AN ALGORITHM FOR STRUCTURAL TOPOLOGY OPTIMIZATION OF MULTIBODY SYSTEMS

TOHEED GHANDRIZ

Master’s thesis
2014:E52

Lund University
Faculty of Science
Centre for Mathematical Sciences
Numerical Analysis
Master's Theses in Mathematical Sciences 2014:E52
ISSN 1404-6342
LUNFNA-3018-2014
Numerical Analysis
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden
http://www.maths.lth.se/
An Algorithm for Structural Topology Optimization of Multibody Systems

Toheed Ghandriz

Department of Numerical Analysis
Lund University
2014

Supervisors:
Hilding Elmqvist, CATIA Systemes/Dymola R&D Technology VP, Dassault Systemes AB
Claus Führer, Professor in Numerical Analysis, Lund University
Dan Henriksson, CATIA Systemes/Dymola R&D Technology Senior manager, Dassault Systemes AB
Abstract

Topology Optimization (TO) of static structures with fixed loading is a very interesting topic in structural mechanics that has found many applications in industrial design tasks. The extension of the theory to dynamic loading for designing a Multibody System (MBS) with bodies which are lighter and stronger can be of high interest. The objective of this thesis work is to investigate one of the possible ways of extending the theory of the static structural Topology Optimization to Topology Optimization of dynamical bodies embedded in a Multibody System (TOMBS) with large rotational and transitional motion. The TOMBS is performed for all flexible bodies simultaneously based on the overall system dynamical response. Simulation of the MBS behavior is done using the finite element formalism and modal reduction. A modified formulation of Solid Isometric Material with Penalization (SIMP) method is suggested to avoid numerical instabilities and non-convergence of the optimization algorithm implemented for TOMBS. The nonlinear differential algebraic equation of motion is solved numerically using Backward Differential Formula (BDF) with variable step size in SundialsTB and Assimulo integrators implemented in Matlab and Python. The approach can find many applications in designing vehicle systems, high-speed robotic manipulators, airplanes and space structures. Also, to show the current capability of the tools in the industry to design a body under dynamic loading using the multiple static load cases, the lower A-arm of double wishbone suspension system is designed in Abaqus/TOSCA, where, the loads are collected from rigid multibody simulation in Dymola.

Keywords

Topology Optimization of Multibody Systems, dynamic body, SIMP, flexible body, finite element method, nonlinear Differential Algebraic Equation, Abaqus/TOSCA, Dymola, flying element
Acknowledgment

I would like to express my gratitude to my supervisors Claus Führer, Hilding Elmqvist, and Dan Henriksson for their useful comments, support and motivation. Furthermore I would like to thank Sven Erik Mattsson and all my colleagues at CATIA Systemes/Dymola R&D Technology, Dassault system, for the support on the way. I would like to give a special thanks to my loved ones, my mother, my siblings and my wife, Fatemeh, who stood by me and helped me throughout entire process. Finally, this work is dedicated to the memory of my father, Abdollah Ghandriz, who inspired engineering thinking in my life.
Table of Contents

1. Introduction .................................................................................................................................................. 12

2. Background .................................................................................................................................................. 17
   2.1. Structural Optimization problem definition for static bodies ......................................................... 17
   2.2. Finite Element model of linear elasticity ............................................................................................ 18
      2.2.1. The strong form of the governing equation, equilibrium equation .......................................... 19
      2.2.2. Transforming the strong form into the weak form .................................................................... 21
      2.2.3. Meshing the solution domain, choosing element type and shape function .............................. 21
      2.2.4. Choosing weight functions and establishing the system of algebraic equations for an element ... 24
      2.2.5. Assembly the element algebraic equations into the global system of algebraic equations ......... 25
      2.2.6. Introducing boundary conditions ................................................................................................. 26
      2.2.7. Solving the equations ................................................................................................................... 27
   2.3. Solving the optimization problem ........................................................................................................ 28
      2.3.1. Approximating the objective function ......................................................................................... 28
      2.3.2. Minimization of Lagrangian function ......................................................................................... 31
      2.3.3. SIMP method ............................................................................................................................. 33
      2.3.4. Filters ......................................................................................................................................... 34
   2.4. Flexible Multibody Dynamics ................................................................................................................ 35
      2.4.1. position vector .............................................................................................................................. 35
      2.4.2. Floating Frame of Reference ....................................................................................................... 36
      2.4.3. Position of an arbitrary point of a flexible body in the space .................................................... 37
      2.4.4. Absolute velocity of an arbitrary point of a flexible body ............................................................ 38
      2.4.5. Equation of Motion of bodies in a Multibody system .................................................................. 39
      2.4.6. Mass matrix of a flexible body ..................................................................................................... 40
         2.4.6.1. Calculating mass matrix Integrals for planar motion ............................................................ 41
         2.4.6.2. Calculating Inertia shape Integrals ....................................................................................... 43
      2.4.7. Body stiffness matrix .................................................................................................................... 44
      2.4.8. Generalized forces ......................................................................................................................... 44
         2.4.8.1. Externally applied forces ....................................................................................................... 44
         2.4.8.2. Spring-Damper-Actuator element ......................................................................................... 45
      2.4.9. Kinematical constraints ................................................................................................................ 47
         2.4.9.1. Revolute joint ....................................................................................................................... 47
         2.4.9.2. Trajectory constraint ............................................................................................................ 48
      2.4.10. Quadratic velocity ....................................................................................................................... 49
      2.4.11. Solving the Equation of Motion ................................................................................................... 49
         2.4.11.1. Choosing a proper element type ......................................................................................... 49
         2.4.11.2. Reference Condition ......................................................................................................... 49
         2.4.11.3. Solving DAE numerically using Sundials solver ................................................................. 50
         2.4.11.4. DAE Index .......................................................................................................................... 51
2.4.12. Modal reduction........................................................................................................52

3. Design of the Lower A-arm of Double Wishbone suspension system in Abaqus/TOSCA using Dymola ..... 54
   3.1. topology optimization of a body subjected to transient loads ...........................................54
   3.2. Topology optimization of the Lower A-arm of Double Wishbone suspension system .... 56
      3.2.1. Choosing the proper load cases coming from dynamical simulation of VDL ............ 56
      3.2.2. Defining the geometry of the first design space .........................................................57
      3.2.3. Defining the material properties and meshing the design space ...............................58
      3.2.4. Defining sets, constraints, loads and boundary conditions .....................................58
      3.2.5. Defining the optimization task and running topology optimization ..........................60
      3.2.6. Converting the result to a real part (verification model) ..........................................62
      3.2.7. Making an approximated part based on the available drawing commands ................63
      3.2.8. Comments on the method .......................................................................................64

4. Topology Optimization of Multibody Systems ......................................................................65
   4.1. Defining and approximating the optimization problem ..................................................65
   4.2. Sensitivity analysis ........................................................................................................67
   4.3. OC approximation of the compliance of a dynamic body .............................................70
   4.4. Modification to SIMP method in TOMBS ..................................................................71
   4.5. Convergence criterion ...............................................................................................73
   4.6. Some comments on the method ..................................................................................74

5. Examples of Topology Optimization of Multibody Systems ................................................75
   5.1. Slider Crank ................................................................................................................76
      5.1.1. Testing TOMBS on a Slider Crank system ..............................................................76
      5.1.2. Topology optimization of the connecting rod using important static load cases ....... 80
      5.1.3. The effect of number of considered modes on MBSTO ...........................................83
   5.2. Nowait train .................................................................................................................84
   5.3. A seven-body MBS .....................................................................................................85

6. The structure of the code ..................................................................................................88
   6.1. Defining the TOMBS problem .......................................................................................89
      6.1.1. Defining a body ......................................................................................................89
      6.1.2. Defining the joints, constraints and external forces ................................................90
      6.1.3. Defining the solution method ...............................................................................91
      6.1.4. Defining a consistent initial condition ..................................................................91
   6.2. Reducing the computation time ..................................................................................92

7. Conclusion and future works ...........................................................................................93

References ............................................................................................................................96

Appendix .............................................................................................................................98
Notations

$A$ transformation or rotation matrix
$A_{	heta_k}$ differential of the rotation matrix with respect to the Euler parameters
$A$ total area of the two dimensional body
$A_e$ the area of the element
$a_k^e$ the vector of vibration amplitudes of elastic DoFs which vibrate with frequency $\omega_k$
$a$ half of the element length
$a_e$ the area corresponding to $X_e$

$B$ the differential of the shape function
$\bar{B}$ defined in (2.86)
$B_{im}^k$ a matrix whose columns are the $n_m$ low frequency eigenvectors
$b$ the body force per unit volume
$b$ half of the element width
$b_e^k$ a constant associated to the $e^{th}$ design variable at iteration $k$

$C$ the vector of kinematic algebraic constraint equations
$C_j^i$ the partial derivative of the vector of constraint functions with respect to $i$
$C_q^i$ the transpose of constraints Jacobian matrix where the differentiation of constraints equation is done with respect to $q^i$
$C$ the objective function or compliance
$C_j^i$ the objective function corresponding to load case $i$
$C_j^{l,k}$ the objective function or the compliance of the body $j$ at time $t_l$
$C_j^k$ the objective function of the body $j$ at iteration $k$
$C_j^l$ the objective function or the compliance of the body $j$
$C_{j,k,OC}^k$ OC approximation of the objective function of the body $j$ at iteration $k$
$c$ the damping coefficient

$D$ the constitutive matrix
$(D)^k$ dual problem at iteration $k$

$E$ Young's modulus
$E_j^i$ the initial Young's modulus of the body $j$
$E_j^p$ the penalized Young's modulus of the body $j$
e the index for the design variables

$F$ the global external force
$F_p^i$ Pseudo load
$F_j^{l,t}$ the all forces including the inertia forces acting on the nodal points of the body $j$ at time $t_l$
$f_e$ the element load vector
$F_a$ the actuator force
$F_{SDA}$ the SDA force vector

$g$ a set of inequality constraints
$g_0$ the objective function
the constraint function

\( g_j \)

the constraint function associated to the body \( j \)

\( g_j' \)

\( g_{0,\text{OC}}^k \)

OC approximation of the objective function at iteration \( k \)

\( g_0(\bar{X}) \)

the objective function after applying the mean sensitivity filter

\( h \)

a set of equilibrium constraint equations

\( h_e \)

the thickness of the element

\( h_a \)

the time step

\( h_0 \)

the initial thickness of element

\( I \)

a \( 3 \times 3 \) identity matrix in three dimensions and \( 2 \times 2 \) in two dimensions

\( I_{i,j}^l \)

inertia shape integral defined in (2.102) or (2.109) or the moment of mass around axes of undeformed body coordinate system

\( I_{i,j}^l \)

inertia shape integral defined in (2.102) or (2.109)

\( I_{i,j}^l \)

inertia shape integral defined in (2.102) or (2.109)

\( i \)

body index

\( j \)

index showing the body number

\( K \)

the global stiffness matrix

\( K^i \)

the body stiffness matrix associated with generalized coordinates \( q^i \)

