **A model** (typically a partial differential equation) to evaluate (or analyse) the mechanical behavior of a structure.
- **An objective function** which has to be minimized or sometimes several objectives (also called cost functions or criteria).

- A **set of admissible designs** which precisely defines the optimization variables, including possible constraints.

Optimal design problems can roughly be classified in three categories from the “easiest” ones to the “most difficult” ones: They are described briefly in the subsequent sections [18].

- **Parametric or sizing optimization** for which designs are parametrized by a few variables (for example, thickness or member sizes), implying that the set of admissible designs is considerably simplified.
- **Geometric or shape optimization** for which all designs are obtained from an initial guess by moving its boundary (without changing its topology, i.e., its number of holes in 2-d).
- **Topology optimization** where both the shape and the topology of the admissible designs can vary without any explicit or implicit restrictions. The following section gives a detailed literature review on the existing topology optimization problems and the algorithms to solve them.

2.2.2 Topology optimization

Structural topology optimization is a mathematical process of finding the optimal layout of prescribed amount of material within a given domain, with the aim of optimizing desired performance objectives.

Most of these methods are based on finite element analysis (FEA) where the design domain is discretized into a fine mesh of elements. In such a setting, the optimization procedure is to find the topology of a structure by determining for every point in the design domain if there should be material (solid element) or not (void element) [11].

Material interpolation used in topology optimization mainly includes the following two types of methods: the **homogenization method** [19, 5], primarily used in theory derivations, such as the existence of solutions, investigating numerical instabilities, etc.; and the material density method, sometimes named the **solid isotropic material with penalization model (SIMP) method**, proposed by Mlejnek and Schirmmacher [20], Sigmund [21], and Bendsoe and Sigmund [22]. Then there is **Bidirectional Evolutionary Structural Optimization (BESO)** [11] which takes advantage of powerful computing technology and intuitive concepts of evolution processes in nature. Advantages and disadvantages of each method are given below.

Homogenization approach

In the original paper by Bendsoe and Kikuchi(1988) [5], the topology optimization problem is tackled by considering two material constituents: substance and void. In that context, the general problem formulation was given as follows [5]:

Consider a mechanical element as a body occupying a domain Ω in R^3 and suppose that the body is subject to body forces f and boundary tractions t on the boundary Γ_T . The fixed or constrained boundaries are denoted as Γ_D (Refer Figure 2.4). In optimal design for minimum compliance we seek the optimal choice of elasticity tensor E_{ijkl} in some given set of admissible elasticity tensors, U_{ad} .

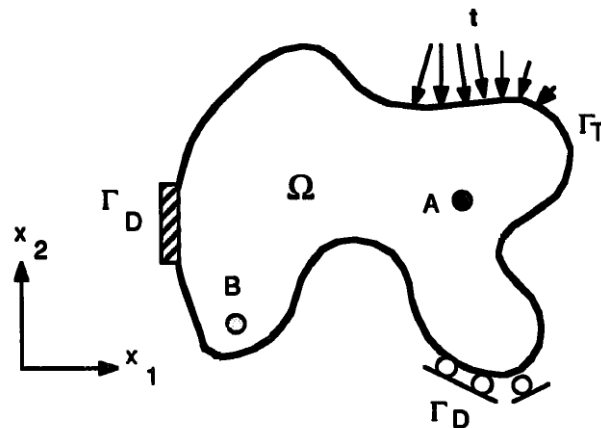


Figure 2.4: Homogenization approach to topology optimization [5].

This method can also be applied for composite materials. It mainly consists of the following steps:

1. Choose a suitable reference domain that allows you to define surface tractions, fixed boundaries, etc.
2. Choose a composite, constructed by period repetition of a unit cell consisting of the given material with one or more holes.
3. Compute the effective material properties of the composite, using homogenization theory. This gives a functional relationship between the density of material in the composite (i.e. sizes of holes) and the effective material properties.
4. Compute the optimal distribution of this composite material in the reference domain, treating the problem as a sizing problem with the density as the sizing variable.
5. Interpret the optimal distribution of material as defining a shape, in the sense of the general shape design formulation given above.

Advantages :

It can be used for any types of materials be it isotropic, anisotropic or even some porous materials.

Disadvantages :

1. Quite complex to understand and apply.
2. Possibilities of checkerboard patterns and mesh-dependencies [23].

SIMP

The presently most popular numerical FE-based topology optimization method for isotropic materials is the SIMP method, which was developed in the late eighties. It is sometimes called “material interpolation”, “artificial material”, “power law”, or “density” method,

but ‘‘SIMP’’ is now used fairly universally [24]. The term ‘‘SIMP’’ stands for **Solid Isotropic Microstructure (or Material) with Penalization** for intermediate densities. The basic idea of this approach was proposed by Bendsoe [25], while the term ‘‘SIMP’’ was coined later by the author and first introduced in a paper by Rozvany et al. [26].

As already explained, topology optimization involves rendering a final design like a black and white image with values 0-1.

Because of the size of this discrete value (0–1) problem, SIMP uses a continuous variable formulation since direct search methods would be prohibitively expensive. Here, material properties are assumed constant within each element used to discretize the design domain and the variables are the element relative densities. The material properties are modelled as the relative material density raised to some power times the material properties of solid material. The **penalization power** assists the optimization to attain a 0-1 topology.

A topology optimization problem based on the power law approach (SIMP), where the objective is to minimize compliance can be written as [27] :

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c(x) = U^T K U = \sum_{e=1}^N (x_e)^p u_e^T k_0 u_e \\ \text{subject to} \quad & \frac{V(x)}{V_0} = f \\ & K U = F \\ & 0 \leq x_{min} \leq x \leq 1 \end{aligned} \tag{2.1}$$

where U and F are the global displacement and force vectors, respectively, K is the global stiffness matrix, u_e and k_e are the element displacement vector and stiffness matrix, respectively, x is the vector of design variables, x_{min} is a vector of minimum relative densities (non-zero to avoid singularity), N is the number of elements used to discretize the design domain, p is the penalization power, $V(x)$ and V_0 is the material volume and design domain volume, respectively and f is the prescribed volume fraction. More details on these parameters can be found in [27].

Solution algorithms

The optimization problem (2.1) could be solved using several different approaches such as Optimality Criteria (OC) methods [27], Sequential Linear Programming (SLP) methods or the Method of Moving Asymptotes [28] and others. These methods are briefly outlined as follows :

Optimality criteria (OC) method

Following Bendsoe [29], in OC, a heuristic update scheme for the design variables is formulated as given in [27]. Also a filtering scheme is used and sensitivity analysis is the method used to obtain an optimum topology [27].

MMA (Method of Moving Asymptotes)

As Svanberg [28] argues, the ideal method for structural optimization should be flexible, general, and able to handle not only element size as design variables, but also other variables such as shape and material orientation angles. It should be able to handle all kinds of constraints. MMA method can handle all of these problems in addition to general non-linear programming problems. Also, it is easy to implement and use. The method of moving asymptotes is based on a special type of convex approximation.

Though MMA is quite a popular method for topology optimization, recent investigations have shown poor performance of the MMA algorithm as compared to other approximations. Hence a two-point gradient based MMA approximation, termed as TG-MMA, by Li and Kapil [30] can be adopted to improve the performance of the MMA algorithm.

Advantages of SIMP :

1. SIMP solves for a continuous minimiser which has low compliance for a specified volume and produces a smooth solution.
2. SIMP is suitable for a combination of a wide range of design constraints, multiple load conditions, multi-physics problems, and extremely large (often 3D) systems [24].

Disadvantages of SIMP :

1. Since the solution algorithms involve mathematical programming techniques, the search region is wide and computational time is quite high.
2. Grey areas appear in the final decision and a post-optimality decision is to be made to keep or remove these grey regions.
3. Has many continuation paths leading to different local solutions [24].

BESO

Bidirectional Evolutionary Structural Optimization (BESO) is an intuitive method inspired by nature. By observing the evolution of naturally occurring structures such as shells, bones and trees it becomes obvious that the topology and shape of such structures achieve their optimum over a long evolutionary period and adapt to whatever environment they find themselves in. It demonstrates the possibilities of achieving similar structural shape and layout optimization by using the finite element analysis and training the software to follow a particular evolutionary path [31, 32, 33]. It is a easy-to understand method and works quite well in comparison with the standard SIMP technique.

Advantages of BESO :

1. Since the problem is binary, the search area is reduced considerably and hence an optimum can be found easily.
2. Easy to understand and implement.
3. Gives just black and white regions.
4. It usually reaches a global optimum compared to homogenization method or SIMP [24].

Disadvantages of BESO :

1. Sometimes it is non-convergent and hence a stabilization method is to be used.
2. ESO breaks down if the sensitivity with respect to element density changes rapidly over finite density variations.

Other algorithms

The topology optimization methods can be divided into two methods :

Node-based methods

These methods fill the design space with solid elements and derive bulk topologies. Other than the most used algorithms OC and MMA, there are some standard mathe-

mathematical programming methods such as the **Sequential Linear Programming (SLP)** algorithm (Gomes and Senne [34], Sigmund [35], Kikuchi et al. [36]) and the **Sequential Quadratic Programming (SQP)** algorithm (Etman et al. [37]). Even a hybrid scheme that uses quadratic approximations of some separable convex approximation functions (Groenwold and Etman [38]) has been successfully employed to solve such problems. **Hybrid Cellular Automata (HCA)** has been adopted for synthesis of topologies by Patel and Mozumder [39]. It uses a regular grid of cells to cover the design space and update rules to modify the density of these cells depending on their internal energy density (elastic and plastic deformation) and therefore derive a topology.

Graph-based methods

These methods derive thin-walled structures exploring only a subset of possible topologies: **Ground Structure Approach (GSA)** [40] consists in filling the space with elementary macro-elements with simplified crash behaviour and using different methods to remove and/or modify these macro elements to reach an optimum design. **Graph based method** [41] is based on an abstract graph representation of the structure, usually for 2D design spaces.

There are many methods in the literature to solve topology optimization problem. Analysis of these methods makes it clear that BESO has many advantages and some disadvantages that can be overcome by slight modifications in the code. For topology optimization of the robotic link, BESO is considered and more details are given in chapter 4 about how the method is derived, adopted and programmed.

2.3 Sensitivity analysis and Robust design

Robust product design is a concept from the teachings of Dr. Genichi Taguchi, a Japanese quality guru [42]. It is defined as reducing variation in a product without eliminating the causes of the variation. In other words, making the product or process insensitive to variation. This variation (sometimes called noise) can come from a variety of factors and can be classified into three main types: internal variation, external variation, and unit to unit variation. Internal variation is due to deterioration such as the wear of a machine, and aging of materials. External variation is from factor relating to environmental conditions such as temperature, humidity and dust. Unit to Unit variation is variations between parts due to variations in material, processes and equipment.

In the case of optimal design of a wooden robot, the properties of wood suffer from external variations due to the change of properties with respect to time, temperature and humidity. Hence, robust design is a must. Nonetheless, there are chances of internal and unit to unit variations due to dimensional variations, joint clearances, misalignment of actuators, link flexibility, joint elasticity, [43] etc. Hence, robust design approach is adopted to obtain an effective design which is the least tolerant to variations in its parameters.

In the subsequent sections, optimal and robust design are compared and one of the approaches to solve a multi-objective robust optimization problem is explained. But first, it is important to see how robust design differs from optimal design.

Optimal design vs. robust design

Optimal design behaves well for design parameters with minute or no variations. However, multi-objective engineering design problems often have design parameters with uncontrollable variations due to noise or uncertainties [6]. Such variations can affect outcomes significantly, such as the performances of objective functions and/or the feasibility of the Pareto optimal solutions. A robust optimal solution is as good as possible with regard to the objective functions, and it offers the lowest possible sensitivity to variations in design variables and design parameters. In practice, all engineering designs are sensitive to uncertainties that can arise from manufacturing operations, variations in material properties, the operating environment and other reasons. Moreover, non-robust designs can be expensive to produce or to operate and can fail frequently in service.

Figure 2.5 depicts that the optimal solution need not always be robust [6]. The performance function $f(x)$ is minimum when the design variable x is equal to x_{opt} . However, the sensitivity of $f(x)$ to variations in x_{opt} is significant. Indeed, Δf_{opt} , which depicts the range of variations in $f(x)$ for a given range of variations in x around x_{opt} , is large. On the contrary, x_{rob} is a local minimum of function $f(x)$, and the sensitivity of $f(x)$ to variations in x_{rob} is very small. Indeed, Δf_{rob} , which depicts the range of variations in $f(x)$ for a given range of variations in x around x_{rob} , is small. In fact, $\Delta f_{rob} < \Delta f_{opt}$. Accordingly, x_{rob} is a good solution to the single-objective robust optimization problem.

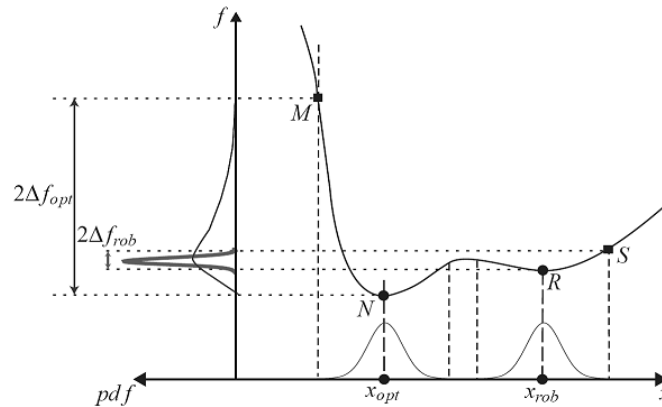


Figure 2.5: Robust vs optimal design [6]

In the case of a multi-objective optimization problem, a robust optimum solution may be located in the neighborhood of the Pareto front. Such a solution should have as little sensitivity as possible to uncertainties, because it cannot violate any constraint and/or acceptable known variations in design objectives in the presence of uncertainties. An effective way to approach to solve a multi-objective robust optimization problem is detailed in the following section before which the advantages and disadvantages of robust design are mentioned.

Advantages of robust design [42] :

1. The effect of robustness on quality is great. Robustness reduces variation in parameters by reducing the effects of uncontrollable variation.
2. There is no need to fix the variation of parameters since it is accounted for in the robust design.

Disadvantages of robust design [42] :

1. To effectively deal with the noise, the designer must be aware of the noise. [6] gives an approach to tackle the problem by splitting the parameters into controllable and uncontrollable ones. Or a sensitivity analysis should be done to check the sensitivity of the manipulator.
2. The problem becomes complex.
3. Design process becomes time consuming.
4. In most of the problems, the noise factors causing variation are considered to be independent. In practice, they might be dependent.

Solving a Multi-Objective Robust Optimization Problem

There are many approaches in the literature to address these kind of problems [6]. One of the efficient approaches is to define a **robustness index** (RI) and to include it as an objective function to be optimized. The flowchart 2.6 illustrates the proposed postoptimality sensitivity analysis technique adopted by Wang et al. [7]:

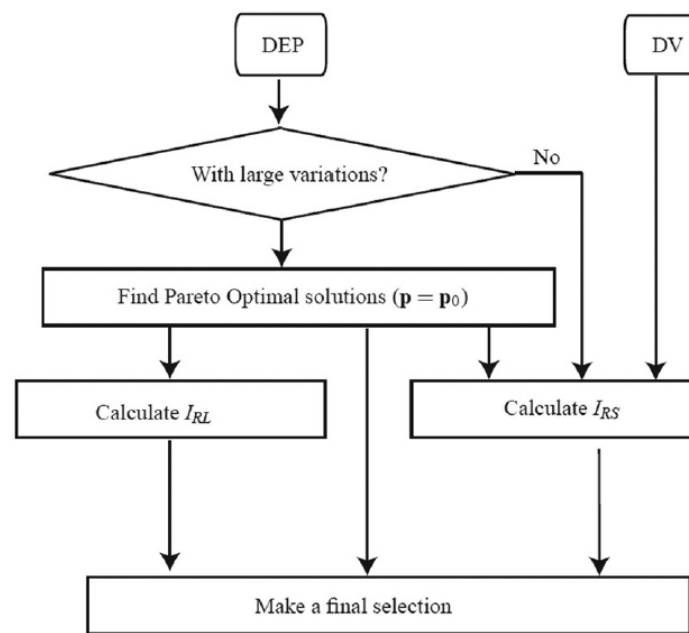


Figure 2.6: flowchart of post-optimality sensitivity analysis [7]

As explained before, the properties of wood vary with moisture, temperature and other environmental factors [12]. Hence **Robust design** approach is to be adopted so that the sensitivity of performances to variations are minimized without controlling the sources of these variations.

2.4 Optimization Methodologies

2.4.1 Introduction

Wikipedia defines mathematical optimization (alternatively, optimization or mathematical programming) as the selection of a best element (with regard to some criteria) from

some set of available alternatives. In the simplest case, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function.

Once the optimal design problem is formulated as shown in the previous sections, the common procedure is to classify it into the existing optimization problems and to look for an algorithm that could solve it. Two main categories of optimization problem are convex and non-convex [44]. The algorithms to find the solution are based on this classification. In many cases, softwares like MATLAB can be used to solve several optimization problems since it has built-in functions. But, it is necessary to understand the algorithm behind these functions to solve the problem in hand, to analyse the results and to modify the algorithm to suit the problem if necessary. The following sections describe various classifications of an optimization problem and the latest and efficient algorithms to find a solution for those problems:

2.4.2 Categories of optimization problems

Recalling equation 3.1, a general optimization problem can be modelled as [45]:

$$\begin{aligned} & \underset{x}{\text{minimize}} && f_0(x) \\ & \text{subject to} && f_i(x) \leq b_i, \quad i = 1, \dots, m. \end{aligned} \tag{2.2}$$

All the optimization problems can be classified as convex and non-convex: The class of optimization problems in which the constraint functions are convex are called **convex optimization** problems [46], which means they satisfy the inequality

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

for all $x, y \in \mathbf{R}^n$ and all $\alpha, \beta \in \mathbf{R}$ with $\alpha + \beta = 1$ and $\alpha \geq 0$ and $\beta \geq 0$.

Two very widely known and used special subclasses of convex optimization: least-squares and linear programming:

- A **least squares** problem is an optimization problem with no constraints and an objective which is a sum of squares of terms of the following form:

$$\text{Minimize } f_0(x) = \|Ax - B\|^2 = \sum_{i=1}^k (a_i^T x - b_i)^2$$

Here, $A \in \mathbf{R}^{k \times n}$ (with $k \geq n$), a_i^T are the rows of A , and the vector $x \in \mathbf{R}^n$ is the optimization variable. Recognizing an optimization problem as a least-squares problem is straightforward; we only need to verify that the objective is a quadratic function (and then test whether the associated quadratic form is positive semidefinite).

The two main techniques in a least square problem are [46]:

1. **Weighted least squares** with objective function

$$f_0(x) = \sum_{i=1}^k w_i (a_i^T x - b_i)^2 \quad ; w_1, \dots, w_k > 0$$

2. Regularization, with objective function

$$f_0(x) = \sum_{i=1}^k (a_i^T x - b_i)^2 + \rho \sum_{i=1}^n x_i^2 \quad ; \rho > 0$$

- **Linear programming (LP)**, is an important class of optimization problems in which the objective and all constraint functions are linear [46, 45] :

$$\begin{aligned} & \underset{x}{\text{minimize}} && c^T x \\ & \text{subject to} && a_i^T x \leq b_i, \quad i = 1, \dots, m. \end{aligned} \quad (2.3)$$

Here the vectors $c, a_1, \dots, a_m \in \mathbf{R}^n$ are scalars $b_1, \dots, b_m \in \mathbf{R}$ are problem parameters that specify the objective and constraint functions. If we consider equation 3.1, an optimization problem is called a linear program if the objective functions satisfy:

$$f_i(\alpha x + \beta y) = \alpha f_i(x) + \beta f_i(y)$$

for all $x, y \in \mathbf{R}^n$ and all $\alpha, \beta \in \mathbf{R}$

Nonlinear optimization (or nonlinear programming, NLP) is the term used to describe an optimization problem when the objective or constraint functions are not linear [46]:

$$\begin{aligned} & \underset{x}{\text{minimize}} && f(x) \\ & \text{subject to} && g_i(x) \leq b_i, \quad \forall i = 1, 2, \dots, m \quad \text{and} \quad x \geq 0 \end{aligned} \quad (2.4)$$

where $f(x)$ and the $g_i(x)$ are given functions of the n decision variables. Sadly, there are no effective methods for solving the general nonlinear programming problem. Methods for the general nonlinear programming problem therefore take several different approaches, each of which involves some compromise: **Local optimization**, in which the compromise is to give up seeking the optimal x , which minimizes the objective over all feasible points. Instead we seek a point that is only locally optimal, which means that it minimizes the objective function among feasible points that are near it, but is not guaranteed to have a lower objective value than all other feasible points.

Quadratic programming (QP) is a special case of NLP with the following mathematical model [46, 45]:

$$\begin{aligned} & \underset{x}{\text{minimize}} && f(x) = c^T x - \frac{1}{2} x^T Q x \\ & \text{subject to} && A^T x \leq b, \quad \text{and} \quad x \geq 0 \end{aligned} \quad (2.5)$$

where c is a row vector, x and b are column vectors, Q and A are matrices, and the superscript T denotes the transpose. QPs generalize both LPs and ordinary least-squares: IF Q is a null matrix, QP reduces to LP and when $Q = Q^T \geq 0$, QP reduces to a least square problem.

When the constraints are also quadratic in a QP, its called a **Quadratic constrained Quadratic Programming (QCQP)** problem [46].

The problems that are not convex are called **non-convex optimization problems** [46]. It encompasses all nonlinear programming problems that do not satisfy the assumptions of convex programming. QPs and QCQPs can be convex or non-convex.

2.4.3 Algorithms for finding the solution

A solution method for a class of optimization problems is an algorithm that computes a solution of the problem (to some given accuracy), given a particular problem from the class, i.e., an instance of the problem [46].

Solution for convex problems

Thus, most of these algorithms to solve a convex problem fall into one of these following three categories.

- **Sequential-approximation algorithms:** includes linear approximation and quadratic approximation methods. These algorithms replace the nonlinear objective function by a succession of linear or quadratic approximations:

1. **Simplex method** is one of the effective methods in solving the LP problems [45]:

The corner points of a feasible region are called the CPF (Corner Point Feasible) solutions. Simplex is an iterative algorithm with the following steps:

- Initialization: Set up to start iterations, including finding an initial CPF solution.
- Optimality test: Is the current CPF solution optimal? If yes, stop. If no, go to the next step.
- Iteration: Perform an iteration to find a better CPF solution.

Extensions and variations of the simplex method also are used to perform postoptimality analysis (including sensitivity analysis) on the model.

2. Ordinary least square problems without any constraints can be solved using linear algebra methods like **SVD (Singular Value Decomposition)**. Consider a QP as in equation 4.4, For such convex quadratic functions, any local minimum is global if $Q = Q^T \geq 0$. In fact, when $Q > 0$, then the unique minimizer is $x^* = -Q^{-1}c$. It turns out one can leverage this approach to minimizing more general functions, using an iterative algorithm, based on a local quadratic approximation of the the function at the current point. The approach can then be extended to problems with constraints, by replacing the original constrained problem with an unconstrained one, in which the constraints are penalized in the objective.

- **Sequential unconstrained algorithms** [46]: includes penalty function and barrier function methods. These algorithms convert the original constrained optimization problem to a sequence of unconstrained optimization problems whose optimal solutions converge to the optimal solution for the original problem:

1. Unconstrained minimization: **Newton's method** : When an unconstrained minimization problem is considered, where we seek to minimize a twice-differentiable function f . If f is convex, global minima might be obtained. We start with initial guess x_0 . At each step t , we update our current guess x_t by minimizing the second-order approximation \tilde{f} of f at x_t , which is the quadratic function. If f is not convex, we might run into a local minima.

2. Constrained minimization: **Interior-point algorithm:**

The basic idea behind interior-point methods is to replace the constrained problem by an unconstrained one, involving a function that is constructed with the original problem functions. The following concepts give a rough sketch of how this algorithm can be implemented.

Concept 1: Shoot through the interior of the feasible region toward an optimal solution.

Concept 2: Move in a direction that improves the objective function value at the fastest possible rate.

Concept 3: Transform the feasible region to place the current trial solution near its center, thereby enabling a large improvement when concept 2 is implemented.

The interior-point approach is limited by the need to form the gradient and Hessian of the function f above. For extremely large-scale problems, this task may be too daunting.

- **Gradient algorithms** [46, 45]:

1. *Unconstrained case:* Gradient methods offer an alternative to interior-point methods, which is attractive for large-scale problems. Typically, these algorithms need a considerably larger number of iterations compared to interior-point methods, but each iteration is much cheaper to process. Perhaps the simplest algorithm to minimizing a convex function involves the iteration $x_{t+1} = x_t - \alpha_t \nabla f(x_t)$ where, $\alpha_t > 0$ is a parameter. The interpretation of the algorithm is that it tries to decrease the value of the function by taking a step in the direction of the negative gradient.
2. *Constrained case:* The gradient method can be adapted to constrained problems, via the iteration $x_{t+1} = P(x_t - \alpha_t \nabla f(x_t))$ where P is the projection operator, which to its argument z associates the point closest (in Euclidean norm sense) to z in \mathbf{C} . Depending on problem structure, this projection may or may not be easy to perform.

Solution for non-convex problems

- **Sequential Unconstrained Minimization Technique (SUMT)** [46]: As the name implies, SUMT replaces the original problem by a sequence of unconstrained optimization problems whose solutions converge to a solution (local optimum) of the original problem.
- **Heuristic algorithms** [47, 48] are recently gaining interest and prove to be efficient most of the times even though they might not produce the exact solution in every case with certainty. Nevertheless, a stochastic high-quality approximation of a global optimum is probably more valuable than a deterministic poor-quality local minimum provided by a classical method or no solution at all.

The most important and versatile heuristic algorithms are briefly described by referring [47, 48] :

1. **Simulated Annealing (SA):**
 - Kirkpatrick, Gelatt and Vecchi (1983).
 - Based on analogy between combinatorial optimization and annealing process of solids.
 - Accepts uphill move, only with given probability (decreases in a number of rounds to zero)
 - Initialise a temperature, T and keep reducing it until the constraints are met for the given number of iterations.
2. **Threshold Accepting (TA):**
 - Dueck and Scheuer (1990).
 - Deterministic analog of Simulated Annealing.
 - Sequence of temperatures T replaced by sequence of thresholds τ .
3. **Tabu Search (TS):**
 - Glover and Laguna (1997).
 - Designed for exploration of discrete search spaces with finite set of neighbor solutions.
 - Avoids cycling (visiting same solution more than once) by use of short term memory (tabu list, most recently visited solutions).
 - A simple way to update memory is to remove older entries from tabu list.
 - Stopping criterion: given number of iterations or number of consecutive iterations without improvement.
4. **Genetic algorithm (GA):**
 - Imitates evolutionary process of species that sexually reproduce.
 - Do not operate on a single current solution, but on a set of current solutions (population).
 - New individuals are generated with cross-over : combines part of genetic patrimony of each parent and applies a random mutation. If new individual (child), inherits good characteristics from parents, it has higher probability to survive.
5. **Ant colonies (AC):**
 - Coloni, Dorigo and Maniezzo (1992).
 - Imitates the way ants search for food and find their way back to their nest.
 - First an ant explores its neighborhood randomly. As soon as a source of food is found it starts to transport food to the nest leaving traces of pheromone on the ground which guide other ants to the source.
 - Intensity of the pheromone traces depend on quantity and quality of food available at source as well as from distance between source and nest, as for a short distance more ants will travel on the same trail in a given time interval.
 - As ants preferably travel along important trails their behavior is able to optimize their work.
 - Pheromone trails evaporate and once a source of food is exhausted the trails will disappear and the ants will start to search for other sources.
 - The search area of the ant corresponds to a discrete set of solutions.
 - The amount of food is associated with an objective function.
 - The pheromone trail is modelled with an adaptive memory.

Meta-heuristics involves one or more of these heuristic procedures working together to find the optimal solution [47, 48].

In general, an optimization problem can be classified into convex and non-convex. Least squares and linear programming fall into the former category where as non-linear programming, especially quadratic programming fall into the latter.

Solutions for convex problems include sequential approximation algorithms, sequential unconstrained algorithms and gradient algorithms. Therefore, categorizing an optimization problem is important since it narrows down the search for algorithms to solve it.