\( K^i_{ij} \)

the global stiffness matrix of body \( i \) associated with elastic coordinates

\( k \)

the spring constant

\( k_e \)

the element stiffness matrix

\( k_e^i \)

the element local normalized or specific stiffness matrix

\( k \)

index of optimization iteration

\( L \)

defined in (2.90)

\( L_{ij}^j \)

the expression given in (2.90) but for body \( j \) and point \( A \)

\( L_{ij}^H \)

the expression given in (2.90) but for body \( j \) and point \( H^l \)

\( L_{ij}^P \)

the expression given in (2.90) but for body \( j \) and point \( P^i \)

\( l \)

a vector from point \( P^i \) to point \( P^i \)

\( l \)

the unit of the vector

\( l_{ij}^G \)

the unit of the vector pointing from \( H^l \) to \( G \)

\( l_{ij}^{P^i} \)

the unit of the vector pointing from \( P^i \) to \( P^i \)

\( L \)

the total number of objective functions

\( l \)

the current length of the spring

\( l_p \)

the outer line boundaries of the body

\( l_0 \)

undeformed length of spring

\( L^k \)

the Lagrangian function at iteration \( k \)

\( L^E \)

the Lagrangian function of the \( e^{th} \) design variable at iteration \( k \)

\( M^i \)

the symmetric mass matrix of body \( i \)

\( m_{i,R}^\text{st} \)

a partition of the mass matrix associated with translational coordinates

\( m_{i,R}^\text{cst} \)

a partition of the mass matrix associated with coupled translational and rotational coordinates

\( m_{i,f}^\text{cst} \)

a partition of the mass matrix associated with coupled translational and elastic coordinates

\( m_{i,\theta}^\text{cst} \)

a partition of the mass matrix associated with the rotational coordinates

\( m_{i,\theta}^\text{cst} \)

a partition of the mass matrix associated with coupled rotational and elastic coordinates
\( m_{ij} \) a partition of the mass matrix associated with elastic coordinates
\( M_j^e \) effective mass of the element \( e \) of the body \( j \)
\( M_j^{e,0} \) the mass of the element at the first iteration of the body \( j \)
\( m \) total number of Elastic Degrees of Freedoms or nodes of the body
\( m^{ij} \) the mass at grid point \( j \) of body \( i \)
\( N^e \) element local shape function
\( N_e \) a set containing the number of elements around the element \( e \) within the given radius
\( n \) the normal vector to the external surface
\( n \) the total number of the design variables or the number of finite elements
\( n_b \) the total number of bodies in MBS
\( n_c \) the total number of constraints
\( n_f \) the total number of the elastic DoFs of the deformable body \( i \)
\( n_j \) the total number of grid points where the lumped masses are distributed
\( n_m \) the number of the lowest eigenvalues used for modal reduction
\( n_r \) the total number of the rotational coordinates
\( n_x \) number of design variables
\( n_z \) number of state variables
\( O \) the origin of the global coordinate system attached to the ground
\( O' \) the origin of the body coordinate system
\( \mathcal{P} \) optimization problem
\( P \) an arbitrary point on the body
\( p_j^e \) the modal elastic coordinates
\( Q_i^e \) the vector of generalized forces associated with generalized coordinates of body \( i \)
\( Q_i^s \) the quadratic velocity vector
\( Q_i^\theta \) quadratic velocity vector associated with the elastic coordinates
\( Q_i^\phi \) quadratic velocity vector associated with the transitional coordinates
\( Q_i^\psi \) quadratic velocity vector associated with the rotational coordinates
\( q \) the vector of generalized coordinates
\( q_i^e \) the displacement vector of all nodal points or vector of elastic coordinates defined relative to body system of coordinates
\( q_i^e \) the element nodal displacement vector with respect to the body coordinate system
\( \dot{q}^i \) the total vector of generalized accelerations
\( q_i^e \) a trial solution of the elastic coordinates of the body \( i \) for the homogenous part of EoM
\( q_j^e \) the elastic DoFs of the body \( j \)
\( q_j^e \) the vector of the nodal displacements of the body \( j \) at time \( t_i \)
\( q_e \) the vector of reference coordinates
\( \dot{q}^i \) the time derivative (velocity) of the generalized coordinates
\( Q \) the order of BDF method which ranges from one to five
\( q \) penalization factor in SIMP
\( R_{o'} \) a vector giving the position of \( O' \), translation coordinates, with respect to the global system of coordinates
\( R_p \) a vector giving the position of an arbitrary point \( P \) on the body with respect to the global system of coordinates
\( RF \) the residual function
\( [R_1^e \ R_2^e] \) a vector which defines the position of the origin of the body coordinate system
$r$ a vector form $O'$ to $P$ measured with respect to the global system of coordinates

$S$ the global shape function

$S_i$ the stress tensor

$\bar{S}_i$ inertia shape integral defined in (2.102) or (2.109)

$S_k^i$ the $k^{th}$ row of the global shape function

$S_{ij}^i$ the global element shape function at grid point $j$ of body $i$

$S_k^i$ the $k^{th}$ row of $S_{ij}^i$

$\bar{S}_k^{ij}$ inertia shape integral defined in (2.102) or (2.109)

$S$ the external surface of the body

$S_y$ a part of the external surface of the body where the displacement is prescribed

$S_h$ a part of the external surface of the body where the traction force acts

$s$ the number of the communication points

$t$ the traction force

$T$ transpose

$T_i^i$ the kinetic energy of body $i$

$t$ time

$t_0$ the initial time

$t_l$ the time at step $l$

$t_n$ the time at step $n$

$t_S$ the stopping simulation time of MBS

$U$ the global displacement vector

$u$ the global displacement of the body

$u_b^i$ the position of point $P$ before deformation with respect to the body system of coordinates

$u_{ij}^i$ a vector giving the position of an arbitrary point $P$ on the body with respect to the body system of coordinates

$u_{0ij}^i$ the vector describing the undeformed position of grid point $j$ in body $i$ with respect to the body system of coordinates

$u_l^i$ the displacement vector of point $P$ with respect to the body system of coordinates

$u_x$ the displacement in $x$ direction

$V$ the volume of the body

$V_i^i$ the volume of the body $i$

$V_{max}$ maximum allowable volume

$\nu$ Poisson's ratio

$w$ a weight function

$w_l$ a weight factor

$X$ the vector of design variables

$X^j$ the vector of the design variables of the body $j$

$X^{l,k}$ the vector of design variables of the body $j$ at iteration $k$

$X_k$ the design variable at iteration $k$

$X_{max}$ the upper bounds of the design variable

$X_{min}$ the lower bound of the design variable

$X^*$ the design variable which minimizes the Lagangian function

$x$ global coordinate system

$x' y' z'$ body coordinate system or the body Floating Frame of Reference

$X_e$ the $e^{th}$ design variable
\( X_{e}^{j} \) the design variable of element \( e \) of the body \( j \)
\( X_{e}^{k} \) the \( e^{th} \) design variable at iteration \( k \)
\( \chi_{e}^{\text{trial}} \) the \( e^{th} \) design variable where \( L_{e}^{k} \) is stationary
\([x_{1}^{ij}, x_{2}^{ij}]\) a vector describing the undeformed position of a point in body \( i \) with respect to the body system of coordinates
\([x_{1}^{ij}, x_{2}^{ij}]\) the vector describing the undeformed position of grid point \( j \) in body \( i \) with respect to the body system of coordinates

\( Y \) the intermediate variable
\( Y_{e} \) the \( e^{th} \) intermediate variable
\( y \) the unknown variable in DAE
\( y_{0} \) the initial values of the unknown variable in DAE

\( Z \) the vector of state variables

\( \alpha \) OC method constant
\( \alpha_{n,i} \) a function of the step size history and the method order in BDF
\( \nabla \) a matrix of differential operator
\( \delta W_{e}^{i} \) the virtual change in all external forces acting on body \( i \)
\( \delta q^{i} \) the virtual change in generalized coordinates
\( \varepsilon \) the strain vector
\( \theta \) the rotation angle of the body around the unit axis \( \nu \)
\( \theta^{i} \) set of Euler parameters
\( \theta^{i} \) the angle of rotation of body \( i \) around \( z \) axis
\( \lambda \) Lagrange multiplier (used in Lagrangian Duality)
\( \lambda \) vector of Lagrange multipliers (used in DAE)
\( \lambda^{*} \) which maximizes the Dual function
\( \rho^{i} \) the density of the body \( i \)
\( \sigma \) the stress vector
\( \sigma_{xy} \) the stress component acting on \( yz \) plane and in \( y \) direction
\( \sigma_{yz} \) the stress component acting on \( yz \) plane and in \( z \) direction
\( \phi_{e}^{i} \) the Dual objective function at iteration \( k \)
\( \chi \) a convex set
\( \omega_{k} \) the vibration frequency, \( k = 1, 2, \ldots, n_{f} \)

\((\cdot)\) differentiation with respect to time
\((\ldots)^{T}\) transpose
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC</td>
<td>Boundary Condition</td>
</tr>
<tr>
<td>BDF</td>
<td>Backward Differentiation Formula</td>
</tr>
<tr>
<td>CONLIN</td>
<td>Convex Linearization</td>
</tr>
<tr>
<td>DAE</td>
<td>Differential Algebraic Equation</td>
</tr>
<tr>
<td>DBM</td>
<td>Dynamic Behavior Modeling</td>
</tr>
<tr>
<td>DoF</td>
<td>Degrees of Freedom</td>
</tr>
<tr>
<td>ESLM</td>
<td>Equivalent Static Loads Method</td>
</tr>
<tr>
<td>EoM</td>
<td>Equation of Motion</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite Element Method</td>
</tr>
<tr>
<td>FFR</td>
<td>Floating Frame of Reference</td>
</tr>
<tr>
<td>MBS</td>
<td>Multibody System</td>
</tr>
<tr>
<td>OC</td>
<td>Optimality Criteria</td>
</tr>
<tr>
<td>RC</td>
<td>Reference Condition</td>
</tr>
<tr>
<td>SDA</td>
<td>Spring-Damper-Actuator</td>
</tr>
<tr>
<td>SIMP</td>
<td>Solid Isometric Material with Penalization</td>
</tr>
<tr>
<td>TO</td>
<td>Topology Optimization</td>
</tr>
<tr>
<td>TOMBS</td>
<td>Topology Optimization of a Multibody System</td>
</tr>
<tr>
<td>VDL</td>
<td>Vehicle Dynamics Library</td>
</tr>
</tbody>
</table>
1. Introduction

In the design process of a product the aim is to enhance the functionality, economy or esthetics. When designing a mechanical structure where the main task is carrying a load, the economical product is obtained if the same functionality can be achieved by using a lighter structure, so material is saved. Furthermore, the product has a ‘better’ design if the functionality is also improved. For example in a vehicle or an airplane lighter parts reduce the overall weight and then also the functionality is enhanced. Mathematically, a best design means an optimized one. Optimality is characterized by minimizing or maximizing an objective function which represents the quality of the structure. For a mechanical product the objective can be to maximize the strength or stiffness; however, it cannot be done without introducing constraints; such as weight. The problem must be formulated as obtaining the highest strength for a limited amount of the material. An example of such an optimization is shown in Fig. 1.1, where, the initial model is enhanced to a lighter and stronger design.