2.5 Steps to be carried out

- *Defining the problem of interest and gathering relevant data* : The problem of interest has already been defined as the optimal and robust design of a wooden robot. The existing techniques are studied and the technique that suits the problem the most is to be chosen. If there is a program to solve this shape optimization problem with inputs as the properties of wood, then analyzing the results will help in making the choice of wood at a later stage. Secondly, the focus would be on sensitivity analysis to find a robust solution.
 - *Formulating a mathematical model to represent the problem* : Calculation of the stiffness matrix of the wooden link and then that of the manipulator is to be done with utmost care. Also, after including the robustness indices, a complete optimization problem would be defined.
 - *Develop a procedure for deriving solutions to the problem from the model* : The solutions for similar optimization problems are studied. This narrows down the search of algorithms to solve it. First priority is to use softwares like MATLAB and to use the already existing functions to get a robust and an optimal solution. Second priority is to improve the algorithm with the main intention of decreasing the computation time while not affecting the solution. Different topology optimization methods are considered. Among these methods, BESO, which is based on the evolutionary process in nature is studied thoroughly and is found to be useful for the optimal and robust design of the link.
 - *Testing* : Once the solution is obtained, it should be tested in a software to check if it behaves the way it is expected to behave. Validation is a crucial step to verify that the thesis going in the right direction.
 - *Implementation* : If time and resources permit, it would be interesting to manufacture a topologically optimized wooden link and to see the difference before and after replacing the metal link on the robot with this wooden link.
-

Chapter 3

FEM

Finite Element Method (FEM) is one of the main parts of the topology optimization process. To come up with a final topology, the initial design domain is discretized into finite elements such that the elements can be removed or kept based on a criteria eventually to get the final topology. More the number of finite elements, better the design. Nonetheless, a mesh-independent filter explained in the following chapters provides a design independent of the number of mesh chosen for the design.

In this chapter, the theory of finite element formulation is expressed and each step in FEA are briefly explained. Mainly the steps involved in FEM are explained with preference to a **4-noded rectangular element** which is used in the later sections for topology optimization.

Lines 206 to 232 perform finite element analysis of the given design domain in the MATLAB code for shape optimization of a wooden robot link given in Appendix A.

3.1 Finite Element Formulation

Finite Element Method is a numerical method to solve problems of engineering and mathematical physics like structural analysis, fluid flow, heat transfer, mass transport and electromagnetic potential [49].

For problems involving complicated geometries, loading and material properties, it is generally not possible to obtain analytical solutions since they involve solutions to the ordinary or partial differential equations which are mostly unobtainable or difficult to obtain. Hence finite element formulation is adopted which results in a system of algebraic equations rather than differential equations. It yields approximate values of unknowns at discrete points in the continuum. Therefore, in the finite element method, instead of solving the problem for the entire body in one operation, equations are formulated for each discretized *finite element* and they are combined to obtain the solution of the whole body.

The general steps in the finite element method are [49] :

- Select the element type.
- Select an interpolation function.
- Define the stress/strain or strain/displacement relationships.

- Derive the element stiffness matrix and equations.
- Assemble the element equations to obtain the global equations and introduce boundary conditions.
- Solve for nodal displacements.

These steps are briefly discussed with some examples:

3.1.1 Selecting the element type and finding the interpolation functions

For a general field problem, the field variable of interest is expressed on an element basis in the discretized form. Field variables can be linear or angular displacements at the nodes. Since the field variables, ϕ_i are computed only at nodes, the field variables at any other point in the element, $\phi^e(x, y, z)$ are given by the following approximate relation [8]:

$$\phi^e(x, y, z) = \sum_{i=1}^M N_i(x, y, z) \phi_i \quad (3.1)$$

where, M is the number of element degrees of freedom. N_i are called the **interpolation or shape functions**. The interpolation functions are most often polynomial forms of the independent variables, derived to satisfy certain required conditions at the nodes.

4 node square element

Rectangular elements are convenient for use in modeling regular geometries [8]. These kind of elements are used in the further sections for topology optimization in two-dimension. The simplest of the rectangular family of elements is the four-node rectangle shown in Figure 3.1, where it is assumed that the sides of the rectangular are parallel to the global Cartesian axes.

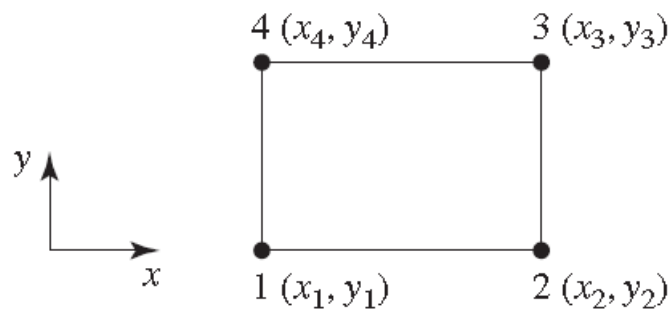


Figure 3.1: 4 node square element [8].

By convention, we number the nodes sequentially in a counterclockwise direction, as shown. As there are four nodes and 4 degrees of freedom, a four-term polynomial expression for the field variable, ϕ is appropriate.

The following expression can be used to ensure geometric isotropy [8] :

$$\phi(x, y) = a_0 + a_1x + a_2y + a_3xy$$

Applying the four nodal conditions $\phi(x_1, y_1) = \phi_1$, $\phi(x_2, y_2) = \phi_2$, $\phi(x_3, y_3) = \phi_3$ and $\phi(x_4, y_4) = \phi_4$ and writing in matrix form gives

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

which gives the polynomial coefficients as

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix}$$

In terms of the nodal values, the field variable is then described by

$$\phi(x, y) = \begin{bmatrix} 1 & x & y & xy \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & x & y & xy \end{bmatrix} \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix}$$

From the equation (3.1), the shape functions can be determined as

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} = \begin{bmatrix} 1 & x & y & xy \end{bmatrix} \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 \end{bmatrix}^{-1}$$

Since, the expression for the interpolation functions seems to be complex and hence a more judicious choice of coordinates is employed to reduce the complexity. The normalized coordinates (also known as natural coordinates or serendipity coordinates) r and s are employed as follows:

$$r = \frac{x - \bar{x}}{a} \quad s = \frac{y - \bar{y}}{b}$$

where $2a$ and $2b$ are the width and height of the rectangle, respectively, and the coordinates of the centroid are

$$\bar{x} = \frac{x_1 + x_2}{2} \quad \bar{y} = \frac{y_1 + y_4}{2}$$

Therefore, r and s are such that the values range from -1 to +1, and the nodal coordinates are as in figure 3.2:

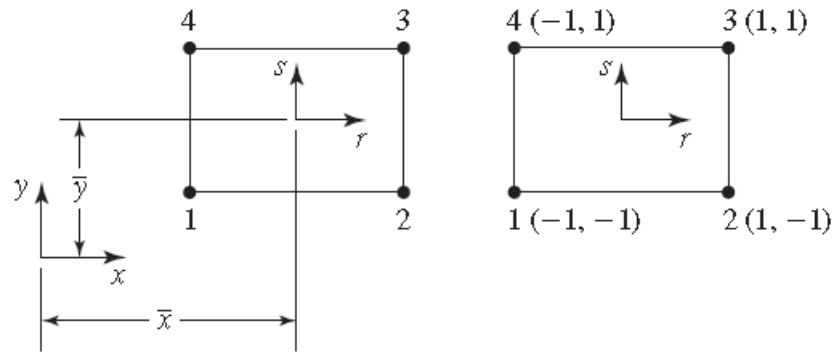


Figure 3.2: Transition to natural coordinates [8].

Considering the condition that the shape functions should have a value of 1 at the corresponding node, the shape functions can be obtained by inspection [8] :

$$N_1(r, s) = \frac{1}{4}(1 - r)(1 - s) \tag{3.2}$$

$$N_2(r, s) = \frac{1}{4}(1 + r)(1 - s) \tag{3.3}$$

$$N_3(r, s) = \frac{1}{4}(1 + r)(1 + s) \tag{3.4}$$

$$N_4(r, s) = \frac{1}{4}(1 - r)(1 + s) \tag{3.5}$$

hence

$$\phi(x, y) = \phi(r, s) = N_1(r, s)\phi_1 + N_2(r, s)\phi_2 + N_3(r, s)\phi_3 + N_4(r, s)\phi_4 \tag{3.6}$$

8 node cubic element

An 8-node element can be used in modelling for a 3-dimensional topology optimization. For a 8 node cubic element (rectangular parallelepiped) as shown in fig 3.3 the interpolation functions can be described as [8] :

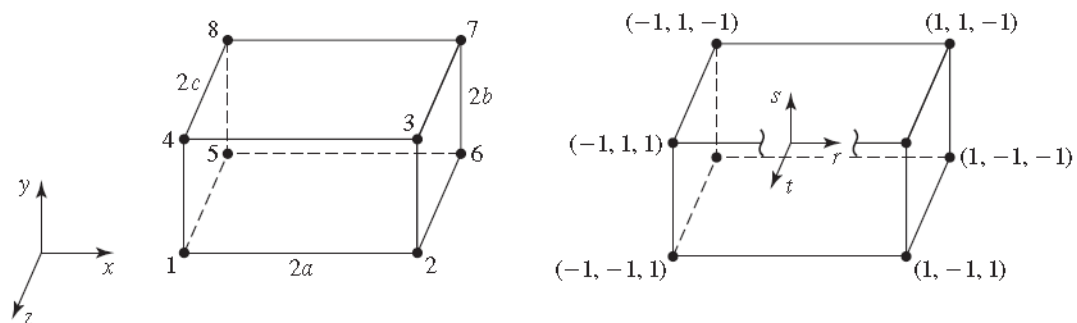


Figure 3.3: 8 noded cubic element in cartesian and natural co-ordinates with origin at the centroid [8].

$$\begin{aligned}
N_1(r, s, t) &= \frac{1}{8}(1-r)(1-s)(1+t) & N_2(r, s, t) &= \frac{1}{8}(1+r)(1-s)(1+t) \\
N_3(r, s, t) &= \frac{1}{8}(1+r)(1+s)(1+t) & N_4(r, s, t) &= \frac{1}{8}(1-r)(1+s)(1+t) \\
N_5(r, s, t) &= \frac{1}{8}(1-r)(1-s)(1-t) & N_6(r, s, t) &= \frac{1}{8}(1+r)(1-s)(1-t) \\
N_7(r, s, t) &= \frac{1}{8}(1+r)(1+s)(1-t) & N_8(r, s, t) &= \frac{1}{8}(1-r)(1+s)(1-t)
\end{aligned}$$

where,

$$\begin{aligned}
r &= \frac{x - \bar{x}}{a} & s &= \frac{y - \bar{y}}{b} & t &= \frac{z - \bar{z}}{b} \\
\bar{x} &= \frac{x_2 - x_1}{2} & \bar{y} &= \frac{y_3 - y_2}{2} & \bar{z} &= \frac{z_5 - z_1}{2}
\end{aligned}$$

and the field variable is described as

$$\phi(x, y, z) = \sum_{i=1}^8 N_i(r, s, t) \phi_i$$

3.1.2 Defining the strain-displacement relationships

Strain-displacement relationship in 3 dimensions for linear displacements u , v and w in x , y and z directions is as follows. More details on the strain for a wooden element are in the next chapter :

$$\epsilon = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & 0 & 0 \\ 0 & \frac{\partial v}{\partial y} & 0 \\ 0 & 0 & \frac{\partial w}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = [L] \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

It is noted that for two dimensions, just the first, second and the fourth columns are considered.

3.1.3 Stress strain relationships for a wooden element

This section explains how the stress strain relations are found out for a wooden element.

Stress tensor σ_{ij} , that completely defines the state of stress at a point inside a material in the deformed placement or configuration can be shown as below in figure 3.4:

By changing the tensor notation to matrix notation, stress tensor and the strain tensor are related by **Hooke's law** as follows [10]:

$$\sigma_i = C_{ij} \epsilon_j = \sigma_i = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \epsilon_i \quad (3.7)$$

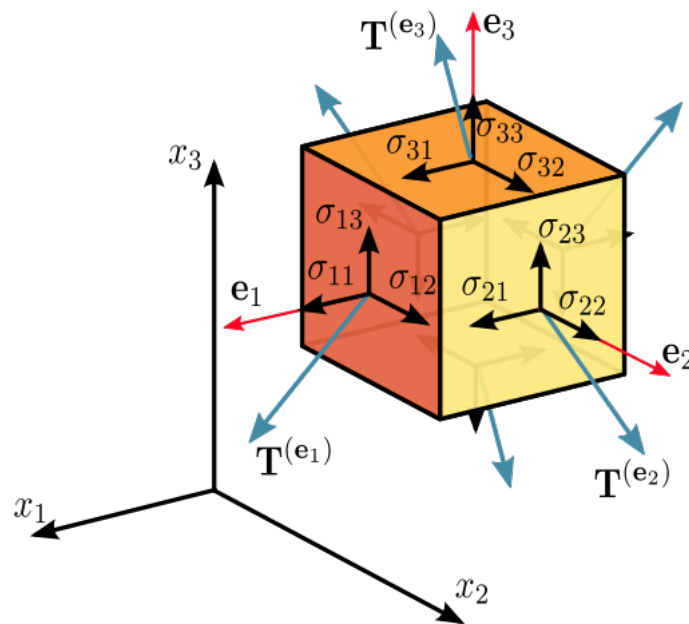


Figure 3.4: Components of stress in three dimensions [9].

C_{ij} is called the **elastic stiffness tensor** written in a matrix form. It is sometimes called as the elastic stiffness matrix. But they are not the same. The derivation of the latter is explained later. If there is a plane of symmetry, we need less independent parameters to express the stiffness matrix. Wood, being an orthotropic material has three planes of symmetry as shown in figure 3.5

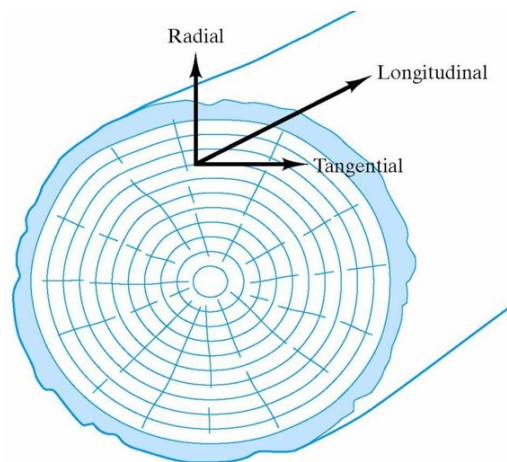


Figure 3.5: Wood modelled as an orthotropic material [10].