Fig. 1.1: (left) initial design; (right) design based on Topology Optimization.

Structural optimization is about finding the best design, where, the main task is carrying a load. A design is described by the material distribution in an available space. Structural optimization has three branches: size optimization, shape optimization and Topology Optimization (TO). TO is a part of conceptual design of a product; whereas, the other two are parts of detail design. In TO a model is initially a box called design space, where the material can be distributed. The design space is discretized by small finite elements. By assigning a design variable to each finite element which might be interpreted as an ‘existence’ variable, optimized state can be reached by removing the material iteratively from the design space while the objective function is minimized and constraints are satisfied. In most TO problems the goal is to minimize the deformation or flexibility of the structure subjected to a static load. This can be achieved by maximizing the stiffness or alternatively minimizing the strain energy stored in the structure that is referred to compliance while there is a given maximum amount of the material that must be used. In this case, the optimized structure is the stiffest one which shows the least deformation as a response to the static load compared to any other non-optimized design with the same weight distributed in the design space.

Structural optimization concerns static loads and optimization is done on a body isolated from the rest of the structure. When a load is applied to a body, the body does not respond instantaneously. It takes some time for the body to reach the rest. If the load is transient, the body response changes accordingly before reaching the rest, making the response nonlinear and time dependent. One strategy to do topology optimization on such a body under transient loads is called component-based approach [23]. In the component-based approach multiple static loads are selected from the dynamic loads acting on the isolated body. It means that it is assumed there is enough
time for the body to rest before the load changes. However, this assumption is not realistic for several reasons. The first is that the inertia forces are neglected which might be important in parts with high accelerations; the second, since the shape and the weight of the body change in every optimization iteration, in case if the transient loads depend on the design for instance in a Multibody System (MBS), the dynamic behavior and forces at joints change accordingly, hence the load cases are not valid anymore; the third, selecting the proper load cases is difficult; the forth, the optimization process is not automated. The same argument can be found in [20]. As an example of using such a method the design process of a body within an MBS, the lower A-arm of a double wishbone suspension system is discussed in this thesis work (see Chapter 3). The difficulties and the level of the approximations made for implementing such a method were motivations to investigate and suggest another more systematic approach.

One other strategy for structural optimization of bodies under dynamic and transient loads is called Equivalent Static Loads Method (ESLM) [12, 21, 22]. In this approach also the body is isolated from the rest of the system where all forces including the inertia forces are accounted for. A set of equivalent quasi-static load cases in every time step must be defined which produces the same displacement field as the one by dynamic loads. Then it would be possible to use the theory of the static structural optimization directly. However, this method is developed mostly for size and shape optimization [20, 21]. The use of the method for topology optimization results in non-convergence and numerical instabilities in most of the cases. There were attempts to use the method also for topology optimization in [21, 22] by removing the unstable elements from the design domain. However, this approach limits the design domain where the element may revail during optimization process in later iterations; moreover, the criterion used for element removal is not clearly defined; thus, the optimized design might be questionable. In other words, the final topology is not the optimal one. According to [21]: “the final results of topology optimization are not mathematically rigorous compared to size and shape optimizations”. The other point is that in ESLM approach the constraints or objective function cannot be defined for overall system behavior [20].

A more systematic approach is to do topology optimization for all flexible bodies simultaneously while they are operating in an MBS based on the system overall response considering all transient reaction forces as well as inertia forces acting on the bodies during the operation time. In this thesis work this approach is called Topology Optimization of a Multibody System (TOMBS). The dynamic topology optimization is not a trivial extension of the static topology optimization. In [15] almost the same approach is used with two different regime of stiffness penalization. The switching criteria between two regimes might be different for different problems, thus the presented formulation in [15] is not always valid. [7, 20] suggests numerical calculation of sensitivity using generalized alpha method; the example used in [7, 20] is size optimization with low number of design variables; so the problem of non-convergence which is observed only in topology optimization does not occur. Also in [9] only an example of shape optimization is presented. In general, no systematic approach is developed yet in literature to handle the problem of instability and non-convergence of topology optimization of the structure under dynamic loadings or topology optimization of multibody systems. In this thesis work it is tried to find the origin of the non-convergence and then the traditional Solid Isometric Material with Penalization (SIMP) method used in topology optimization based on static response is modified. The modification is needed to handle the non-convergence of topology optimization process based on dynamic response. This is done through not only the stiffness penalization but also the element or lumped mass penalization.

The idea of TOMBS is first, to solve the nonlinear differential algebraic equation of motion of MBS at each optimization iteration with reduced coordinates; then to retain the physical real coordinates of the flexible bodies;
after that, to approximate the sensitivities and minimize the objective function at iteration $k$ using the modified SIMP method; Finally, update the design variables and solve the EoM again but with the updated design variables. This process must be repeated until the convergence criterion is satisfied. Direct applying the traditional Solid Isometric Material with Penalization (SIMP) method to TOMBS results in numerical difficulties and non-convergence. Moreover, calculating the sensitivity numerically is expensive. In this thesis work it is explained how to overcome these problems. A modification to SIMP is proposed here that helps to get convergence in any TOMBS problem. In addition, sensitivity expressions are approximated by eliminating terms which have low order of magnitude but numerically expensive to calculate; so, the optimal design can be found in a reasonable computation time in a problem with large number of design variables. Finally, some examples of TOMBS are presented to demonstrate the method performance. It is shown that TOMBS gives a better design with less deflection than the TO design based on the multiple static loads. This approach is described in detail in following chapters. A result of such an approach is shown in Fig. 1.2.

Fig. 1.2: (top) TOMBS performed on a simple slider crank system, total number of design variables = 17500; (bottom) TOMBS on a seven body MBS, total number of design variables = 31900; see Chapter 5.

Fig. 1.3 illustrates the general procedure of performing TOMBS suggested in this thesis work. Each of the steps in Fig. 1.3 is explained in the following chapters.
A code in this thesis work is developed for building the flexible multibody system and performing TOMBS. In Chapter 2, the background theory of topology optimization based on static response is given which is the basis for developing the theory of TOMBS in Chapter 4. Also, the basics of finite element formalism implemented in TOMBS as well as the theory behind the flexible multibody system based on floating frame of reference approach are explained. Some of the approximations and numerical problems in TOMBS inherit from the approximations and numerical problems occurred in flexible MBS simulation. Having an insight to the theory of flexible MBS together with an independent code helps better understanding the origin of the numerical problems.

To show the current capability of the tools in the industry to design a body under dynamic loading using the multiple static load cases, the lower A-arm of Double Wishbone suspension system is designed in Abaqus/TOSCA, where, the loads are collected from rigid multibody simulation in Dymola, see Chapter 3. The level of approximations can be a motivation to try TOMBS.

In Chapter 4 the basic idea of TOMBS is developed. Topology optimization is done based on direct dynamic response of the system. The sensitivity analysis is performed. Also the effect of flying element, the term proposed in this thesis work, which is the origin of the non-convergence of optimization process is explained. The mesh is distorted during optimization iterations resulting in non-convergence. The best way for avoiding this effect proposed in this thesis work is penalization of mass with no element removal. The effect of flying elements is reduced by element mass penalization, but it is not a prefect penalization, since what is seen by the nonlinear

Fig. 1.3: Illustration of the general procedure of performing TOMBS.
dynamical equation of motion is lumped masses not element masses. A way of penalizing the lumped massed also is proposed in this thesis work which shows better convergence.

In Chapter 5 three examples of TOMBS are provided. It is shown that in addition to the parameters in static TO, the final optimized topology in TOMBS is sensitive to other parameters such as the simulation time and number of modes used in modal reduction. Also, it is shown that the optimized design based on TOMBS gives stiffer structure than the design obtained using component-based approach.

Finally, in Chapter 6 a short description of the code structure is provided.
2. Background

2.1. Structural Optimization problem definition for static bodies

A structural optimization problem is about finding the best design for the structure. Being the best is described by minimizing an objective function in discretized space, \( g_0(X, Z) \), \( g_0 : \mathbb{R}^{n_x \times n_z} \rightarrow \mathbb{R} \), which represents the quality of the design. Examples of objective functions in structural optimization are weight, displacement, stress, volume and strain energy. The objective function is a function of design variable \( X \) and state variable \( Z \). Design variable represents the design. Examples are the geometry, the normalized density of the material or the normalized thickness. Given a design variable the state variables indicate the response of the structure. Similar to the objective function a state can be the structure displacement, deformation, internal forces, volume, strain or stress.

Topology Optimization (TO) is a branch of the Structural Optimization where the goal is to find the optimized material distribution of a body in an available space called design space such that the objective function is minimized and constraints are satisfied. In TO the design variable mostly is chosen to be the normalized density of the material or the normalized thickness. In TO a design is defined by the topology of the material in the design space.

In order for a general optimization problem to have a solution some inequality constraints on the design and state variables must be defined; such a constraint on a state variable and on the design variable is called Behavioral Constraint and Design Constraint, respectively. There is another type of constraint relating the state and design variables that comes from the physics of the problem. In structural optimization it is called equilibrium constraint that is an equality constraint; thus, state variables can be eliminated in objective function through equilibrium constraint.

The mathematical form of the optimization problem is written as:

\[
\begin{align*}
\min_{X,Z} & \quad g_0(X, Z) \\
\text{s.t.} & \quad h(X, Z) = 0 \\
& \quad g(X, Z) \leq 0
\end{align*}
\]

(2.1)

where, (\( \mathcal{P} \)) stands for the optimization Problem, \( g_0(X, Z) \) is the scalar objective function, \( h(X, Z) = 0 \) is the equilibrium constraint, \( g(X, Z) \leq 0 \) is the inequality constraints and (s. t.) stands for ‘subject to’. The bold letters represent vectors.

The main goal of most structural optimization problems is to minimize the deformation or flexibility of the structure subject to the volume constraint and with the lower and upper bounds of the design variable. This can be achieved by maximizing the stiffness or alternatively minimizing the strain energy stored in the structure that is equivalent to minimization of the compliance, which in discretized space is defined by \( F^T \cdot u \), where \( F \) is the global external force and \( u \) is the global displacement of the body that is a state variable. Another measure of the stiffness is the magnitude of the displacement; but it is not a good function to be minimized since it can be a non-convex function of the design variable, whereas the compliance is a convex function [1]. The general form of the optimization problem of the compliance subject to the volume constraint and with the lower and upper bounds of the design variable, in discretized space, is written as follows:
\[
\begin{align*}
\min_{X, \mathbf{u}} & \quad g_0(X, \mathbf{u}) = F^T(X) \cdot \mathbf{u} \\
\text{s. t.} & \quad K(X) \mathbf{u} = F(X) \\
& \quad g_1(X) = \int_A X \, da - V_{\text{max}} \leq 0 \\
& \quad X \in \chi = \{X^{\text{min}} \leq X \leq X^{\text{max}}\}
\end{align*}
\]  

(2.2)

where, \(g_0\) is the objective function, \(X\) is the design variable, for two dimensional problems \(X\) is the normalized thickness, \(K\) is the stiffness matrix, \(g_1\) is the constraint function, \(A\) is the total area of the body, \(\chi\) denotes a convex set, \(V_{\text{max}}\) is the maximum allowable volume and \(X^{\text{min}}\) and \(X^{\text{max}}\) are the lower and upper bounds of the design variable which also says to be box constraint. Bold letters illustrate column vectors and superscript \(T\) stands for the transpose of the column vector.

In order to be able to solve the above problem more efficiently the objective function must only be a function of the design variable. The state variable in the objective function can be written in terms of the design variable by solving the equilibrium constraint. It can be done by discretizing the space and writing the equilibrium constraint \(K(X)\mathbf{u} = F(X)\) in the form \(\mathbf{u}(X) = K^{-1}(X)F(X)\) using Finite Element Method (FEM). All variables are discretized. Here it is assumed that the material is linear elastic, homogenous and isotropic, hence, the stiffness matrix is invertible, symmetric and positive definite. Now it is possible to write the objective function as a function of only the design variable \(X\). The optimization problem becomes:

\[
\begin{align*}
\min_{X} & \quad g_0(X) = F^T(X) \cdot \mathbf{u}(X) \\
\text{s. t.} & \quad g_1(X) = \sum_{e=1}^{n} a_e X_e - V_{\text{max}} \leq 0 \\
& \quad X \in \chi = \{X \in \mathbb{R}^n, X_e^{\text{min}} \leq X_e \leq X_e^{\text{max}}, e = 1, \ldots, n\}
\end{align*}
\]  

(2.3)

where, \(n\) is the total number of the design variables after discretizing which is the same as the number of finite elements, \(e\) is the index for the design variables, \(X_e\) is the \(e^{th}\) design variable, \(a_e\) is the area corresponding to \(X_e\) and \(X_e^{\text{min}}\) and \(X_e^{\text{max}}\) are the lower and upper bounds of the \(e^{th}\) design variable.

### 2.2. Finite Element model of linear elasticity

To find the objective function in (2.3) as a function of only the design variable it is needed to find the displacement with respect to the design variable. It can be done employing Finite Element Method (FEM). The reader is referred to [3] for in depth understanding of FEM. In the following, the steps towards Finite Element formulation of linear elasticity are shortly described.
2.2.1. The strong form of the governing equation, equilibrium equation

According to the theory of linear elasticity and Hooke's generalized law the stress and strain of a linear elastic material are related through the constitutive relation:

\[ \sigma = D \varepsilon \]  \hspace{1cm} (2.4)

where,

\[ \sigma^T = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{zz} & \sigma_{xz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{yz} \end{bmatrix} \]  \hspace{1cm} (2.5)

\[ \varepsilon^T = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{zz} & \varepsilon_{xz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{yz} \end{bmatrix} \]  \hspace{1cm} (2.6)

\[ D = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 - \nu & \nu & 0 & 0 & 0 & 0 \\ \nu & 1 - \nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 - \nu & 0 & 0 & 0 \\ 0 & 0 & 0 & (1 - 2\nu)/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1 - 2\nu)/2 & 0 \end{bmatrix} \]  \hspace{1cm} (2.7)

where, \( E \) is Young's modulus and \( \nu \) is Poisson's ratio. \( \sigma \) is the stress tensor of an arbitrary point on the body. For instance, the first subscript of \( \sigma_{xx} \) indicates the normal vector of the plane where the stress component \( \sigma_{xx} \) acts and the second subscript shows the direction of the stress component in \( xyz \) coordinates. In the matrix format \( D \) is the constant constitutive \( 6 \times 6 \) matrix and \( \varepsilon \) is the strain \( 6 \times 1 \) vector at the same point of the body where the stress acts. If the body is isotropic, \( D \) is a symmetric matrix. The derivation is based on the assumption of solid isotropic linear elastic material and Hook's law [3]. In two dimensions and for planar stress \( \sigma \) and \( \varepsilon \) are \( 3 \times 1 \) vectors and \( D \) is a \( 3 \times 3 \) matrix given as in the following:

\[ \sigma^T = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \sigma_{xy} \end{bmatrix} \]  \hspace{1cm} (2.8)

\[ \varepsilon^T = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{xy} \end{bmatrix} \]  \hspace{1cm} (2.9)

\[ D = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1 - \nu \end{bmatrix} \]  \hspace{1cm} (2.10)

Equilibrium equation or balance principle of the body is given by

\[ \bar{\nabla}^T \sigma + b = 0 \]  \hspace{1cm} (2.11)

where, \( \bar{\nabla} \) is a differential operator which can be written in matrix form as
20

\[ \nabla^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & 0 & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \]  

(2.12)

in the planar stress case:

\[ \nabla^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} \end{bmatrix} \]  

(2.13)

\( b \) is the body force per unit volume.

\[ b^T = [b_x \ b_y \ b_z] \]  

(2.14)

Another important relation in linear elasticity is the relation between displacement \( u \) of an arbitrary point of the body with respect to the body reference coordinates and the strain at the same point. It is called kinematic relation and is given by

\[ \varepsilon = \nabla u \]  

(2.15)

where,

\[ u^T = [u_x \ u_y \ u_z] \]  

(2.16)

The set of equations (2.4), (2.11) and (2.15) constitute the strong form of the governing equilibrium differential equation. Equations (2.4), (2.11) and (2.15) are called field equations. Their derivation can be found in [3] or [10].

Field equations require boundary conditions. There are two types of boundary conditions: Neuman boundary condition where the traction force, \( t \), is given. The traction force is a force per unit area acting on a part of the external surface of the body, \( S_b \); Dirichlet boundary condition where displacement is prescribed on the part of surface of the body, \( S_y \). These two parts should not have intersections. The traction vector on an arbitrary surface is related to the stress tensor on that surface by the following relation.

\[ t = S_t n \]  

(2.17)

Where, \( n \) is the normal vector to the surface, \( S_t \) is the stress tensor given by

\[ S_t = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \]  

(2.18)

If the surface is the external surface of the body, (2.17) correspond to Neuman boundary condition.
2.2.2. **Transforming the strong form into the weak form**

The weak form is obtained multiplying (2.11) by a weight function $w$, integrating over the whole volume of the body, integrating by parts using Green-Gauss theorem [3] and finally using (2.17) as the boundary condition:

$$
\int_{V} (\vec{\nabla}w)^{T} \sigma dV = \int_{S} w^{T} t dS + \int_{V} w^{T} b dV
$$

(2.19)

where,

$$
w = \begin{bmatrix}
w_{x} \\
w_{y} \\
w_{z}
\end{bmatrix}
$$

In two dimensions there is no component $w_{z}$. In (2.19) $V$ is the volume and $S$ is the external surface of the body. The weak form (2.19) is also called principle of virtual work for static bodies that is general for any constitutive relation. $w$ in (2.19) is arbitrary. In two dimensions $V$ is replaced by the area, $A$, and $S$ is replaced by the outer line boundaries of the body, $l_{b}$.

2.2.3. **Meshing the solution domain, choosing element type and shape function**

The solution domain is the whole volume of the body. The solution domain is discretized or meshed into the finite elements. In three dimensions a finite element is a small volume of the body and in two dimensions it is a small area. Here Four-node rectangular element is employed, since all initial topologies of bodies are two dimension al boxes which are more straightforward to be meshed with Four-node rectangular elements. Fig. 2.1 shows the meshing of a rectangular two dimensional body.

![Fig. 2.1: Meshing a rectangular two dimensional body; there are three types of numbering: global nodal numbering, local nodal numbering and global element numbering. The local numbering of element number 5 is illustrated.](image)
Three types of numbering are required in meshing: global nodal numbering, local nodal numbering and global element numbering. The mesh should provide information about the relation between these numbers as well as the position of the nodes.

The shape functions of such an element are given by

\[
\begin{align*}
N_1 &= \frac{1}{4ab} (x - x_2)(y - y_4) \\
N_2 &= -\frac{1}{4ab} (x - x_1)(y - y_3) \\
N_3 &= \frac{1}{4ab} (x - x_4)(y - y_2) \\
N_4 &= -\frac{1}{4ab} (x - x_3)(y - y_1)
\end{align*}
\]  \hspace{1cm} (2.20)

where, \(a \) and \(b \) are half of the element sides shown in Fig. 2.2, \((x_i, y_i)\) is the coordinate of the node \(i \) of the element with respect to the body coordinate system and \((x, y)\) is the coordinate of an arbitrary point inside or on the boundaries of the element with respect to the body coordinate system. If the value of a property associated to the body behavior is given at the nodal points, its value at position \((x, y)\) can be interpolated using shape functions (2.20). For instance, displacement \(u\) at an arbitrary point with position \((x, y)\) inside or on the boundaries of the element can be found using displacement at the nodal points \((u_{x1}, u_{y1})\) as

\[
\begin{bmatrix}
u_x \\
u_y
\end{bmatrix} =
\begin{bmatrix}
N_1 & N_2 & N_3 & N_4 \\
0 & N_1 & N_2 & N_3 \\
0 & 0 & N_2 & N_3 \\
0 & 0 & 0 & N_3
\end{bmatrix}
\begin{bmatrix}
[\begin{array}{c} u_{x1} \\ u_{y1} \end{array} ] \\
[\begin{array}{c} u_{x2} \\ u_{y2} \end{array} ] \\
[\begin{array}{c} u_{x3} \\ u_{y3} \end{array} ] \\
[\begin{array}{c} u_{x4} \\ u_{y4} \end{array} ]
\end{bmatrix}
\]  \hspace{1cm} (2.21)

Combining (2.21) and (2.22) into one equation gives

\[
\begin{bmatrix}
u_x \\
u_y
\end{bmatrix} = N^e q^e_f 
\]  \hspace{1cm} (2.23)

where,

\[
\begin{align*}
\begin{bmatrix}
u_x \\
u_y
\end{bmatrix} &=
\begin{bmatrix}
u_{x1} \\
u_{x2} \\
u_{x3} \\
u_{x4} \\
u_{y1} \\
u_{y2} \\
u_{y3} \\
u_{y4}
\end{bmatrix} \\
N^e &=
\begin{bmatrix}
N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\
0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4
\end{bmatrix} \\
(q^e_f)^T &=
\begin{bmatrix}
u_{x1} & u_{y1} & u_{x2} & u_{y2} & u_{x3} & u_{y3} & u_{x4} & u_{y4}
\end{bmatrix}
\]  \hspace{1cm} (2.24, 2.25, 2.26)

\(q^e_f\) is the element nodal displacement vector. It can be concluded that the approximated continuous property can be calculated knowing the discretized one, using (2.23).
In two dimensions every node has two Degrees of Freedom (DoF), one in \( x \) and one in \( y \) direction; hence, a Four-node Rectangular Element has eight DoFs.