Hence, 9 independent parameters are sufficient to express C_{ij} as shown in the following matrix [13]:

$$\sigma_i = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \epsilon_i$$

Rather, it's easy to express the elastic compliance tensor for the tangential(T), radial(R) and longitudinal(L) directions of the wood than the elastic stiffness tensor as shown below in matrix form. σ , τ , ϵ , γ denote the direct stress, shear stress, direct strain and shear strain respectively. E , G and ν are the Young's modulus of elasticity, Shear modulus of elasticity and Poisson's ratio respectively [13]. From equation (3.7)

$$\sigma_i = C_{ij}\epsilon_i$$

Hence,

$$\epsilon_i = C_{ij}^{-1}\sigma_i$$

For a wooden element, this relation can be expressed as

$$\begin{bmatrix} \epsilon_{LL} \\ \epsilon_{RR} \\ \epsilon_{TT} \\ \gamma_{RT} \\ \gamma_{TL} \\ \gamma_{LR} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_L} & -\frac{\nu_{LR}}{E_L} & -\frac{\nu_{LT}}{E_L} & 0 & 0 & 0 \\ -\frac{\nu_{RL}}{E_R} & \frac{1}{E_R} & -\frac{\nu_{RT}}{E_R} & 0 & 0 & 0 \\ -\frac{\nu_{TL}}{E_T} & -\frac{\nu_{TR}}{E_T} & \frac{1}{E_T} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{RT}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{TL}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{LR}} \end{bmatrix} \begin{bmatrix} \sigma_{LL} \\ \sigma_{RR} \\ \sigma_{TT} \\ \tau_{RT} \\ \tau_{TL} \\ \tau_{LR} \end{bmatrix} \quad (3.8)$$

where, σ , τ , ϵ , γ denote the direct stress, shear stress, direct strain and shear strain respectively. E , G and ν are the Young's modulus of elasticity, Shear modulus of elasticity and Poisson's ratio respectively [13].

Since there are 12 variables, 9 are independent due to the relation

$$\frac{\nu_{ij}}{E_i} = \frac{\nu_{ji}}{E_j}, \quad i \neq j; \quad i, j = L, R, T \quad (3.9)$$

where, ν_{ij} is the poisson's ratio. The first letter, i of the subscript refers to direction of applied stress and the second letter, j to direction of lateral deformation. The values of young's moduli E and moduli of torsion G for different species of wood for different density and humidity are listed in table 3.1 [13]:

Species	Density (G/cm^3)	Humidity (%)	E_L (MPa)	E_R (MPa)	E_T (MPa)	G_{RT} (MPa)	G_{TL} (MPa)	G_{LR} (MPa)
Balsa	0.1	9	2490	120	40	10	90	130
Maple	0.58	9.6	10200	1550	890	300	1120	1240
Oak	0.67	11.6	5810	2190	990	400	780	1320
Beech	0.74	10.5	14010	2280	1160	470	1080	1640
Fir	0.39	12	11830	920	510	40	730	760
Pine	0.54	9.7	16610	1120	380	70	680	1780

Table 3.1: List of elastic constants for some wood species [13].

Similarly, poisson's ratio ν for some species of wood at approximately 12% moisture content is listed in table 3.2 [12]. The dashes denote that those values are very small and are less precisely determined than the other values.

Species	ν_{LR}	ν_{LT}	ν_{RT}	ν_{TR}	ν_{RL}	ν_{TL}
Balsa	0.229	0.488	0.665	0.231	0.018	0.009
Maple	0.434	0.509	0.762	0.354	0.063	0.044
Oak	0.350	0.448	0.560	0.292	0.064	0.033
Fir	0.341	0.332	0.437	0.336	-	-
Pine	0.347	0.315	0.408	0.308	-	-

Table 3.2: Poisson's ratios for various species at approximately 12% moisture content [12].

3.1.4 Deriving the elemental stiffness matrix

The elemental stiffness matrix is derived for a 4-node square element. The same procedure can be used for other element types. For a 4-node element, assuming the degrees of freedom at each node to be the linear displacements in x and y directions, u and w , the matrix L relating the strain and displacements $\epsilon = L[u, v]^T$ can be written as [8]

$$L = \begin{bmatrix} \frac{\partial u}{\partial x} & 0 \\ 0 & \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial x} \end{bmatrix}$$

Referring to (3.6), the field variables, u and v can be expressed in terms of the nodal displacements as

$$\begin{bmatrix} u \\ v \end{bmatrix} = [N]\delta = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix}$$

where, N_i are the interpolation functions as defined in equations (3.2) to (3.5) and u_i and v_i are the linear displacements at node i . Thus the relation between the strain and the nodal displacements can be written as :

$$\epsilon = [L] \begin{bmatrix} u \\ v \end{bmatrix} = [L][N]\delta = [B]\delta$$

If the stress-strain relationship is given as

$$\sigma = [D]\epsilon \quad (3.10)$$

The matrix $[D]$ for plain stress of a two-dimensional isotropic element with Young's modulus, E and Poisson's ratio, ν is given by

$$[D] = \begin{bmatrix} -\frac{E}{\nu^2-1} & -\frac{\nu E}{\nu^2-1} & 0 \\ -\frac{\nu E}{\nu^2-1} & -\frac{E}{\nu^2-1} & 0 \\ 0 & 0 & \frac{E(\nu-1)}{2(\nu^2-1)} \end{bmatrix}$$

Similarly, using the stress strain relations for wood, elemental stiffness matrix can be derived for wood that depends on the Young's moduli along the longitudinal and radial directions, Poisson's ratio and shear modulus in the LR plane, E_L , E_R , ν_{LR} and G_{LR} respectively. The D matrix used to derive the stiffness matrix for wood is given below:

$$[D] = \begin{bmatrix} \frac{E_L^2}{E_L - \nu_{LR}^2 E_R} & \frac{\nu_{LR} E_L E_R}{E_L - \nu_{LR}^2 E_R} & 0 \\ \frac{\nu_{LR} E_L E_R}{E_L - \nu_{LR}^2 E_R} & \frac{E_L E_R}{E_L - \nu_{LR}^2 E_R} & 0 \\ 0 & 0 & G_{LR} \end{bmatrix}$$

then the elastic elemental stiffness matrix can be derived as shown below [50]:

The elastic potential energy of an element of a body is given by [50]:

$$U_e = \frac{1}{2} \int_V \sigma^T \epsilon dV \quad (3.11)$$

where, V is the volume of the body, σ and ϵ are the six-dimensional stresses and strains.

The elastic potential energy of an element can also be defined in terms of its elastic stiffness matrix K_e as [50]:

$$U_e = \frac{1}{2} q_e^T K_e q_e \quad (3.12)$$

From, equations (3.10), (3.11) and (3.12)

$$K_e = \int_V [B]^T [D] [B] dV \quad (3.13)$$

Using the above equation, the stiffness matrix for a 4-node isotropic element can be derived as:

$$K_e = \frac{E}{1 - \nu^2} \begin{bmatrix} \frac{3-\nu}{6} & \frac{1+\nu}{8} & \frac{-3-\nu}{12} & \frac{-1+3\nu}{8} & \frac{-3+\nu}{12} & \frac{-1-\nu}{8} & \frac{\nu}{6} & \frac{1-3\nu}{8} \\ \frac{1+\nu}{8} & \frac{3-\nu}{6} & \frac{1-3\nu}{8} & \frac{\nu}{6} & \frac{-1-\nu}{8} & \frac{-3+\nu}{12} & \frac{-1+3\nu}{8} & \frac{-3-\nu}{12} \\ \frac{-3-\nu}{12} & \frac{1-3\nu}{8} & \frac{3-\nu}{6} & \frac{-1-\nu}{8} & \frac{\nu}{6} & \frac{-1+3\nu}{8} & \frac{-3+\nu}{12} & \frac{1+\nu}{8} \\ \frac{-1+3\nu}{8} & \frac{\nu}{6} & \frac{-1-\nu}{8} & \frac{3-\nu}{6} & \frac{1-3\nu}{8} & \frac{-3-\nu}{12} & \frac{1+\nu}{8} & \frac{-3+\nu}{12} \\ \frac{-3+\nu}{12} & \frac{-1-\nu}{8} & \frac{\nu}{6} & \frac{1-3\nu}{8} & \frac{3-\nu}{6} & \frac{1+\nu}{8} & \frac{-3-\nu}{12} & \frac{-1+3\nu}{8} \\ \frac{-1-\nu}{8} & \frac{-3+\nu}{12} & \frac{-1+3\nu}{8} & \frac{-3-\nu}{12} & \frac{1+\nu}{8} & \frac{3-\nu}{6} & \frac{1-3\nu}{8} & \frac{\nu}{6} \\ \frac{\nu}{6} & \frac{-1+3\nu}{8} & \frac{-3+\nu}{12} & \frac{1+\nu}{8} & \frac{-3-\nu}{12} & \frac{1-3\nu}{8} & \frac{3-\nu}{6} & \frac{-1-\nu}{8} \\ \frac{1-3\nu}{8} & \frac{-3-\nu}{12} & \frac{1+\nu}{8} & \frac{-3+\nu}{12} & \frac{-1+3\nu}{8} & \frac{\nu}{6} & \frac{-1-\nu}{8} & \frac{3-\nu}{6} \end{bmatrix}$$

In the same way, the stiffness matrix for wood can be derived. K_e for wood is not displayed here since the elements of it are quite big though it has the dimension 8×8 for a 4-node two dof rectangular node.

In Appendix A, the MATLAB code for topology optimization is given where lines 235 to 247 define the elemental stiffness matrix for yellow poplar in the LR (Longitudinal-Radial) plane.

Once we have the elastic stiffness matrix of the wooden link, it can be used in optimizing the design of the link which will be discussed in the subsequent sections.

3.1.5 Finding the global stiffness matrix and finding the unknown displacements

The corresponding nodal elements of the local stiffness matrices, K_e are added to obtain the global stiffness matrix K for the N finite elements:

$$[K] = \sum_{e=1}^N [K_e]$$

The nodal displacements U are solved for the forces F applied by the following relation :

$$[K]U = F$$

The derived stiffness matrix is validated with a FE software ANSYS. Before that stiffness modelling in ANSYS is understood.

3.2 Stiffness modelling in ANSYS workbench

To verify if FEM code written in MATLAB, for some beams the results are compared with a Finite Element Software, ANSYS whose results are shown later. Hence, the theory behind the modelling in ANSYS workbench is studied and a brief explanation is given below [51].

- Stiffness matrix : ANSYS uses the same stiffness matrix expression for a finite element as given in equation (3.13).
- Failure criteria : ANSYS follows a failure criteria and gives an error if the loads applied are causing material failure [51]. In our case, fatigue loads are not applied and hence this problem need not be addressed.
- Type of elements : To have a correct comparison, a 4-node rectangular element is used in ANSYS. But, in ANSYS reference guide, they are called as 2-D and axisymmetric 4 node quadrialteral solids [51] and in ANSYS terminology, they are called PLANE13 or PLANE42 elements. However the shape functions in the ANSYS guide for these elements are exactly the ones given in equation (3.2). Though 8 node elements are not used in this thesis, they could be used for 3d topology optimization in the future. They are termed as SOLID45 in ANSYS terminology.
- Meshing parameters: All elements are forced to be quadratic and the element size is kept same. each element is of dimension $1mm \times 1mm$ with unit thickness.

Validation of the derived stiffness matrix

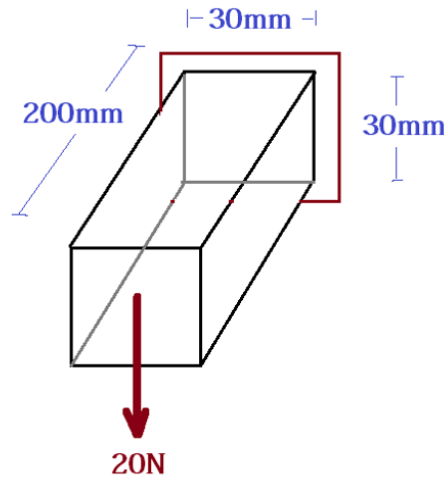


Figure 3.6: Cantilever beam.

A cantilever beam as shown in figure 3.6 is considered. For the 20 N load applied, two types of materials are considered:

1. Structural steel. The properties considered are Young's modulus $E = 200GPa$ and Poisson's ratio, $\nu = 0.3$
2. Yellow poplar. The orthotropic properties in three different directions, longitudinal(L), radial(R) and tangential(T) are taken from the wood handbook [12] as follows: $E_L = 10893.7Mpa$, $\frac{E_T}{E_L} = 0.043$, $\frac{E_R}{E_L} = 0.092$, $\frac{G_{LR}}{E_L} = 0.075$, $\frac{G_{LT}}{E_L} = 0.069$, $\frac{G_{RT}}{E_L} = 0.011$, $\nu_{LR} = 0.318$, $\nu_{LT} = 0.392$, $\nu_{RT} = 0.703$, $\nu_{TR} = 0.329$, $\nu_{RL} = 0.030$, $\nu_{TL} = 0.019$.

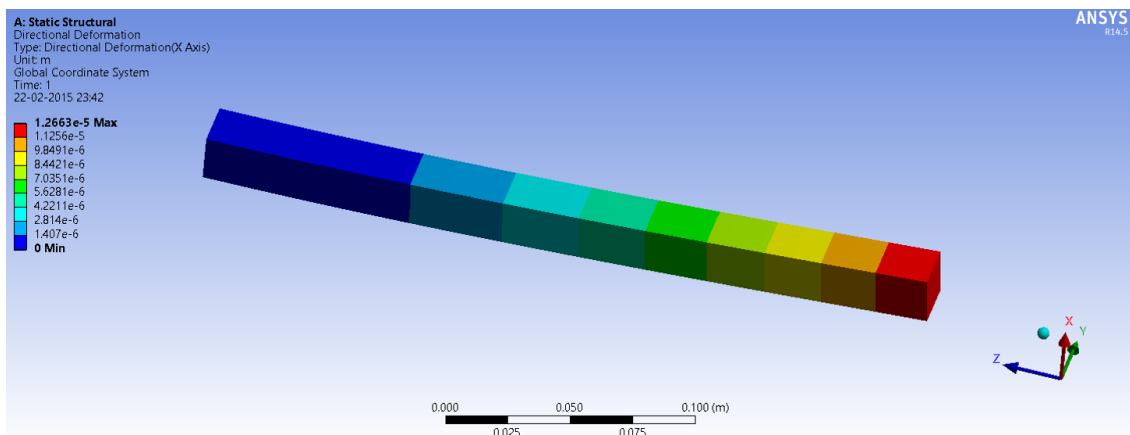


Figure 3.7: FE analysis of a cantilever beam of yellow poplar in ANSYS

The deformation is calculated at the hanging end using the stiffness matrix, K derived as given in the previous section in MATLAB. The deformation is also calculated by modelling the cantilever beam in ANSYS. The results are listed in the following table 3.3. Also the figure 3.7 shows the FE analysis for the yellow poplar in ANSYS where the deformation can be seen on the left top of the image.