![Four-node Rectangular Element](image)

\textbf{Fig. 2.2}: Four-node Rectangular Element, reproduced from [3].

Shape functions should satisfy compatibility and completeness requirements [3]. Above shape functions are compatible if the sides of the Finite Element rectangles are parallel to the body coordinate axes. It is very difficult to mesh a complex geometry in such a way that the sides of all elements be parallel to the body coordinate axes. To overcome this problem four-node isoparametric elements are introduced where the four-node elements with arbitrary shape in the real domain is mapped to a parent domain. In the parent domain the element boundaries are parallel to the axes. The weak form also is mapped to the parent domain where usually the integration is done by Four-point Gauss quadrature integration method. Since, in this thesis work, all bodies are initially rectangular boxes, the sides of the elements are parallel to the body coordinate axes; hence, there is no need to use isoparametric elements or Gauss integration.

It should be noted that the element shape functions depend only on the geometry of the element. If the value of a property associated to the body behavior chosen to be, for instance, displacement at the nodal points, applying the differential operator (2.13) on (2.23) considering (2.15) gives:

\[
\dot{\mathbf{E}} = \mathbf{B} \mathbf{q} = \mathbf{N}^e \mathbf{q}^e = B \mathbf{q}^e_f
\]

where,

\[
B = \mathbf{N}^e
\]

For two dimensional four-node rectangular element \( B \) can be calculated as

\[
B = \frac{1}{4ab} \begin{bmatrix} y_4 - y_3 & 0 & y_3 - y & 0 & y_2 - y & 0 & y_2 - y & 0 \ x - x_2 & 0 & x_1 - x & 0 & x - x_3 & 0 & x_1 - x & 0 \ 0 & y - y_4 & y - y_3 & 0 & y - y_2 & 0 & y - y_2 & 0 \ x - x_2 & 0 & x_1 - x & 0 & x - x_3 & 0 & x_1 - x & 0 \ 0 & y - y_4 & y - y_3 & 0 & y - y_2 & 0 & y - y_2 & 0 \ 0 & x - x_2 & 0 & x_1 - x & 0 & x - x_3 & 0 & x_1 - x \ \end{bmatrix}
\]
2.2.4. Choosing weight functions and establishing the system of algebraic equations for an element

Weak form (2.19) is derived for the whole body but there is no restriction for limiting the integration domain to an element of the body. In two dimensions for planar stress the stresses, tractions and body forces do not depend on the z-coordinate hence \( dV_e = h_e dA_e \) and the surface integral in (2.19) becomes:

\[
\int_S w^T t \, dS = \int_{l_e} w^T h_e \, dl_e
\]

where, \( h_e \) is the thickness of the element, \( A_e \) and \( l_e \) are the area and the outer boundary line of the element respectively. The boundary traction vector \( t \) is unknown for an element. Hence, the weak form in two dimensions is

\[
\int_{A_e} (\nabla w)^T \sigma \, h_e \, dA_e = \int_{l_e} w^T t \, h_e \, dl_e + \int_{A_e} w^T b \, h_e \, dA_e
\]  

(2.30)

Substituting (2.4) and (2.15) in (2.30):

\[
\int_{A_e} (\nabla w)^T D \nabla u \, h_e \, dA_e = \int_{l_e} w^T t \, h_e \, dl_e + \int_{A_e} w^T b \, h_e \, dA_e
\]  

(2.31)

(2.31) is a continuous differential equation where the unknown variable is \( u \). Galerkin method suggests that the weight vector \( w \) is the same as the unknown sought variable \( u \). Hence,

\[
\int_{A_e} (\nabla u)^T D \nabla u \, h_e \, dA_e = \int_{l_e} u^T t \, h_e \, dl_e + \int_{A_e} u^T b \, h_e \, dA_e
\]  

(2.32)

The most important part in Finite Element formulation is to replace the sought continuous variable \( u \) by its approximated discretized unknown values at nodal points according to (2.23). Doing so and considering (2.27) the differential equation (2.32) is converted to an algebraic equation:

\[
\int_{A_e} (Bq_f^e)^T D B q_f^e \, h_e \, dA_e = \int_{l_e} (Nq_f^e)^T t \, h_e \, dl_e + \int_{A_e} (Nq_f^e)^T b \, h_e \, dA_e
\]  

(2.33)

Finally, noting that \((q_f^e)^T\) cancels out from the two sides of (2.33) one has

\[
\left( \int_{A_e} B^T D B \, h_e \, dA_e \right) q_f^e = \int_{l_e} N^T t \, h_e \, dl_e + \int_{A_e} N^T b \, h_e \, dA_e
\]  

(2.34)

where, \( q_f^e \) is the element unknown nodal displacement vector with respect to body coordinate system shown in (2.26). Equation (2.34) can be written as

\[
k_e q_f^e = f_e
\]  

(2.35)
where,

\[
k_e = \left( \int_{A_e} B^T D B h_e dA_e \right) \tag{2.36}
\]

\[
f_e = \int_{l_e} N^T t h_e dl_e + \int_{A_e} N^T b h_e dA_e \tag{2.37}
\]

\(k_e\) is called element stiffness matrix and \(f_e\) is called element load vector. Mostly these integrals are evaluated numerically. It is more straightforward to transform the line integral into the parent domain and then use Gauss integration. The complete explanation is provided in [3]. Doing the assembly the element load vectors for inner nodes cancel out; moreover, for many of the problems the force vector at outer boundaries is given; so, there is no need to evaluate the force integrals. \(k_e\) of a Four-node rectangular element does not have a very complicated structure; so, it is possible to calculated the integral analytically.

2.2.5. Assembly the element algebraic equations into the global system of algebraic equations

Equation (2.35) is an element system of algebraic equations. In order to find the global system of algebraic equations for the whole body, algebraic equations of all elements must be assembled into the one global equation. In the literature there are different methods for assembling the global stiffness and force matrices. Here, a method that beside simplicity is efficient for computer programs is described. The numbering shown in Fig. 2.1 is essential for such an assembling process.

Assume that the element stiffness matrix is a \(4 \times 4\) matrix (in (2.35) the element stiffness matrix is a \(8 \times 8\) matrix).

For instance for element number 5 shown in Fig. 2.1 the element stiffness matrix is:

\[
\begin{align*}
\text{Global numbering:} & \quad 6 \quad 7 \quad 11 \quad 10 \\
\text{Local numbering:} & \quad 1 \quad 2 \quad 3 \quad 4 \\
\begin{bmatrix}
6 & 1 \\
7 & 2 \\
11 & 3 \\
10 & 4 \\
\end{bmatrix} & \times \begin{bmatrix}
k_{5}^{11} \\
k_{5}^{21} \\
k_{5}^{31} \\
k_{5}^{41} \\
\end{bmatrix} & = \begin{bmatrix}
k_{5}^{12} \\
k_{5}^{22} \\
k_{5}^{32} \\
k_{5}^{42} \\
\end{bmatrix} & \times \begin{bmatrix}
k_{5}^{13} \\
k_{5}^{23} \\
k_{5}^{33} \\
k_{5}^{43} \\
\end{bmatrix} & \times \begin{bmatrix}
k_{5}^{14} \\
k_{5}^{24} \\
k_{5}^{34} \\
k_{5}^{44} \\
\end{bmatrix} & = k_{5}
\end{align*}
\]

where, for instance the subscript in \(k_{5}^{12}\) indicates the number of the element, the first digit in superscript indicates the row number and the second digit in superscript shows the column number in the element stiffness matrix. These row and column numbers are also associated with the local numbering of the element nodes shown in Fig. 2.1.

If local node numbers at superscript are replaced by the global node numbers the corresponding position in global stiffness matrix is obtained. This means that for example \(k_{5}^{12}\) in element stiffness matrix must be placed at \(K_{6,7}\) in
global stiffness matrix and be summed with the other values coming from neighboring elements. For instance, for element 2:

\[
\begin{pmatrix}
1 & 2 & 3 & 4 \\
2 & 1 & k_{21}^{12} & k_{21}^{13} & k_{21}^{14} \\
3 & 2 & k_{22}^{12} & k_{22}^{13} & k_{22}^{14} \\
7 & 3 & k_{23}^{12} & k_{23}^{13} & k_{23}^{14} \\
6 & 4 & k_{24}^{12} & k_{24}^{13} & k_{24}^{14}
\end{pmatrix} = k_2
\]

\(k_2^{43}\) of element number 2 also must be placed at \(K_{67}\) in global stiffness matrix and be summed with \(k_5^{12}\) of element number 5 and (if the mesh is three dimensional) it also must be summed with other values coming from neighboring elements. Hence,

\[K_{67} = k_5^{12} + k_2^{43}\]

The same assembly process is applicable for the global force vector. Finally,

\[Ku = F \quad (2.38)\]

where, \(K\) is a \(m \times m\) matrix which is called global stiffness matrix, where, \(m\) is the total number of Elastic Degrees of Freedom or DoFs. In two dimensions there are two Elastic DoFs for each node; so, \(m\) is twice the number of nodes. These elastic DoFs must be distinguished from the reference DoFs that are defined later in Flexible Multibody Dynamics; \(u\) is the global displacement vector relative to the body coordinate system and \(F\) is the global force vector. The size of both vectors is \(m \times 1\). Note that \(K\) is a symmetric, sparse and bounded matrix; in Multibody dynamics notation \(K_{ij}^{ij}\) is used for showing the global stiffness matrix of body \(i\) associated with elastic coordinates (DoFs).

### 2.2.6. Introducing boundary conditions

As it was mentioned earlier there are two types of boundary conditions: Neuman boundary condition or natural boundary condition where the traction force, \(\mathbf{t}\), is given on a part of the external surface of the body, \(S_h\); Dirichlet boundary condition where displacement is prescribed on the part of surface of the body, \(S_g\). These two parts should not have intersections.

Neuman boundary condition or natural boundary condition has already been applied when deriving the weak form (2.19). This is also necessary to apply Dirichlet boundary condition where displacement is prescribed on the part of surface of the body.