Type	ANSYS	MATLAB(derived K)
Structural steel	0.0673mm	0.067mm
Yellow poplar	1.26mm	1.2mm

Table 3.3: Deformation of the cantilever beam.

3.3 Conclusion

FEA is a vital part of the topology optimization process. The steps involved in Finite Element Analysis are explained with preference to the 4-node rectangular element since it is the most widely used element for two-dimensional topology optimization.

Also, the methodology used to derive the stiffness matrix for a wooden element is explained which is validated in ANSYS. The stiffness matrix is used for the topology optimization in the following chapter. Since, the results are validated in ANSYS at every step of the design process, FEM in ANSYS is briefly explained and it shows how the stiffness modelling explained matches well with that of ANSYS.

Chapter 4

Shape optimization of a wooden robot link

The method of Bidirectional Evolutionary Structural Evolution (BESO) [31, 32, 33, 11] is used as the basis for the shape optimization of a wooden robot link.

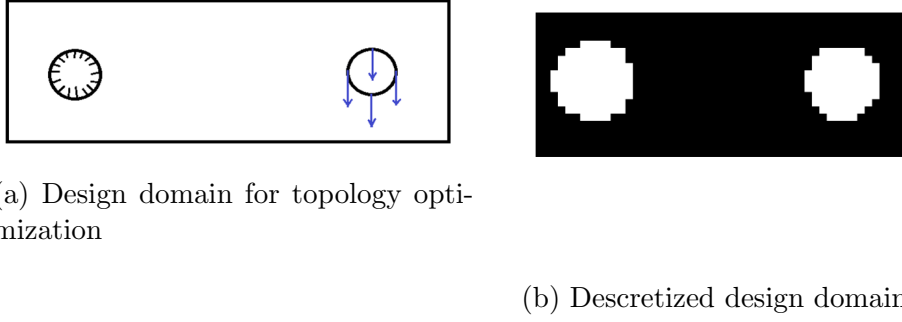
Evolutionary Structural Optimization (ESO) [31] method is inspired by the evolution of organisms over the period of time where material is removed to obtain an optimum topology. BESO (Bi-directional Evolutionary Optimization) is the extension of ESO [11] where the material can be added or removed to obtain an optimum topology.

In ESO/BESO methods, a structure is optimized by removing and adding elements. That is to say that, the element itself, rather than its associated physical or material parameters, is treated as the design variable [11]. The following sections explains the steps involved in the topology optimization of the wooden link along with references to the MATLAB code. The optimized designs are also depicted which are validated using the FE software ANSYS.

4.1 Design domain

The robotic arm for which the topology is optimized has 2 holes with fixed and loaded parts as shown in figure as the design domain 4.1a. The first hole which is fixed is the actuated joint where the motor is connected and the second hole where the load is applied is connected to the next link in the robot. The discretized design domain is as shown in figure 4.1b. It is noted that the material is always kept for a small region around the holes. Moment of inertia is considered always about the center of the fixed hole.

Lines 4 to 60 in the MATLAB code written in Appendix A show how the design domain, BESO parameters and the boundary conditions are defined.



(a) Design domain for topology optimization

(b) Descretized design domain

Figure 4.1: Initial design environment to find an optimum topology

4.2 Optimization problem

Since BESO [11] involves density function x taking values 0 or 1 which do not depend on the material properties, it can be easily adopted to solve a topology optimization problem for wooden materials. Hence, BESO was adopted to solve the following topology optimization problem, to find the optimum topology such that the mass moment of inertia about the fixed hole is minimum under a given set of displacement constraints:

$$\begin{aligned}
 & \underset{x}{\text{minimize}} && I(x) = \sum_{i=1}^N x_i I_i \\
 & \text{subject to} && u_j \leq u_j^* \\
 & && KU = F \\
 & && x_i = x_{min} \text{ or } 1
 \end{aligned} \tag{4.1}$$

where, $I(x)$ is the total moment of inertia which is calculated as the sum of the moment of inertia of all the elements. I_i is the inertia of each element, x_i is called the density function taking values 0 or 1 and N is the total number of elements in the design domain. . To avoid singularity problems, x_i is not allowed to take a 0 value instead x_{min} which is usually a value close to zero. Here, $x_{min} = 0.001$. u_j and u_j^* denote the j th displacement and its constraint where j can be any node in the FE analysis. Here, j is the centre of the second hole where the displacement is constrained. Note that the displacement constraint may be replaced by the mean compliance constraint [52, 53]. K is the stiffness matrix relating the displacement vector U to the load applied F .

Inertia of each element, I_i is calculated as shown below:

Referring to figure 4.2, for the link shown, mass moment of inertia about the centre of each parallelepiped element (whose co-ordinates are (x_1, y_1)) with unit dimensions and mass m ,

$$I_1^i = \frac{m}{12}$$

By the principal axis theorem, moment of inertia about the centre of the fixed hole

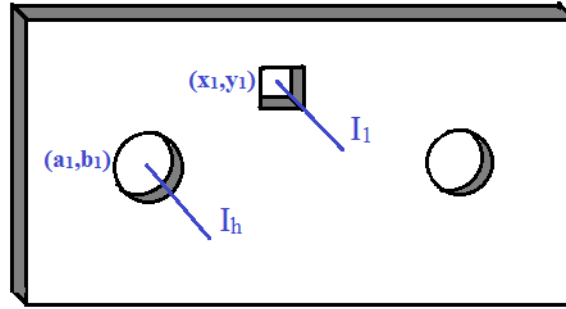


Figure 4.2: Calculating mass moment of inertia of each finite element

with co-ordinates (a_1, b_1) ,

$$I_i = I_h^i = I_1^i + m[(x_1 - a_1)^2 + (y_1 - b_1)^2]$$

Refer lines 91 and 92 of the MATLAB code in Appendix A.

A similar problem with volume as the objective function and a displacement constraint has been solved in [54]. Using the same technique, inertia can be minimized for a given displacement constraint.

In order to solve this problem using the BESO method, the displacement constraint is added to the objective function by introducing a Lagrangian multiplier λ (following the procedure in [52]) :

$$f_1(x) = \sum_{i=1}^N I_i x_i + \lambda(u_j - u_j^*) \quad (4.2)$$

Refer line 96 of the MATLAB code in Appendix A.

It is seen that the modified objective function is equivalent to the original one and the Lagrangian multiplier can be any constant if the displacement constraint is satisfied.

4.3 The BESO procedure adopted and flowchart

The evolutionary iteration procedure to solve the problem (4.1) is given as follows [11]:

1. **Define the design domain, boundary conditions** : The design domain is a link with two holes as defined in section 4.1. Boundary conditions are the fixed and loaded nodes as shown in figure 4.1a.
2. **Discretize the design domain using a finite element mesh and assign initial property values (0 or 1) for the elements to construct an**

initial design : Finite element meshing is done as explained in the chapter on FEM. 4-node square elements are used to mesh. The initial design looks like figure 4.1b. The number of finite elements are the choice of the user. For example, for a design domain of dimensions 50mm by 20mm, Meshing of 50 by 20 elements or 100 by 40 elements can be used. It is noted that, more the number of finite elements, more the computational time to obtain an optimum topology.

3. **Perform finite element analysis and then calculate the elemental sensitivity number** : Knowing the loads applied and the global stiffness matrix K , finite element analysis is done to obtain the displacement at each node. This displacement is used to obtain the sensitivity number which will be explained in section 4.4. It can be considered as the derivative of the objective function or how the objective function changes with changes in the design variable or density function x_i .
 4. **Filter the sensitivity number and average it with its history information and then save the resulted sensitivity number for next iteration** : The algorithm is based on BESO and hence is inherent to problems like checkerboards (consecutive 0-1 values making the final design difficult to interpret and manufacture) and mesh-dependencies (optimal design depends on the mesh when it is varied from being coarse to fine meshing). Filtering the sensitivities as in section 4.5 and averaging them with their history information can help tackle the problem. BESO is also known to cause disturbances and sometimes it might not converge to a point, These problems are tackled again by averaging the sensitivity numbers as it is made clear in section 4.6.
 5. **Determine the j th displacement in the next iteration, U_j^{k+1} and set it to the displacement constraint U_j^* as specified by the user** : Knowing the displacement, U_j^k in the current iteration k at the j th node, the displacement in the next iteration can be found out using the equation (4.6).
 6. **Determine the threshold of sensitivities and the corresponding target volume for the next iteration** : By setting the displacement in the next iteration to be the constraint i.e. $U_j^{k+1} = U_j^*$, threshold of the sensitivities and the corresponding design values x_i can be calculated. By summing the values of x_i over the total volume of the design domain gives the corresponding volume fraction V_c . The methodology to find the threshold th and the corresponding volume fraction V_c are explained in section 4.7.
 7. **Add and delete elements according to the procedure described in section 4.8** : Once, the volume, V_c is determined, to ensure that the design slowly converges to this volume, only a small amount of material is added removed. Hence, volume added or deleted is decided by the evolution ratio ER as shown in equation (4.7). Section 4.8 describes how it is done. Nonetheless, this step is optional just to see the gradual change of volume from the initial domain.
 8. **Repeat steps 2–7 until the convergence criterion is satisfied** : The steps are repeated until volume fraction V_c or the optimal design x_i corresponding to the displacement constraint is achieved and the convergence criterion (4.8) is satisfied.
-

Each step is described in detail in the following sections. A flowchart for shape optimization of the wooden link using BESO is given in following figure 4.3.

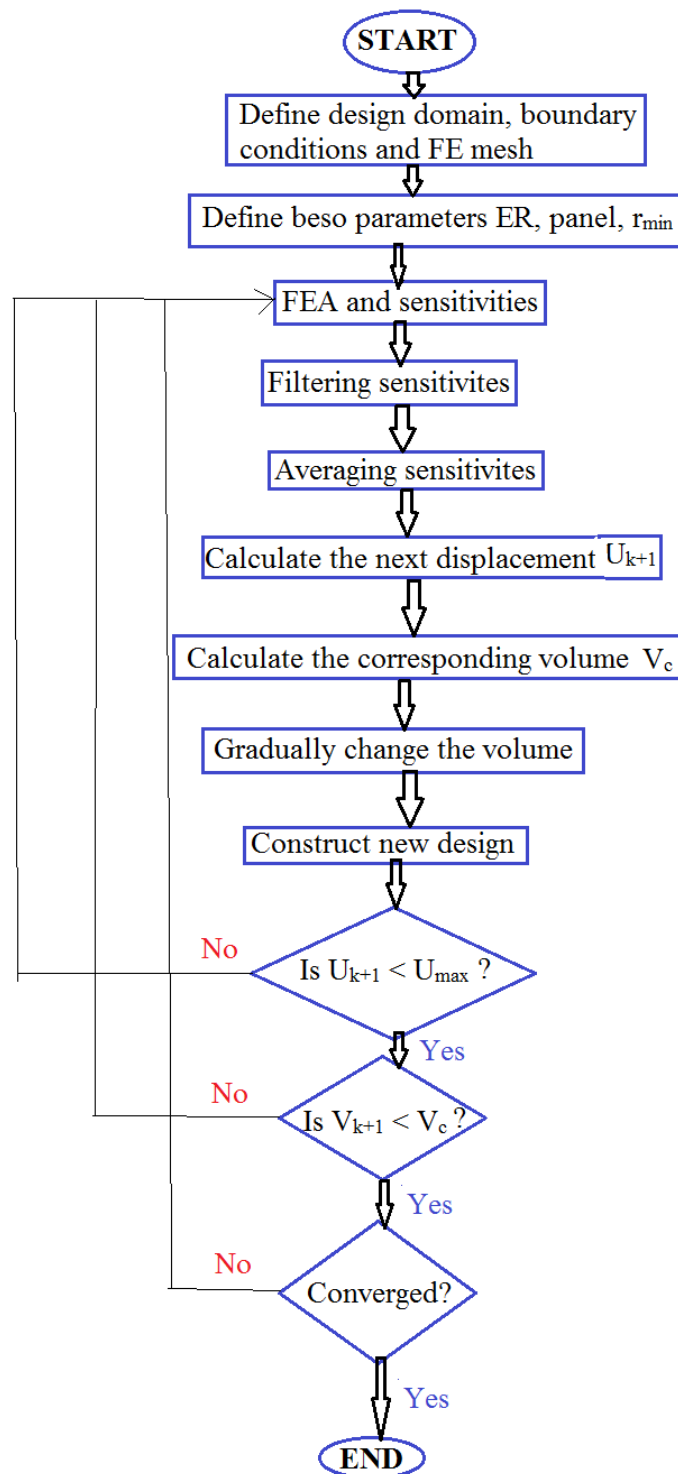


Figure 4.3: Flowchart for the shape optimization of the wooden link.

4.4 Sensitivity analysis

The core of the BESO procedure in sensitivity analysis. It is quite similar to the gradient descent method in mathematical programming. The sensitivity of each element is calculated and a threshold is defined based on the constraints. The element is kept ($x = 1$) or removed ($x = 0.001$) if the sensitivity of that element is more or less than the threshold. The latter steps are explained in sections 4.7 and 4.8.

The derivative of the modified objective function $f_1(x)$ defined in equation (4.2) is

$$\frac{df_1}{dx_i} = I_i + \lambda \frac{du}{dx_i}$$

To calculate $\frac{du_j}{dx_i}$, a virtual unit load f_j is introduced, in which only the corresponding j th component is equal to unity and all other components are equal to zero. Therefore,

$$u_j = f_j^T u$$

Based on the sensitivity analysis in previous sections, we obtain

$$\frac{du_j}{dx_i} = -px_i^{p-1} u_{ij}^T K_i^0 u_i \quad (4.3)$$

where u_{ij} is found from the following adjoint equation

$$f_j - K u_{ij} = 0$$

The above equation shows that u_{ij} is the virtual displacement vector of the i th element resulted from a unit virtual load f_j . Note that the dimension of u_i and u_{ij} is the same as that of u . However, all components in u_i and u_{ij} that are not related to element i are zero. Similarly, the dimensions of K_i^0 are the same as those of K but all components in K_i^0 that are not related to element i are zero. Hence, substituting (4.3) in the equation (4.2), we get

$$\frac{df_1}{dx_i} = I_i - \lambda p x_i^{p-1} u_{ij}^T K_i^0 u_i$$

Note that here, the virtual load is considered the same as the actual load applied. The reason behind that is the displacement that matters the most is at the centre of the second hole where the loads are applied. That's the node j where the constraint is applied. Considering the nodes around it where the loads are applied, will automatically constrain the displacement at the centre. Moreover, the total virtual load applied should be unity as is the total actual load applied.

When a uniform mesh is used (i.e. elements being of the same volume), the relative ranking of sensitivity of each element can be defined by the following sensitivity number

$$\alpha_i = -\frac{1}{\lambda p} \left(\frac{df_1}{dx_i} - I_i \right) = x_i^{p-1} u_{ij}^T K_i^0 u_i \quad (4.4)$$

Sensitivity numbers for every element are calculated in line 93 of the MATLAB code in Appendix A.

4.5 Filter scheme for BESO method

As mentioned in the literature review, often the topology optimization procedures result in checkerboard patterns where there are consecutive white and black regions like a chessboard. Sometimes the design depends on the number of finite elements meshed. To avoid checkerboards and mesh-dependencies, a filter scheme is to be applied. For BESO, the following heuristic and simple filter scheme is applied which is quite similar to the SIMP filter scheme [11]. Before applying the filter scheme, nodal sensitivity numbers which do not carry any physical meaning on their own are defined by averaging the elemental sensitivity numbers as follows [11]

$$\alpha_j^n = \sum_{i=1}^M w_i \alpha_i^e$$

where M denotes the total number of elements connected to the j th node. w_i is the weight factor of the i th element and $\sum_{i=1}^M w_i = 1$. w_i can be defined as

$$w_i = \frac{1}{M-1} \left(1 - \frac{r_{ij}}{\sum_{i=1}^M r_{ij}} \right)$$

where r_{ij} is the distance between the centre of the i th element and the j th node. The above weight factor indicates that the elemental sensitivity number has larger effect on the nodal sensitivity number when it is closer to the node. The above nodal sensitivity numbers will then be converted into smoothed elemental sensitivity numbers. This conversion takes place through projecting nodal sensitivity numbers to the design domain. Here, a filter scheme is used to carry out this process. The filter has a length scale r_{min} that does not change with mesh refinement. The primary role of the scale parameter r_{min} in the filter scheme is to identify the nodes that will influence the sensitivity of the i th element. This can be visualized by drawing a circle of radius r_{min} centred at the centroid of i th element, thus generating the circular sub-domain Ω_i as shown in Figure 4.4.

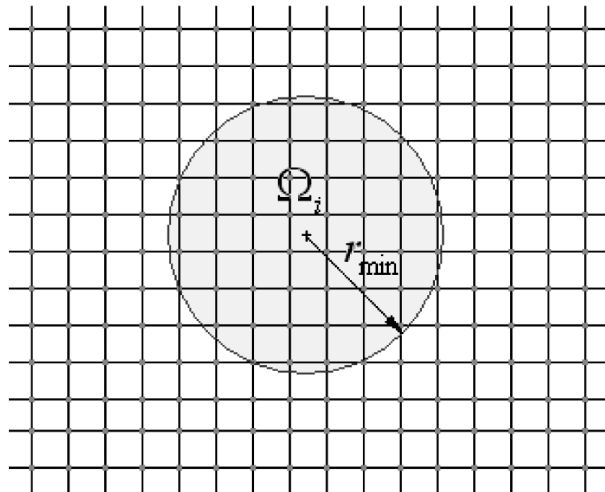


Figure 4.4: Nodes located inside the circular sub-domain Ω_i are used in the filter scheme for the i th element [11].

Usually the value of r_{min} should be big enough so that Ω_i covers more than one element. The size of the sub-domain Ω_i does not change with mesh size. Nodes located inside Ω_i contribute to the computation of the improved sensitivity number of the i th element as

$$\alpha_i = \frac{\sum_{j=1}^K w(r_{ij}) \alpha_j^n}{\sum_{j=1}^K w(r_{ij})}$$

where K is the total number of nodes in the sub-domain Ω_i , $w(r_{ij})$ is the linear weight factor defined as

$$w(r_{ij}) = r_{min} - r_{ij} \quad j = 1, 2, \dots, K$$

It can be seen that the filter scheme smoothes the sensitivity numbers in the whole design domain. Thus, the sensitivity numbers for void elements are automatically obtained. They may have high values due to high sensitivity numbers of solid elements within the sub-domain Ω_i . Therefore, some of the void elements may be changed to solid elements in the next iteration.

The mesh-independency filter used by Sigmund and Petersson [55] is similar to this filter except that they use element sensitivities than the nodal sensitivities. Also, since they include the density of the element in the filter, the sensitivity number for void elements will be infinite. The BESO filter produces results very similar to those obtained by applying a local gradient constraint [56]. Also, averaging the sensitivities as explained in the following section 4.6 is found to be an effective way to avoid mesh-dependencies.

Subroutine lines 181 to 196 in the MATLAB code in Appendix A are for filtering the sensitivities.

4.6 Stabilizing the Evolutionary Process

Unlike SIMP and homogenization methods, with ESO/BESO methods, large oscillations are often observed in the evolution history of the objective function [11]. The reason for such chaotic behaviour is that the sensitivity numbers of the solid (1) and void (0) elements are based on discrete design variables of element presence (1) and absence (0). This makes the objective function and the topology difficult to converge. Huang and Xie [57] has found that averaging the sensitivity number with its historical information is an effective way to solve this problem. The simple averaging scheme is given as

$$\alpha_i = \frac{\alpha_i^k + \alpha_i^{k-1}}{2} \quad (4.5)$$

where k is the current iteration number. Then let $\alpha_i^k = \alpha_i$ which will be used for the next iteration. Thus, the updated sensitivity number includes the whole history of the sensitivity information in the previous iterations.

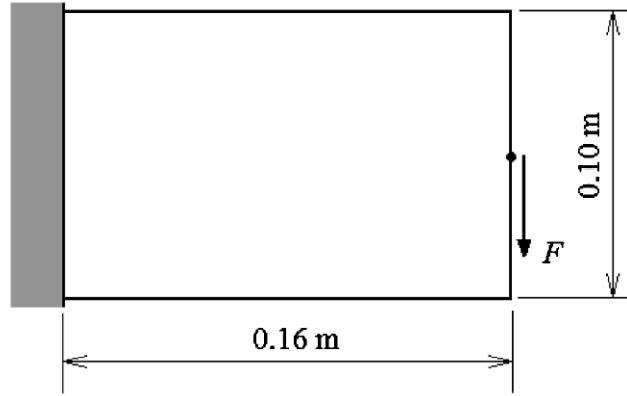


Figure 4.5: Design domain of a short cantilever [11].

For example, for a cantilever beam shown in figure 4.5 with a coarse mesh of 32 by 20 elements, Figure 4.6b shows the evolution history obtained by adopting the stabilization scheme defined in equation (4.5). Compared to the result in Figure 4.6a, the new solution is highly stable in both the topology and the objective function (the mean compliance) after the constraint volume fraction (50%) is achieved.

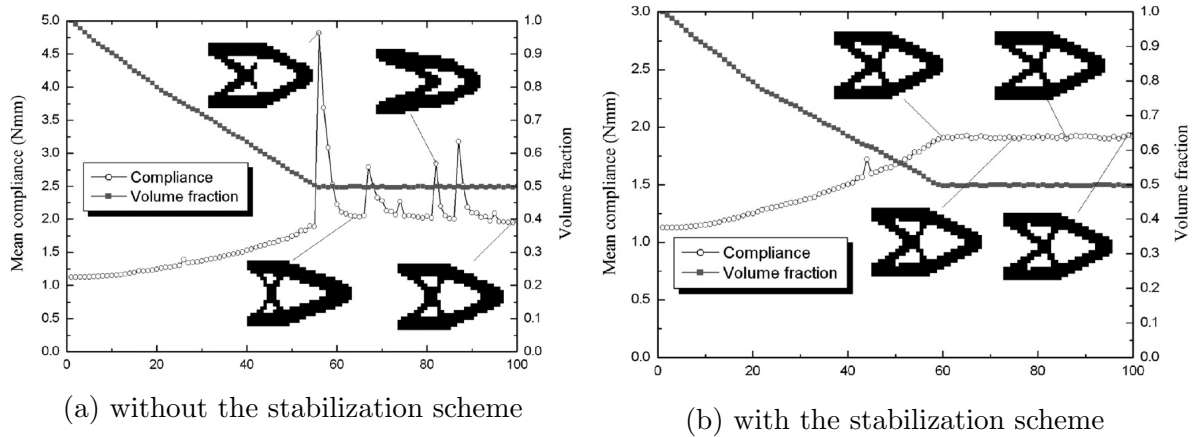


Figure 4.6: Comparison of evolution histories [11]

Line 105 to 107 in the MATLAB code in Appendix A stabilize the evolution process.

4.7 Determination of the structural volume to satisfy the constraints

Once, the sensitivities are found, filtered and averaged, the prescribed constraints are considered and the design x_i which meets these constraints is found out. In other words, the corresponding volume fraction V_c can be found out which is used later to evolve gradually to an optimal design from an initial design.

With the sensitivity of the displacement u_j in equation (4.3), the variation of the

displacement due to changes in design variables can be estimated by

$$u_j^{k+1} = u_j^k + \sum_i \frac{du_j^k}{dx_i} \Delta x_i \quad (4.6)$$

where u_j^k and u_j^{k+1} denote the j th displacement in the current and next iterations respectively. From the above equation, the threshold of the sensitivity number as well as the corresponding volume, V_c , can be easily determined by letting $u_j^{k+1} = u^*$.

The threshold is determined using the bi-sectioning algorithm given for SIMP in [27]. The bi-sectioning algorithm is initialized by taking the lowest sensitivity of the elements as l_1 and the highest one as l_2 . Initially, the threshold is considered as the half of the bounds l_1 and l_2 . Then the threshold, th is calculated at every iteration such that the constraint is satisfied. This procedure is repeated until the interval which bounds the threshold has its size less than the convergence criteria. Also the design is changed at every iteration in this subroutine and the final design values of x_i are obtained corresponding to the threshold, th that satisfies the constraints. The design is changed by adding and deleting elements. The general procedure to add and delete the elements is described in the following section 4.8 Using x_i , the volume fraction V_c is calculated which is just the sum of all values of x_i over the total volume of the initial design domain. If x_{new}^{ij} is the design for which the constraints are satisfied, then, the corresponding volume is

$$V_c = \frac{\sum_{i=1}^N \sum_{j=1}^M x_{new}^{ij}}{N \times M}$$

where, N and M are the number of finite elements in the x and y directions of the design domain. Their values will be proportional to the dimensions of the initial design domain. Now, to reach this volume fraction, the following section explains how the elements are removed and added.

Subroutine lines 130-160 in the MATLAB code in Appendix A find out the corresponding volume fraction V_c satisfying the constraint.

4.8 Element Removal/Addition and Convergence Criterion

We now have the final design and the corresponding volume fraction V_c . However, the resultant volume fraction V_c may be much larger or far smaller than that of the current design. In order to have a gradual evolution of the topology, the following equation is adopted to determine the structural volume for the next iteration:

$$V^{k+1} = \begin{cases} \max(V^k(1 - ER), V^c) & \text{when } V^k > V^c \\ \min(V^k(1 + ER), V^c) & \text{when } V^k \leq V^c \end{cases} \quad (4.7)$$

The above equation ensures that the volume change in each iteration be less than the prescribed evolutionary volume ratio, ER , which defines the maximum variation of the structural volume in a single iteration. But this step is optional since a

design which meets the constraints based on the sensitivity of the objective function is already obtained.

To add or remove elements to attain the required design gradually, the sensitivity numbers of all elements, both solid and void, are calculated as described in the section 4.4. The elements are sorted according to the values of their sensitivity numbers (from the highest to the lowest). For solid element (1), it will be removed (switched to 0) if

$$\alpha_i \leq \alpha_{del}^{th}$$

For void elements (0), it will be added (switched to 1) if

$$\alpha_i > \alpha_{add}^{th}$$

where α_{del}^{th} and α_{add}^{th} are the threshold sensitivity numbers for removing and adding elements, and α_{del}^{th} is always less than or equal to α_{add}^{th} . α_{del}^{th} and α_{add}^{th} can be considered as the same and the threshold can be found out by the bi-sectioning algorithm explained in the previous section 4.7.

The cycle of finite element analysis and element removal/addition continues until the corresponding design or volume fraction V_c is reached while the displacement constraints are satisfied and the following convergence criterion (4.8) (defined in terms of the change in the objective function) is satisfied.

$$error = \frac{|\sum_{i=1}^N I_{k-i+1} - \sum_{i=1}^N I_{k-N-i+1}|}{\sum_{i=1}^N I_{k-i+1}} \leq \tau \quad (4.8)$$

where I is the objective function denoting moment of inertia about the centre of the fixed hole, k is the current iteration number, τ is a allowable convergence tolerance and N is an integer number. Normally, N is selected to be 5 which implies that the change in the mean compliance over the last 10 iterations is acceptably small.

Subroutine ADDEL : lines 163-178 in the MATLAB code in Appendix A are to find out the new design by adding or deleting elements

4.9 Implementation

The MATLAB program given in Appendix A is implemented for the design domain as shown in figure 4.1a. Two cases are considered :

CASE 1 : Forces on the second hole of the link are acting only in one direction : As shown in figure 4.1a, the forces are acting only in y direction. This is usually the case when the robot is vertical and due to gravity, the forces act only in one direction. The procedure in section 4.3 is adopted to solve this kind of problem.

CASE 2 : Forces on the second hole might act in opposite directions : This is the case when the robot is horizontal. Referring to figure 4.1a, the forces

might be in y direction when the link is moving in one direction and in negative y direction when the link starts moving in the opposite direction. It is obvious that in this case, the final optimal design should be symmetric about the horizontal axis AA' in figure, 4.1a to account for forces in both directions. Hence, an extra step is included in the optimization algorithm :

Once the sensitivities are filtered, the sensitivity of an element and its mirror element about the axis AA' are checked and the maximum of these sensitivities is considered as the new sensitivity value for both the elements.

In Appendix A, the lines of the subroutine 198 to 203 are to obtain a symmetrical design for CASE 2.

Consider a displacement constraint of $u_j^* = 0.02mm$ for a design domain of dimension $50mm - by - 20mm$ (hence $n_{elx} = 50$ and $n_{ely} = 20$) and unit thickness. With BESO parameters, evolutionary rate, $er = 0.02$ and $r_{min} = 3$, the optimal designs for the two cases mentioned above are shown in figure 4.7 :

```
[U]=beso_inertia(50, 20, 0.02, 3)
```



(a) Case 1 : Forces in only one direction (b) Case 2 : Forces in opposite directions

Figure 4.7: Optimal design for the two cases based on direction of force applied

For a displacement constraint $u_j^* = 0.02mm$ as shown in Appendix A, at the centre of the second hole, case 1 yielded a displacement of $u_j = 0.0202mm$ and the symmetric design of case 2 resulted in $u_j = 0.0192mm$. These displacements are quite close to the constraint indicating the constraints are satisfied while minimizing the inertia of the wooden robot link.