If \(u\) corresponds to a rigid body motion, where all components of \(u\) have the same value meaning that all nodes of the body or all nodes of an element have the same displacement, strain is zero according to (2.15), thus \(K\) is a singular matrix. Therefore, rigid body motion must be avoided by fixing at least three elastic DoFs in two dimensions and 6 elastic DoFs in a three dimensional problem. This means that the displacement at fixed DoFs are prescribed to be zero; thus, corresponding rows and columns in global stiffness matrix, displacement and force vectors must be eliminated. In Flexible Multibody Dynamics prescribing elastic DoFs is called defining Reference Condition (RC) [2].
2.2.7. Solving equations

After eliminating the fixed DoFs the inverse of stiffness matrix exists; hence, (2.38) has a unique solution:

\[ u = K^{-1} F \]  

(2.39)

It was already stated that in topology optimization displacement, stiffness and in some problems also the force are functions of the design variable \( X \). In two dimensional problems normalized thickness of the body can be chosen as the design variable. In this thesis work the normalized thickness is defined as

\[ X_e = \frac{h_e}{h_0}, \quad h_e = X_e h_0 \]  

(2.40)

Where, \( h_e \) is the thickness of the element number \( e \), \( h_0 \) is the initial thickness of element, also, \( h_0 \) is the thickness of the whole body before starting the optimization, \( X_e \) is the design variable or the normalized thickness of the element number \( e \); thus, (2.36) and (2.37) are modified to

\[ k_e = \left( \int_{A_e} B^T D B X_e h_0 dA_e \right) \]  

(2.41)

\[ k_e^0 = \frac{\partial k_e}{\partial X_e} = \left( \int_{A_e} B^T D B h_0 dA_e \right) \]  

(2.42)

\[ f_e = \int_{l_e} N^T t X_e h_0 dl_e + \int_{A_e} N^T b X_e h_0 dA_e \]  

(2.43)

where, \( k_e^0 \) is the element local specific stiffness matrix. With the above modification in accordance with the equilibrium constraint (2.2), (2.39) can be rewritten as

\[ u(X) = K^{-1}(X) F(X) \]  

(2.44)

where, \( X = [X_1, X_2, ... , X_n] \) and \( n \) is the total number of the design variables which equals to the number of elements.
2.3. Solving the Topology Optimization problem

For reasons that will be explained later Solid Isotropic Material with Penalization (SIMP) method is mostly used for Topology Optimization (TO) problems [4]. SIMP itself uses the Convex Linearization (CONLIN) or the Optimality Criteria (OC) [1 & 4] methods. The big advantage of CONLIN and OC methods is that they make an explicit and convex approximation of the objective and constraint functions. More importantly the result of the approximation is a separable function with respect to the design variable. These properties make it possible to find a local minimum in an efficient way when the number of design variables is huge.

The idea of CONLIN or OC is to change the objective and constraint functions by linearizing them at the intermediate variable $Y(X^k)$ and writing the two first terms of their Taylor expansion at $Y(X^k)$, where, $X^k$ is the design variable (normalized thickness for two dimensional problem) at iteration $k$ which is a constant vector; then, to solve the optimization sub-problem in the vicinity of $X^k$ with Lagrangian Duality method. The solution of the sub-problem, $X^{k+1}$, is then assigned to $X^k$ and the method is repeated until the convergence criterion is achieved. The convergence criterion can be the change of the value of the objective function from one iteration to the next or the average change in the design variable as well as the change in the norm of the design variable: $||X^{k+1} - X^k|| < \varepsilon$, where, $\varepsilon$ is a small value.

2.3.1. Approximating the objective function

CONLIN or OC approximation of the objective and constraint functions can be done as follows:

$$g_i(X) \approx g_i(X^k) + \sum_{\varepsilon=1}^{n} \frac{\partial g_i(X^k)}{\partial Y_e(X_e)} (Y_e(X_e) - Y_e(X_e^k)) \quad (2.45)$$

The choice of the intermediate variable $Y(X^k)$ depends on the function that is linearized, $g_i(X)$. A good choice of $Y(X^k)$ results in a fast convergence of the optimization algorithm. The OC method suggests the following intermediate variable for solving topology optimization problems:

$$Y_e(X_e) = \begin{cases} X_e & \text{if } \frac{\partial g_i(X)}{\partial X_e} > 0 \\ X_e - \alpha & \text{if } \frac{\partial g_i(X)}{\partial X_e} \leq 0 \end{cases} \quad (2.46)$$

For $\alpha = 3$ the algorithm shows fast convergence. The idea is that the objective function is a more smooth function of the intermediate variable than the design variable itself. (2.46) shows that the choice of the intermediate variable $Y$ also depends on the sign of the sensitivity of the objective or the constraint function. The sensitivity of the compliance can be calculated as follows:

$$\frac{\partial (F^T(X), u(X))}{\partial X_e} = \frac{\partial F^T(X)}{\partial X_e} u(X) + F^T(X) \frac{\partial u(X)}{\partial X_e} \quad (2.47)$$
\( \frac{\partial u(X)}{\partial X_e} \) can be found by differentiating the equilibrium constraint with respect to the design variable. For the static problems the equilibrium constraint is given by:

\[ K(X).u(X) = F(X) \]  \hspace{1cm} (2.48)

After differentiation:

\[ \frac{\partial K(X)}{\partial X_e} u(X) + K(X) \frac{\partial u(X)}{\partial X_e} = \frac{\partial F(X)}{\partial X_e} \]  \hspace{1cm} (2.49)

Solving (2.49) for \( \frac{\partial u(X)}{\partial X_e} \):

\[ \frac{\partial u(X)}{\partial X_e} = K^{-1}(X) \left( \frac{\partial F(X)}{\partial X_e} - \frac{\partial K(X)}{\partial X_e} u(X) \right) \]  \hspace{1cm} (2.50)

Substituting (2.50) into (2.47) gives an expression for the sensitivity of the objective function, compliance, with respect to the design variable.

\[ \frac{\partial (F^T(X).u(X))}{\partial X_e} = \frac{\partial F^T(X)}{\partial X_e} u(X) + F^T(X) K^{-1}(X) \left( \frac{\partial F(X)}{\partial X_e} - \frac{\partial K(X)}{\partial X_e} u(X) \right) \]  \hspace{1cm} (2.51)

In general, \( F \) is a nonlinear function of displacement and design variables, but here for simplicity it is assumed that \( F \) is a known constant vector; hence, \( \frac{\partial F(X)}{\partial X_e} = 0 \). According to (2.47):

\[ F^T(X) K^{-1}(X) = u^T(X) \]

where, symmetry property of the stiffness matrix is used; so, (2.51) is simplified to:

\[ \frac{\partial g_e(X)}{\partial X_e} = \frac{\partial (F^T(X).u(X))}{\partial X_e} = -u^T(X) \frac{\partial K(X)}{\partial X_e} u(X) \]  \hspace{1cm} (2.52)

The stiffness matrix \( K(X) \) is symmetric positive definite and has a linear dependence to the design variable \( X_e \), see (2.42), so the sensitivity of the objective function, compliance, is always negative. Hence, according to (2.46) the choice of the intermediate variable is only \( X_e^{-a} \).

In finite element formulation of (2.52), \( u(X) \) is the global displacement vector and \( K(X) \) is the global stiffness matrix, whereas, \( X_e \) is the element design variable which is a normalized optimization property that represents the existence of the element in topology optimization. It can be chosen to be the element normalized thickness in two dimensional or element normalized density in three dimensional problems. According to the definition of the stiffness matrix that is given in Section 2.2, \( K(X) \) is built by assembling the element stiffness matrices \( k_e \)'s and \( k_e \) linearly depends on \( X_e \) (see equation (3.38)). It can be shown that
\[
\frac{\partial k_e}{\partial X_i} = \begin{cases} 
  k^0_e & \text{if } i = e \\
  0 & \text{if } i \neq e 
\end{cases} \tag{2.53}
\]

where, \( k^0_e \) is the normalized or specific element stiffness matrix. Note that \( k^0_e \) is not a function of \( X \). Having this explanation in mind, (2.52) can be written in the form:

\[
\frac{\partial g_0(X)}{\partial X_e} = -u^T_e(X)k^0_e u_e(X) \tag{2.54}
\]

where, \( u_e(X) \) is the element displacement vector which is a column vector similar to (2.26).

Back to (2.45), the OC approximation of the objective function of the optimization problem defined in (2.3) at iteration \( k \) is:

\[
g_0(X) \approx g_0(X^k) + \sum_{e=1}^{n} \frac{\partial g_0(X^k)}{\partial X_e} \frac{\partial X_e^k}{\partial Y_e}(Y_e(X_e) - Y_e(X_e^k))
\]

After some work:

\[
g_0(X) \approx g_0(X^k) + \sum_{e=1}^{n} \frac{\partial g_0(X^k)}{\partial X_e} \left( -\frac{(X_e^k)^{1+\alpha}}{\alpha} \right)(X_e^{-\alpha} - (X_e^k)^{-\alpha}) \tag{2.55}
\]

Writing (2.54) for \( X_e^k \) and substituting it in above expression:

\[
g_0(X) \approx g_0(X^k) + \sum_{e=1}^{n} u^T_e(X^k)k^0_e u_e(X^k) \left( -\frac{(X_e^k)^{1+\alpha}}{\alpha} \right)(X_e^{-\alpha} - (X_e^k)^{-\alpha})
\]

or

\[
g_0(X) \approx g_0(X^k) + \sum_{e=1}^{n} u^T_e(X^k)k^0_e u_e(X^k) \frac{(X_e^k)^{1+\alpha}}{\alpha}X_e^{-\alpha} - \sum_{e=1}^{n} u^T_e(X^k)k^0_e u_e(X^k) \frac{(X_e^k)^{1+\alpha}}{\alpha}(X_e^k)^{-\alpha} \tag{2.55}
\]

Note that at iteration \( k \) the approximation is done around the point \( X^k \), so \( X^k \) is a constant vector, therefore the first and the third terms of (2.55) are constants. Since the goal is to find the optimum design, constant values make no contribution to the minimization problem. The exact value of the objective function can be found using (2.48) and (2.3) later; hence, the approximated objective function about \( X_e^k \) can be changed to:

\[
g_0^k(X) = g_0^{k,OC}(X) = \sum_{e=1}^{n} u^T_e(X^k)k^0_e u_e(X^k) \frac{(X_e^k)^{1+\alpha}}{\alpha}X_e^{-\alpha} = \sum_{e=1}^{n} b_e^k X_e^{-\alpha} \tag{2.56}
\]
where, superscript $k$ illustrates the iteration $k$, superscript $OC$ illustrates the Optimality Criteria approximation and $b^k_e$ is a constant at iteration $k$ given by

$$b^k_e = \frac{u^k_e(X^k)k^2u^k_e(X^k)(X^k)^{1+\alpha}}{\alpha}$$  \hspace{1cm} (2.57)

Since the constraint function in current optimization problem (2.3) is linear there is no need to calculate its approximation. The sub-problem approximation of the optimization problem (2.3) now can be written in the form:

$$\begin{align*}
\begin{cases}
\min_X g^{k,OC}_0(X) = \sum_{e=1}^{n} b^k_e X_e - \alpha \\
\text{s. t.} \quad g_1(X) = \sum_{e=1}^{n} a_e X_e - V_{max} \leq 0 \\
X \in \mathcal{X} = \{X \in \mathbb{R}^n; X^e_{min} \leq X_e \leq X^e_{max}, e = 1, ..., n\}
\end{cases}
\end{align*}$$  \hspace{1cm} (2.58)

### 2.3.2. Minimization of Lagrangian function

As a result of OC approximation the objective function is converted to the sum of separated convex functions of the design variable. Convexity and separability properties of the optimization sub-problem make Lagrangian Duality method very efficient for minimization. The reader is referred to references [1] or [4] for in depth understanding the Lagrangian Duality method. The idea of Lagrangian Duality is to solve a max-min problem that is equivalent to the original optimization problem. The Lagrangian function which is a function of the design variable and Lagrange multiplier $\lambda$ should be minimized with respect to the design variable. The design $X^*(\lambda)$ which minimizes the Lagrangian function is itself a function of $\lambda$. The result of such a minimization is also a function of $\lambda$ which is called Dual objective function. The number of Lagrange multipliers $\lambda$ is equal to the number of constraints. Since for current problem there is only one constraint there exist only one $\lambda$ here. Dual objective function must be maximized with respect to $\lambda$. For topology optimization problem Dual objective function is always concave [1]. Fixed point iteration or Newton method can be used for finding the stationary point. Fixed point iteration requires only the first derivative. $\lambda^*$ which maximizes the Dual function is put back in design $X^*(\lambda^*)$ to find an explicit value for $X^*$.