4.10 Validation of deformation for the link with two holes

Both these final designs for the two cases mentioned in the previous section 4.9 are modelled in ANSYS to check if the values of the deformations at the centre of the

Type	ANSYS	MATLAB(softbeso code)
Case 1 : Force only in one direction	0.015	0.020mm
Case 2 : Force in two opposite directions	0.016	0.019mm

Table 4.1: Deformation of the optimized topology of a cantilever beam.

second hole match. The figure 4.8 indicates the FE analysis done in ANSYS. The deformations are compared in the table 4.1 and it is concluded that they match quite well.

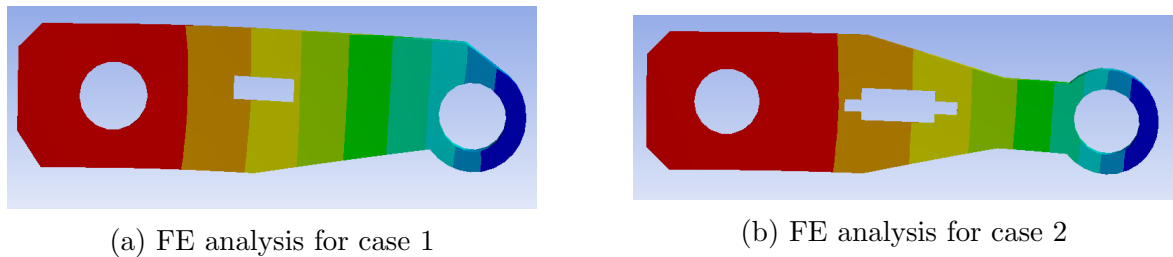


Figure 4.8: FE analysis of the optimized designs in figure 4.7 in ANSYS

4.11 Robust design

It is evident by now that the properties of wood might vary due to differences like temperature, geographical location, humidity etc. Hence robust design approach is studied to be able to tackle the problem. The following figure 4.9 shows the variation of properties of different materials. For example, poplar wood in longitudinal directions has its Young's modulus varying from 0.8GPa to 1.4GPa with 1.1GPa as the nominal value.

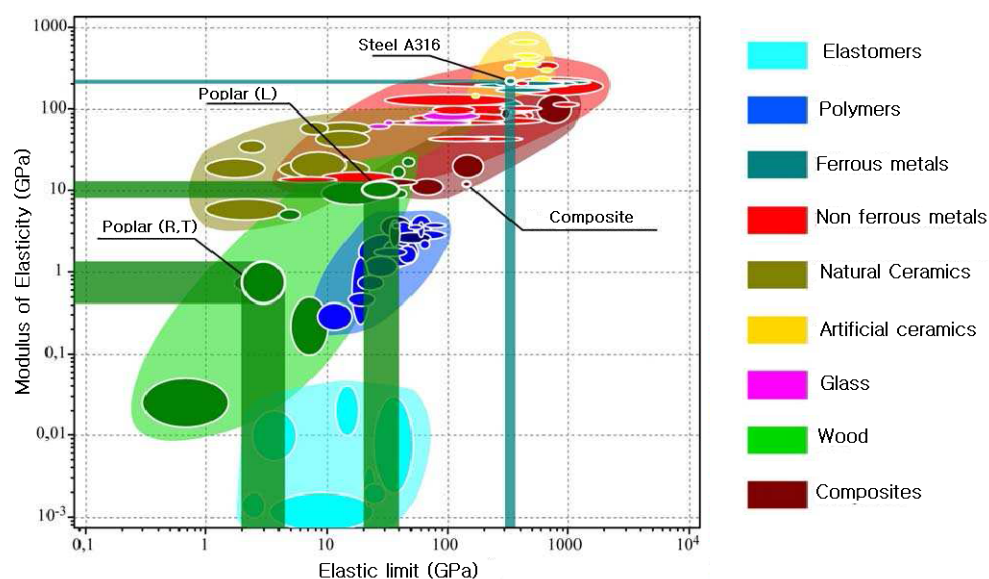


Figure 4.9: Variation of mechanical properties of several materials [2].

The design which is least sensitive to these changes can be formulated as an optimization problem with the objective function being

$$C = F^T dU$$

where, F is the force vector with the dimension of number of nodes in the FEM, C is the compliance due to the change in deformation. A smaller value of C implies a less sensitive design. Hence C can be called as the **Robustness index** which is to be minimized in the optimization problem. dU is a vector having the same dimension as F and it contains changes in deformation due to the changes in the properties of the wood. It can be expressed as follows

$$dU = (K + dK)^{-1} dKU$$

which is derived from $KU = F$, the changes in dK are due to the changes in the Young's modulus, E_L and E_R denoted as dE_L and dE_R of wood as shown in figure 4.9. It is calculated in the following way :

$$dK = \frac{\partial K}{\partial E_L} dE_L + \frac{\partial K}{\partial E_R} dE_R$$

A multi-objective optimization problem can be formulated considering both the inertia and the robustness index which is not done here due to the time constraint.

4.12 Conclusion

The core of the thesis is to perform the topology optimization. As explained in the literature review, BESO is chosen to solve the topology optimization problem to minimize the inertia since it has many advantages compared to other topological optimization problems.

The stages in a topology optimization include Finite Element Analysis, Sensitivity Analysis and Filtering and Stabilization to avoid some inherent problems associated with the optimization problem. These stages are explained and the equations used are derived which are useful to apply for minimizing the mass moment of inertia.

At every stage, reference is done to Appendix A to better understand the MATLAB code written for the shape optimization of the wooden robot link. Finally, the obtained optimized designs are modelled in ANSYS to compare the results and they are seen to match well, though the little discrepancies may be due to the modelling part. One thing to note is that the forces might act in different directions as shown in section 4.9. One of the cases is chosen based on the application so that the corresponding final design can be selected.

Robustness index gives an idea on the sensitivity of the design to the change in parameters of the wood due to different uncontrollable factors.

Chapter 5

Conclusions and future work

Conclusions

Recent developments in Robotics are tremendous and robots are used almost everywhere. But the environmental impact of these bots made of components of metals or alloys or plastics are often ignored. Hence, the project aims at replacing one of the links of a parallel manipulator with an environment-friendly material, wood.

High speed parallel manipulator **IRSbot2**, developed in IRCCyN is introduced and the the aim of the project is dictated as replacing one of its metal links with a wooden link. The low-embodied energy, low carbon impact and sustainability of wood make it a better choice in replacing the metal links. But, it also has its demerits. Since wood is **orthotropic**, wood modelling is not simple. The change of properties of wood with moisture and temperature make it challenging to design a wooden link.

The problem considered is, the **optimal design** of the link. For the optimal design of a link considering mass moment of inertia, topological optimization is employed. The shape optimization of a wooden robot link with moment of inertia as the objective function and a deformation constraint is solved adopting **BESO** method. All the codes for **topology optimization** are written in MATLAB (Appendix A gives the MATLAB code written for the topology optimization of a wooden robot link) where the discretization of the design domain is done based on **Finite Element Analysis**. Hence it is very important to validate the results with the help of a Finite Element software. Here, ANSYS is used to find out the deformations at the required points of the design if it matches with the deformation obtained in MATLAB. **Robust design** is a must due to the use of wood for the design. A **robustness index** is introduced to obtain a least sensitive design.

Future work

Using the **robustness index** as shown in the last chapter, a multi-objective optimization problem is to be solved with objective functions as moment of inertia and the robustness index. BESO code can be applied for the optimization by considering the objective function as the weighted sum of those objective functions. Drawing a **Pareto front** will let us make a decision on which design is to be chosen for the required application of replacing the metal link of the robot.

Once, the optimal and robust design of the wooden link are accomplished, optimal design of the whole manipulator is to be done with the metallic links replaced by wood. Here, the idea would be to perform the **stiffness modelling** of the manipulator and to find a **robust design** for the manipulator with the wooden link that can match the performance of the original manipulator using standard optimization techniques.

This work will be continued under the project of **RobEcolo**, whose purpose is to prove, by developing brand new optimal and robust design approaches (for each design step, i.e. conceptual design, embedded design, detailed design, prototyping) combined with advanced controllers, that it is possible to reach the accuracy performance of current industrial robots made of metallic materials which have stable dimensions with respect to the variation of the atmospheric conditions and constant or known mechanical properties. The objective for the project is defined as the design of a 2 degrees of freedom wooden parallel robot with repeatability lower than 500 microns, deformations (under a load of 1 kg) lower than 500 microns in a workspace of about $500mm \times 500mm$.

Appendices

Appendix A

MATLAB code for shape optimization of the wooden link

The next code will be directly imported from a file

```
1 %%%%% SHAPE OPTIMIZATION OF A WOODEN ROBOT LINK %%%%%%%%%%%%%%
2 %%%%% BASED ON THE SOFT-KILL BESO CODE BY X. HUANG and Y.M. Xie %%%%%
3 function [U]=beso_inertia (nelx , nely , er , rmin) ;
4 %% INITIALIZE BESO PARAMETERS
5 nelx=50;
6 nely=20;
7 rmin=1.5;
8 er=0.02;
9
10 x(1:nely , 1:nelx) = 1; % Initial design value for optimization
11 volfrac=1; % Initial volume fraction
12 i = 0;
13 change = 1.;
14 penal = 3.; % penalization power as in SIMP
15 Umax=-200; % Constraint on deformation
16 lambda=1; % Lagrangian parameter
17
18 %% DEFINE THE DESIGN DOMAIN
19 % Two holes in the link
20 r1=nely/7; r2=nely/7;
21 b1=nely/2; b2=nely/2;
22 a1=nelx/6; a2=5*nelx/6;
23 e1=nely/12; e2=nely/12;
24 for ely=1:nely
25     for elx=1:nelx
26         if sqrt((ely-b1)^2+(elx-a1)^2)<r1 || sqrt((ely-b2)^2+(elx-a2)
^2)<r2
27             passive(ely , elx)=1;
28             x(ely , elx)=0.001;
29         else
30             passive(ely , elx)=0;
31         end
32 % Material for a thickness e=e1=e2 around the holes
33     if sqrt((ely-b1)^2+(elx-a1)^2)>r1 && sqrt((ely-b1)^2+(elx-a1)
^2)<r1+e1 || sqrt((ely-b2)^2+(elx-a2)^2)>r2 && sqrt((ely-b2)^2+(elx
-a2)^2)<r2+e2
34         keep(ely , elx)=1;
35         x(ely , elx)=1;
```

```

36         else
37             keep(ely , elx)=0;
38         end
39     end
40 end
41 %% DEFINING BOUNDARY CONDITIONS
42 %FIXED HOLE
43 fixed(1:(nely+1)*(nelx+1))=0;
44 for ely=1:nely
45     for elx=1:nelx
46         if round(sqrt((ely-b1)^2+(elx-a1)^2))==round(r1)
47             fixed((nely+1)*(elx-1)+ely) = 1;
48         end
49     end
50 end
51 %HOLE WITH LOAD
52 load(1:(nely+1)*(nelx+1))=0;
53 for ely=1:nely
54     for elx=1:nelx
55         if elx==round(a2) && ely==round(b2-r2) || elx==round(a2) &&
ely==round(b2+r2) || elx==round(a2-r2) && ely==round(b2) || elx==
round(a2+r2) && ely==round(b2)
56             load((nely+1)*(elx-1)+ely)=1;
57         end
58     end
59 end
60
61 %% START iTH ITERATION
62 while change > 0.001
63     i = i + 1;
64     % Finding the next volume fraction for gradual evolution of
topology
65     if i >1;
66         volfrac=sum(sum(x))/(nelx*nely);
67         olddI = dI;
68         if volfrac>Vc
69             volfrac=max(volfrac*(1-er),Vc);
70         else
71             volfrac=min(volfrac*(1+er),Vc);
72         end
73     end
74
75     % FE-ANALYSIS
76     [U,Uj]=FE(nelx , nely , x , penal , load , fixed);
77
78     % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
79     [KE] = lk;
80     I(i) = 0.;
81     I0=0.217; % Maximum value of inertia
82     density=4.2e-7; %in kg/mm^3
83     m=density; % Volume of each element =lmm^3
84     for ely = 1:nely
85         for elx = 1:nelx
86             n1 = (nely+1)*(elx-1)+ely;
87             n2 = (nely+1)* elx +ely;
88             dI(ely , elx)=0;
89             Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*
n1+2],1);

```



```

90         Uej = Uj([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1
+1;2*n1+2],1);
91         Ie = m*(1/6+(elx-a1)^2+(ely-b1)^2); % Moment of inertia of
each element
92         I(i) = I(i) + x(ely,elx)^penal*Ie; % Total moment of
inertia
93         dI(ely,elx) = (x(ely,elx)^(penal-1)*Uej'*KE*Ue)/abs(Umax);
% Sensitivity number
94         end
95     end
96     I1(i)=I(i)/I0+lambda*(U(1742,1)-Umax)/abs(Umax); % New objective
function (normalized)
97
98     % FILTERING OF SENSITIVITIES
99     [dI] = check(nelx,nely,rmin,dI);
100
101     % SYMMETRICAL DESIGN UPDATE
102     [dI] = symmetry(nelx,nely,dI);
103
104     % STABILIZATION OF EVOLUTIONARY PROCESS
105     if i > 1;
106         dI = (dI+olddI)/2.;
107     end
108
109     % CORRESPONDING VOLUME FRACTION BASED ON THE DISPLACEMENT
CONSTRAINT
110     [Vc]=volume(nelx,nely,dI,x,U,Uj,Umax,passive,keep);
111
112     % BESO DESIGN UPDATE
113     [x] = ADDDEL(nelx,nely,dI,volfrac,passive,keep);
114
115     % CONVERGENCE CRITERION
116     if i>10;
117         change=abs(sum(I(i-9:i-5))-sum(I(i-4:i)))/sum(I(i-4:i));
118     end
119
120     % PRINT RESULTS
121     disp([' It.: ' sprintf('%4i',i) ' Obj.: ' sprintf('%10.4f',I1(i))
...
122         ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
123         ' ch.: ' sprintf('%6.3f',change )])
124
125     % PLOT DENSITIES
126     colormap(gray); imagesc(-x); axis equal; axis tight; axis off;
pause(1e-6);
127 end
128
129 %%%% FINDING Vc, CORRESPONDING VOLUME FRACTION %%%%%%%%%%
130 function [Vc]=volume(nelx,nely,dc,x,U,Uj,Umax,passive,keep)
131 l1 = min(min(dc));
132 l2 = max(max(dc));
133 j=0;
134 [KE] = lk;
135 penal=3;
136 while (l2-l1)/l2 > 1.0e-5
137     j=j+1;
138     du(j)=0;
139     xold=x;

```

```

140     th = (l1+l2)/2.0; % threshold of the sensitivity
141     x = max(0.001, sign(dc-th)); % Update the design based on the
threshold
142     x(find(passive))=0.001;
143     x(find(keep))=1;
144     for ely = 1:nely
145         for elx = 1:nelx
146             n1 = (nely+1)*(elx-1)+ely;
147             n2 = (nely+1)* elx +ely;
148             Uej = Uj([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1
+1;2*n1+2],1);
149             Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*
n1+2],1);
150             du(j)=du(j)-penal*x(ely , elx)^(penal-1)*Uej'*KE*Ue*(x(ely ,
elx)-xold(ely , elx));
151         end
152     end
153     Uk=max(U(1742,1)+du,Umax); % Finding the displacement in the next
iteration
154     % Checking the constraint and changing the interval of
sensitivities
155     if Uk > Umax;
156         l1 = th;
157     else l2 = th;
158     end
159 end
160 Vc=sum(sum(x))/(nelx*nely); % Corresponding volume fraction for the
new design
161
162 %%%%%%%%% UPDATE FOR THE NEXT ITERATION %%%%%%%%%
163 function [xnew]=ADDDEL(nelx , nely , dc , volfrac , passive , keep)
164 l1 = min(min(dc));
165 l2 = max(max(dc));
166 while (l2-l1)/l2 > 1.0e-5
167     th = (l1+l2)/2.0;
168     %if dc is greater than th, add element, else delete element.
169     xnew = max(0.001, sign(dc-th)); %Updating the design variables x
170     xnew(find(passive))=0.001;
171     xnew(find(keep))=1;
172     %Checking if volume fraction due to the new design < the target
volume
173     if sum(sum(xnew))-volfrac*(nelx*nely) > 0;
174         l1 = th;
175     else
176         l2 = th;
177     end
178 end
179
180 %%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%
181 function [dcf]=check(nelx , nely , rmin , dc)
182 dcf=zeros(nely , nelx);
183 for i = 1:nelx
184     for j = 1:nely
185         sum=0.0;
186         for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
187             for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
188                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
189                 sum = sum+max(0, fac);

```

```

190         dcf(j,i) = dcf(j,i) + max(0, fac)*dc(1,k);
191     end
192 end
193     dcf(j,i) = dcf(j,i)/sum;
194 end
195 end
196
197 %%%%%%%%%% SYMMETRY %%%%%%%%%%
198 function [dI] = symmetry(nelx, nely, dI)
199 for ely=1:nely
200     for elx=1:nelx
201         dI(ely, elx)=max(dI(ely, elx), dI(nely-ely+1, elx));
202     end
203 end
204
205 %%%%%%%%%% FE-ANALYSIS %%%%%%%%%%
206 function [U, Uj]=FE(nelx, nely, x, penal, load, fixed)
207 [KE] = lk;
208 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
209 F = sparse(2*(nely+1)*(nelx+1), 1); % Actual load
210 U = zeros(2*(nely+1)*(nelx+1), 1); % Displacement due to actual load
211 Fj = sparse(2*(nely+1)*(nelx+1), 1); % virtual load
212 Uj = zeros(2*(nely+1)*(nelx+1), 1); % displacement due to virtual load
213 for elx=1:nelx
214     for ely = 1:nely
215         n1 = (nely+1)*(elx-1)+ely;
216         n2 = (nely+1)* elx +ely;
217         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*
n1+2];
218         K(edof, edof) = K(edof, edof) + x(ely, elx)^penal*KE;
219     end
220 end
221
222 % DEFINE LOADS AND SUPPORTS
223 F(2*find(load), 1) = -0.25;
224 Fj(2*find(load), 1) = -0.25;
225 fixeddofs = union(2*find(fixed), 2*find(fixed)-1);
226 alldofs    = [1:2*(nely+1)*(nelx+1)];
227 freedofs   = setdiff(alldofs, fixeddofs);
228 % SOLVING
229 U(freedofs, :) = K(freedofs, freedofs) \ F(freedofs, :);
230 U(fixeddofs, :) = 0;
231 Uj(freedofs, :) = K(freedofs, freedofs) \ Fj(freedofs, :);
232 Uj(fixeddofs, :) = 0;
233
234 %%%%%%%%%% ELEMENT STIFFNESS MATRIX for YELLOW POPLAR (LR) %%%%%%%%%%
235 function [KE]=lk
236 E_L=1.1; %in GPa
237 E_R=0.092*E_L;
238 G_LR=0.075*E_L;
239 nu_LR=0.318;
240 KE = 1/2*[(2*G_LR)/3 + (2*E_L^2)/(3*(E_L - nu_LR^2*E_R)), G_LR/2 + (
nu_LR*E_L*E_R)/(2*(E_L - nu_LR^2*E_R)), G_LR/3 - (2*E_L^2)/(3*(E_L
- nu_LR^2*E_R)), (nu_LR*E_L*E_R)/(2*(E_L - nu_LR^2*E_R)) - G_LR
/2, - G_LR/3 - E_L^2/(3*(E_L - nu_LR^2*E_R)), - G_LR/2 - (nu_LR*
E_L*E_R)/(2*(E_L - nu_LR^2*E_R)), E_L^2/(3*(E_L - nu_LR^2*E_R)) -
(2*G_LR)/3, G_LR/2 - (nu_LR*E_L*E_R)/(2*(E_L - nu_LR^2*E_R))];

```


Chapter 6

Bibliography

- [1] Coralie Germain, Stéphane Caro, Sébastien Briot, Philippe Wenger. Singularity-free design of the translational parallel manipulator IRSbot-2. *Mechanism and Machine Theory*, (64):262–285, 2013.
- [2] Thierry Laurent, Jean-Louis Kergueme, Olivier Arnould, David Dureisseix. Eco-conception d’un bras de robot grande vitesse en bois 1. Méthode de conception et justification du choix du matériau. (French) [On the eco-design of a high speed robot arm: Method of design and justification of choice of material]. *Technologie*, 168:pp.28–36, 2010, hal-00537853.
- [3] Thierry Laurent, Jean-Louis Kergueme, Olivier Arnould, and David Dureisseix. Eco-conception d’un bras de robot grande vitesse en bois 2. Enseignement par projet en master ” Création Industrielle en Mécanique ”. *Technologie*, 169:24–35, 2010.
- [4] Philippe Martinet. Sensor based control : task function approach for visual servoing, 16-20 September, 2002.
- [5] Bendsøe, M.P.; Kikuchi, N. Generating optimal topologies in optimal design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.
- [6] Oscar Brito Augusto, Fouad Bennis and Stéphane Caro. Multiobjective engineering design optimization problems: A sensitivity analysis approach. *Pesquisa Operacional*, 32(3):575–596, 2012.
- [7] Wang W., Caro S., Bennis F., Soto R., and Crawford B.,. Multi-objective Robust Optimization using a Post-optimality Sensitivity Analysis Technique : Application to a Wind Turbine Design. *ASME Journal of Mechanical Design*, Vol. 137(DOI : 10.1115/1.4028755):pp. 011403–1–011403–11, 2015.
- [8] David V Hutton. *Fundamentals of Finite Element Analysis*. McGraw-Hill, Jun 1, 2003.
- [9] Wikimedia Commons. Components stress tensor cartesian, 12 March 2009.
- [10] College of San Francisco. Course handouts of composite materials, 2009.

-
- [11] Mike Xie Xiaodong Huang. *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*. John Wiley and sons, Australia, 2010.
- [12] General Technical Report FPL-GTR-190. *Wood Handbook, Wood as an engineering material*. United States Department of Agriculture Forest Service, Madison, Wisconsin, April 2010.
- [13] M.Maya, Ecole Nationale Supérieure d'arts et métiers. Le comportement mécanique du bois.
- [14] Professor Ilean Bonev. Parellemic: The parallel mechanisms information center, <http://www.parellemic.org/>, Accessed: 13 November 2014.
- [15] Jean Pierre Merlet. *Parallel Robots*. Hermes, 1996.
- [16] Accsys Technologies. Accoya, acetylated wood, <http://www.accoya.com/>, Accessed: 26 November 2014.
- [17] Maya Hage Hassan. Aide à l'optimisation de structures : modèles paramétrique, géométrique et topologique, Master's Bibliographic Report, IR-CCyN, March 2010.
- [18] G.Allaire. *Conception Optimale de structures*. Springer-Verlag, 2007.
- [19] Hinton E Hassani B. *Homogenization and structural topology optimization: theory, practice and software*. Springer, London, 1999.
- [20] Mlejnek HP, Schirmacher R . An engineering approach to optimal material distribution and shape finding. *Methods in Applied Mechanical Engineering*, 106:1–26, 1993.
- [21] Sigmund O. Design of material structures using topology optimization. *PhD thesis, Department of Solid Mechanics, Technical University of Denmark*, 1994.
- [22] Bendsoe, M.P. and Sigmund, O. Material interpolations in topology optimization. *Architectural Applied Mechanics*, 69(9-10):635–654, 1999.
- [23] Sigmund, O.; Petersson, J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16:68–75, 1998.
- [24] Rozvany G.I.N. A critical review of established methods of structural topology optimization. *Structural Multidisciplinary Optimization*, 37:217–237, 1992.
- [25] Bendsoe, M.P. Optimal shape design as a material distribution problem. *Structural Optimization*, 1:193–202, 1989.
- [26] Rozvany GIN, Zhou M, Birker T. Generalized shape optimization without homogenization. *Structural Optimization*, 4:250–254, 1992.
- [27] Sigmund O. A 99 line topology optimization code written in Matlab. *Structural Multidisciplinary Optimization*, 21:120–127, 2001a.
-

-
- [28] Svanberg, K. The method of moving asymptotes – a new method for structural optimization. *Numerical Methods in Engineering*, 24:359–373, 1987.
- [29] M.P. Bendsøe. *Optimization of structural topology, shape and material*. Springer, Berlin, Heidelberg, New York, 1995.
- [30] Lei Li, Kapil Khandelwal. Two-point gradient-based MMA (TGMMA) algorithm for topology optimization. *Computers and structures*, 131:34–45, 2014.
- [31] Xie, Y.M. and Steven, G.P. Shape and layout optimisation via an evolutionary procedure. *Proceedings of International Conference on Computational Engineering Science, Hong Kong*, page 471, 1992.
- [32] Xie, Y.M. and Steven, G.P. A simple evolutionary procedure for structural optimization. *Computers Structures*, 49:885–96, 1993.
- [33] Xie, Y.M. and Steven, G.P. Evolutionary structural optimization for dynamic problems. *Computers Structures*, 58:1967–73, 1996.
- [34] Gomes FAM, Senne TA. The method of moving asymptotes – a new method for structural optimization. *Computational and Applied Mathematics*, 30:53–89, 2011.
- [35] Sigmund, O. On the design of compliant mechanisms using topology optimization. *Mech. Struct. Mach.*, 25:495–526, 1997.
- [36] Kikuchi N, Nishiwaki S, Ono JSF, Silva ECS. Design optimization method for compliant mechanisms and material microstructure. *Computational and Applied Mathematics*, 151:401–417, 1998.
- [37] Etman LFP, Groenwold AA, Rooda JE. First-order sequential convex programming using approximate diagonal QP subproblems. *Structural and Multidisciplinary Optimization*, 45:479–488, 2012.
- [38] Groenwold AA, Etman LFP. A quadratic approximation for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 82:505–524, 2010.
- [39] Patel N., Mozumder C. Crashworthiness design using topology optimization. *PhD thesis, University of Notre Dame, IL, USA*, 2007.
- [40] Pedersen C.B.W. On topology design of frame structures for crashworthiness. *PhD thesis, Technical University of Denmark*, 2002.
- [41] Zhan Kang, Chi Zhang, Gengdong Cheng. Structural Topology Optimization Considering Mass Moment of Inertia. *6th World Congresses of Structural Multidisciplinary Optimization*, 30 May - 03 June 2005.
- [42] Katie Maurer. Industrial engineering course on robust design, February 11, 2000.
-

-
- [43] Davide Corradi, Stéphane Caro, Damien Chablat and Philippe Cardou. Assembly conditions of parallel manipulators considering geometric errors, joint clearances, link flexibility and joint elasticity. *IEEE International Conference on Robotics and automation*, pages 4067 – 4072, June 7 2014.
- [44] Stephen Boyd, Seung-Jean Kim, Lieven Vandenbergh, Arash Hassibi. A tutorial on geometric programming. *Optimization and Engineering*, 8:67–127, March 2007.
- [45] Gerald J. Liebermann Frederick S. Hillier. *Introduction to Operations Research, Seventh edition*. Stanford University, 2001.
- [46] Stephen Boyd, Lieven Vandenbergh. *Convex Optimization*. Cambridge University press, 2004.
- [47] Winker, P. and M. Gilli. Applications of optimization heuristics to estimation and modelling problems. *Computational Statistics Data Analysis*, (47):211–223, 2004.
- [48] Manfred Gilli. An Introduction to Optimization Heuristics, University of Cyprus.
- [49] Daryl L. Logan. *First Course in the Finite Element Method*. Chris Carson, University of Wisconsin, 2007.
- [50] Ahmed A. Shabana. *DYNAMICS OF Multibody Systems, Third edition*. Cambridge University press, 2005.
- [51] Inc ANSYS. *Theory Reference for the Mechanical APDL and MEchanical Applications*. Canonsburg, PA, April 2009.
- [52] X. Huang · Y. M. Xie. A further review of ESO type methods for topology optimization. *Structural Multidisciplinary Optimization*, 2009.
- [53] Chiandussi, G. On the solution of a minimum compliance topology optimization problem by optimality criteria without a priori volume constraint specification. *Computational Mechanics*, 38(1):77–79, 2006.
- [54] X. Huang · Y. M. Xie. Evolutionary topology optimization of continuum structures with an additional displacement constraint. *Structural Multidisciplinary Optimization*, 40:409–416, 9 April 2009.
- [55] Sigmund, O. and Petersson, J. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16:68–75, 1998.
- [56] M.P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Method and Application*. Springer, Berlin, 2003.
- [57] Huang, X. and Xie, Y.M. Convergent and mesh-independent solutions for bi-directional evolutionary structural optimization method. *Finite Elements in Analysis and Design*, 43(14):1039–49, 2007.
-