The Lagrangian function at iteration $k$ can be written as

$$L^k(X, \lambda) = g^{k,OC}_0(X) + \lambda g_1(X)$$  \hspace{1cm} (2.59)

Where, $g^{k,OC}_0(X)$ and $g_1(X)$ are given in (2.58). The Lagrangian function contains both objective and constraint functions of the optimization problem (2.3). Only box constraint is left that needs to be considered during minimization of Lagrangian function. The Dual objective function at iteration $k$ is

$$\varphi^k(\lambda) = \min_X L^k(X, \lambda) = \min_X \sum_{e=1}^{n} [b^k_e X_e - \alpha + \lambda a_e X_e] - \lambda V_{max}$$  \hspace{1cm} (2.60)
Above summation is a sum of separable functions, thus:

\[ \varphi^k(\lambda) = \min_{X^k} \sum_{e=1}^{n} \min_{X_e} \left[ b_e^k X_e^{-\alpha} + \lambda a_e X_e \right] - \lambda V_{\text{max}} = -\lambda V_{\text{max}} + \sum_{e=1}^{n} \min_{X_e} [L_e^k(X_e, \lambda)] \]

Hence, the minimization is done on \( n \) single variable convex functions instead of one function with \( n \) variables. It is very straightforward to perform minimization on single variable convex functions. The minimum occurs at stationary point, but it also must be in box \( X_e^{\text{min}} \leq X_e \leq X_e^{\text{max}} \). Stationary point can be found by

\[ \frac{\partial L_e^k(X_e, \lambda)}{\partial X_e} = -\alpha b_e^k X_e^{-\alpha - 1} + \lambda a_e = 0 \]

Hence,

\[ X_e^{\text{trial}}(\lambda) = \left( \frac{\alpha b_e^k}{\lambda a_e} \right)^{\frac{1}{1+\alpha}} \] (2.61)

where, \( X_e^{\text{trial}} \) is the design where \( L_e^k(X_e, \lambda) \) is stationary. Considering the box constraint the minimum \( X_e^*(\lambda) \) is given by

\[ X_e^*(\lambda) = \begin{cases} 
X_e^{\text{min}} & \text{if } X_e^{\text{trial}}(\lambda) < X_e^{\text{min}} \\
X_e^{\text{max}} & \text{if } X_e^{\text{trial}}(\lambda) > X_e^{\text{max}} \\
X_e^{\text{trial}}(\lambda) & \text{otherwise} 
\end{cases} \] (2.62)

\( \lambda \) must be determined by solving the dual problem that is

\[ (D)^k \begin{cases} 
\max_{\lambda} \quad \varphi^k(\lambda) \\
\text{s. t. } \lambda \geq 0 
\end{cases} \] (2.63)

Putting \( X_e^*(\lambda) \) which minimizes the dual function \( \varphi^k(\lambda) \) back in (2.60) gives an explicit expression for \( \varphi^k(\lambda) \):

\[ \varphi^k(\lambda) = \sum_{e=1}^{n} \left[ b_e^k (X_e^*(\lambda))^{-\alpha} + \lambda a_e X_e^*(\lambda) \right] - \lambda V_{\text{max}} \] (2.64)

As it was already mentioned \( \varphi^k(\lambda) \) is concave; hence, it has only one maximum that can be determined by finding the zero of its derivative using a numerical method like Newton’s method, Bi-section or any other. In the following it is shown how the first derivative of \( \varphi^k(\lambda) \) with respect to \( \lambda \) can be found. \( \varphi^k_e(\lambda) \) is defined as

\[ \varphi^k_e(\lambda) = b_e^k (X_e^*(\lambda))^{-\alpha} + \lambda a_e X_e^*(\lambda) \]

and obtains

\[ \frac{\partial \varphi^k_e(\lambda)}{\partial \lambda} = \frac{\partial \varphi^k_e(\lambda)}{\partial X_e^*} \frac{\partial X_e^*}{\partial \lambda} + \frac{\partial \varphi^k_e(\lambda)}{\partial \lambda} \]
The first term of the right hand side of above equality is always zero; as if the stationary point is in the box then \( \frac{\partial f^k(x^e)}{\partial x_e} = 0; \) hence, \( \frac{\partial f^k(x^e)}{\partial x_e} = 0. \) If the stationary point is out of the box then \( X^*_e = X^e_{min} \) or \( X^*_e = X^e_{max} \) that are constant values; thus, \( \frac{\partial f^k(x^e)}{\partial x_e} = 0. \) Hence,

\[
\frac{\partial f^k(\lambda)}{\partial \lambda} = \sum_{e=1}^{n} \left[ \frac{\partial f^k(x^e)}{\partial \lambda} \right] - V_{max} = \sum_{e=1}^{n} a_e X^*_e(\lambda) - V_{max} = 0 \tag{2.65}
\]

where, \( X^*_e(\lambda) \) is given by (2.62). Note that according to (2.65) \( \lambda \) must be found such that the volume constraint is satisfied. This is more straightforward to find the zero by Bi-section method.

After finding positive value of \( \lambda \) which maximizes the dual problem (2.63), its value is used in (2.62) to get \( X^*_e. \) The procedure must be done simultaneously for all \( e = 1, \ldots, n \) to get \( X^*. \) Note that \( X^{k+1} = X^*. \) If the convergence criterion is not satisfied \( X^{k+1} \) is treated as \( X^k \) and \( k \) is added by one. The whole process from (2.45) must be repeated until convergence criterion is satisfied.

### 2.3.3. SIMP method

When doing the topology optimization an existence variable (the same as the design variable) is assigned to each finite Element. For instance, in Abaqus this variable is called ‘Normalized Optimization Property’. In two dimensional problems it can be the normalized thickness and in three dimensions it can be the normalized density. Conventionally, when this variable equals to one it means that the element exists, on the other hand, if it is zero there should be a ‘hole’ at the place of the corresponding element. However, zero values introduce singular stiffness matrices which are difficult to handle numerically. Instead a small value close to zero can be used, for instance, \( 1e - 3. \) In topology optimization, the desired value of the design variable after convergence is either \( X^e_{max} = 1 \) or \( X^e_{min} = 1e - 3. \) The intermediate values between 1 and \( 1e - 3 \) need to be avoided. However, the intermediate values cannot be totally avoided in OC method even if the convergence criterion is satisfied, but they can be penalized to reach the box limits \( X^e_{max} \) and \( X^e_{min} \) with less number of iterations. Penalizing is done by introducing an effective Young’s modulus \( (X^e)^{q-1}E \) \([1 & 4]\). The reason why this kind of penalization works is explained in chapter ‘Topology Optimization of Multibody systems’ of this thesis work where there is an explanation about the effect of flying elements.

According to constitutive relation given in Section 2.2, Equations (2.10), (2.41) and (2.42), \( X^e \) in element stiffness matrix changes to \( (X^e)^q \); the only modification on OC is to change (2.57) to:

\[
b^k_e = u^e(X^k)q(X^e)^{q-1}k^0e(X^k) \left( \frac{X^e}{X^k} \right)^{1+\alpha} \tag{2.66}
\]

Mostly, \( q = 3 \), which causes less number of iterations to reach convergence than other values of \( q. \)

An important point that should be noted here is that SIMP modification makes the problem non-convex. This means that there can be several local minima and the problem with different initial states converges to different local optima; so, there is not a unique optimal design. Mesh dependency and appearance of the checkerboard pattern are other numerical problems of topology optimization (that is called archetype problems). If the optimal
design changes by using different or finer meshes, then the design is called mesh dependent. Using filters the problems of Mesh dependency and appearance of the checkerboard pattern can be fixed.

### 2.3.4. Filters

The mesh dependency of the design and the checkerboard pattern can be cured or alleviated using restrictions or filters. A mesh independent design can be achieved by filtering the sensitivity in OC approach. There exist several ways of filtering the sensitivity [5], for instance, Original mesh-independent sensitivity filter, Alternative sensitivity filter, Mean sensitivity filter and Bi-lateral sensitivity filter. Each of them has some advantages and disadvantages. The Mean sensitivity filter is implemented here, since it is easy to use and in most of cases it works very well. The Mean sensitivity filter is defined as

$$\frac{\partial g_0(X)}{\partial x_e} = \sum_{i \in N_e} \frac{\partial g_0(X)}{\partial x_i} \frac{1}{\sum_{i \in N_e} 1}$$

(2.67)

where, $g_0(X)$ is the objective function after applying the mean sensitivity filter, $N_e$ is a set containing the number of elements around the element $e$ within the radius $R$ (Neighbor elements).

$$N_e = \{ i \mid \| x_i - x_e \| \leq R \}$$

(2.68)

Thus, (2.66) changes to

$$b_e^k = \frac{\sum_{i \in N_e} \left[ q(X_i)^{q-1} u_i^T (X_e^k) k_i u_i (X_e^k) \right] (X_e^k)^{1+\alpha}}{\sum_{i \in N_e} 1} \frac{1}{\alpha}$$

(2.69)

An example of topology optimization of a plate subject to a vertical load applied on the lower right end of the plate is shown in Fig. 2.3. The optimization setup is the same as what has been described so far.

![Example of topology optimization](image)

**Fig. 2.3:** (left) A plate with a fixed left boundary where a static load is applied on the lower right end. The uniform initial thickness is $h_0 = 1$. The value of the force as well as all material properties is 1. (right) The topology optimization result, minimization of compliance. Number of design variables is 1500; the result is obtained after 85 iterations where the final volume is 0.4 of the initial one.
2.4. Flexible Multibody Dynamics

A Multibody system is a set of bodies with mechanical interactions. The interactions are through joints, contacts or force elements between bodies. If a solid body with a high stiffness within a Multibody system experiences a small acceleration while there are small forces acting on its connections, it can be assumed as a rigid body. The deformation of such a body, as a reaction to the inertia forces and forces at connections, is small and negligible. On the other hand, if the body is weak or there are high accelerations and forces in the system, body may bend and deform as a reaction to the forces, causing vibrations and waves propagation in the system, which are important for system dynamics; thus, the body must be assumed to be flexible in system simulation in order to be able to predict the true system behavior. In general, a Multibody system may be considered to consist of both rigid and flexible bodies. The examples of such systems are vehicle systems, high-speed robotic manipulators, airplanes and space structures.

2.4.1. Position vector

Two coordinate systems are used to describe the motion of a Multibody system. First one is the global coordinate system $xyz$ with zero or constant velocity which is an inertia system. This system can be attached to the ground with the origin at point $O$. The second one is the body coordinate system $x' y' z'$ attached to an arbitrary point $O'$ of the body. Body coordinate system translates and rotates with the body, see Fig. 2.4.

![Fig. 2.4: Global and body coordinates systems.](image)

The position of an arbitrary point $P$ on the body with respect to the global coordinates system is given by vector $R_p$, where,

$$ R_p = R_{o'} + r $$  \hspace{1cm} (2.70)

$R_{o'}$ and $r$ both are given with respect to the global system of coordinates. The position of point $P$ can also be described with respect to the body coordinate system by vector $u'_p$. Intuitively, it can be seen that $r$ and $u'_p$ are
the same vector in the space but described with respect to different coordinates systems. Vector $u'_p$ is related to vector $r$ through coordinate transformation:

$$r = Au'_p$$  \hspace{1cm} (2.71)

where, $A$ is a transformation or rotation matrix. Assume that the axes of two coordinates systems are initially parallel; if the body coordinate system rotates with the angle $\theta$ around a rotation axis with the unit vector $v = [v_1 \quad v_2 \quad v_3]$, it can be shown that the transformation matrix is described by

$$A = \begin{bmatrix}
2((\theta_0)^2 + (\theta_1)^2) - 1 & 2(\theta_1\theta_2 + \theta_0\theta_3) & 2(\theta_2\theta_3 + \theta_0\theta_1) \\
2(\theta_1\theta_2 + \theta_0\theta_3) & 2((\theta_0)^2 + (\theta_2)^2) - 1 & 2(\theta_1\theta_2 + \theta_0\theta_3) \\
2(\theta_1\theta_3 + \theta_0\theta_2) & 2(\theta_2\theta_3 + \theta_0\theta_1) & 2((\theta_0)^2 + (\theta_3)^2) - 1
\end{bmatrix}$$  \hspace{1cm} (2.72)

where, $\theta_0$, $\theta_1$, $\theta_2$ and $\theta_3$ are Euler parameters [2] which is called also rotational coordinates defined by

$$\theta^T = [\theta_0 \quad \theta_1 \quad \theta_2 \quad \theta_3]$$

$$\theta_0 = \cos \frac{\theta}{2} \quad \theta_1 = v_1 \sin \frac{\theta}{2}$$

$$\theta_2 = v_2 \sin \frac{\theta}{2} \quad \theta_3 = v_3 \sin \frac{\theta}{2}$$  \hspace{1cm} (2.73)

Hence, (2.70) can be written as

$$R_p = R'_o + Au'_p$$  \hspace{1cm} (2.74)

Euler parameters give the orientation of the body in the space. These 4 parameters are not independent. The advantage of using Euler parameters is avoiding the problem of singularities of the rotation matrix at a specific orientation.

2.4.2. Floating Frame of Reference

The expression given in (2.74) is general; there is no limitation for the body being rigid or flexible provided that the body coordinate system is rigidly attached to the body. However, for flexible bodies, since the body deforms, defining a rigid coordinate system for the body is not trivial. For this reason the concept of Floating Frame of Reference (FFR) is used in the literature. A Reference Condition (RC) is needed to be introduced in order to fix a FFR on the body. Reference Condition in a dynamical system can be seen to be the same as the Boundary Condition in a static structure, in the sense that both eliminate the rigid body motion of the coordinates which are described by the body coordinate system. Moreover, RC in multibody dynamics also means fixing the Floating Frame of reference to the body such that there is a unique solution for the body deformation. There will be more explanation about RC in this chapter (section 2.4.13.2). Large transitional and rotational motion of a flexible body can be described using FFR.
2.4.3. Position of an arbitrary point of a flexible body in the space

Consider a deformed body shown in Fig. 2.5. Global system of reference \(xyz\) together with Floating Frame of Reference \(x'y'z'\) is used for describing the position of an arbitrary point \(P\) in the body. Position of point \(P\) with respect to the FFR is given by \(u'_p\), thus

\[
u'_p = u'_0 + u'_f\] (2.75)

where, \(u'_0\) is the position of point \(P\) before deformation of the body and \(u'_f\) is the displacement of point \(P\) with respect to FFR. Substituting (2.75) into (2.74) gives the position of an arbitrary point in a flexible body with respect to the global system of coordinates:

\[
R_p = R'_o + A(u'_0 + u'_f)
\] (2.76)

![Fig. 2.5: global coordinate system and Floating Frame of Reference used to describe body deformation;](image)

There is infinite number of points in a body. It is not possible to calculate exact displacement of all these points. Instead, one finds the displacement of several points and then approximates the deformation of the other points using the found displacements. This can be done with the aid of Finite Element Method. If the body is discretized to finite elements, then the displacement of an arbitrary point inside or on the boundaries of element \(e\) can be approximated using the element shape function and the displacement at element nodal points. In the same manner as in (2.23):

\[
u'_f = N^e q^e_f \] (2.77)

where, \(N^e\) is the element local shape function, for instance, for a two dimensional body using Four-node Rectangular element shape function \(N^e\) is given by (2.25), and \(q^e_f\) is the displacement at element nodal points. In other form, considering displacement of all nodal points, \(q_f\), it is preferred to write:

\[
u'_f = S q_f \] (2.78)
where, $S$ is the global shape function, for instance, for a two dimensional body, using Four-node Rectangular element, shape function $S$ is a $2 \times m$ matrix with all entries to be zero except entries given by (2.25) but with global numbering of the elastic DoFs, where, $m$ is the total number of elastic DoFs. Note that $N^e$ and $S$ are functions of $u'_0$ and nodal positions (grid points) of the finite element mesh. In (2.78), $q_f$ is the displacement vector of all nodal points which is called vector of elastic coordinates defined relative to body system of reference. Substituting (2.78) into (2.76) gives:

$$ R_p = R'_o + A(u'_0 + S q_f) \tag{2.79} $$

(2.79) is the general formula for finding the global position of an arbitrary point of a flexible body in the space. It can be written in the form:

$$ R_p = R'_o + A u'_0 + A S q_f \tag{2.80} $$

The first two terms of right hand side in (2.80) give the rigid body motion of the body, where, $R'_o$ gives the rigid body translation which is called translation coordinates and $A u'_0$ gives the rigid body rotation of the body. The third term of right hand side in (2.80) presents the deformation of the body at point $P$ in the space. Presenting transitional, rotational and elastic coordinates of a body in one vector:

$$ q = [q_r, q_f] = \begin{bmatrix} R'_o \\ \theta \\ q_f \end{bmatrix} \tag{2.81} $$

where, $q$ is the vector of generalized coordinates and $q_r$ is called vector of reference coordinates of the body. Reference coordinates describe the rigid body position and orientation with respect to the global coordinate system, whereas, elastic coordinates explain the deformation of the body with respect to the body coordinate system.

### 2.4.4. Absolute velocity of an arbitrary point of a flexible body,

Differentiating (2.74) with respect to time yields

$$ \dot{R}_p = \dot{R}_o + \dot{A} u'_0 + \dot{A} u'_p \tag{2.82} $$

where, dot above the letters denotes differentiation with respect to time. Noting that $u'_p = u'_0 + S q_f$, where, $u'_0$ is constant and $S$ is only a function of the geometry, (2.82) can be written as

$$ \dot{R}_p = \dot{R}_o + \dot{A} u'_0 + A \dot{q}_f \tag{2.83} $$

The differentiation of the rotation matrix with respect to time is

$$ \dot{A} = \sum_{k=1}^{n_r} \frac{\partial A}{\partial \theta} \dot{\theta}_k = \sum_{k=1}^{n_r} A_{\theta} \dot{\theta}_k \tag{2.84} $$

where, $n_r$ is total number of the rotational coordinates, for instance, $n_r = 4$ for $\theta$ shown in (2.73), and
If $\mathbf{B}$ is defined as

$$\mathbf{B} = [A_{\theta_1}, \ldots, A_{\theta_n}] \mathbf{u}_p$$

then,

$$\dot{\mathbf{A}} \mathbf{u}_p = \mathbf{B} \dot{\theta}$$

and hence (2.83) can be written as

$$\dot{\mathbf{R}}_p = \dot{\mathbf{R}}_o + \mathbf{B} \dot{\theta} + \mathbf{A} \dot{\mathbf{q}}$$

Finally, (2.88) in matrix partitioned form is

$$\dot{\mathbf{R}}_p = \frac{\partial \mathbf{R}_p}{\partial \dot{\mathbf{q}}} \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{I} & \mathbf{B} & \mathbf{A} \mathbf{S} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{R}}_o \\ \dot{\theta} \\ \dot{\mathbf{q}}_f \end{bmatrix} = \mathbf{L} \dot{\mathbf{q}}$$

where, $\mathbf{I}$ is a $3 \times 3$ identity matrix, $\dot{\mathbf{q}}$ is the total vector of generalized velocities, and

$$\frac{\partial \mathbf{R}_p}{\partial \dot{\mathbf{q}}} = \mathbf{L} = \begin{bmatrix} \mathbf{I} & \mathbf{B} & \mathbf{A} \mathbf{S} \end{bmatrix}$$

2.4.5. Equation of Motion of bodies in a Multibody system

A Multibody System (MBS) is a set of bodies with mechanical interactions. The interactions are through joints, contacts or force elements between bodies. The system Equations of Motion (EoM) can be derived based on the energy conservation or dynamic equilibrium and principle of virtual work using Lagrange’s equation developed in [2]. According to [2] EoM of body $i$ of an MBS can be written in the form

$$\begin{cases} M_i \ddot{q}^i + K_i q^i + C^T_q \lambda = Q^i_e + Q^i_v \\ C(q, t) = 0 \end{cases} \quad i = 1, 2, \ldots, n_b$$

where, $\mathbf{C}(\mathbf{q}, t)$ is the vector of kinematic algebraic constraint equations describing the MBS joints and prescribed trajectories, $n_b$ is the total number of bodies in MBS, $M_i$ is the symmetric mass matrix of body $i$, $\dot{q}^i$ is the total vector of generalized accelerations, $K_i$ is the body stiffness matrix associated with generalized coordinates $q^i$, $C^T_q$ is the transpose of constraints Jacobian matrix where the differentiation of constraints equation is done with respect to $q^i$, $\lambda$ is the vector of Lagrange multipliers, $Q^i_e$ is the vector of generalized forces associated with generalized coordinates of body $i$ and finally $Q^i_v$ is a quadratic velocity vector. The derivation of these terms is given in the following sections. Note that in (2.91) one has $C^T_q$ not $C^{RT}_q$. In the latter case, $C^{RT}_q \lambda^i$ must be used;
where $\lambda_i$ is a vector containing only those Lagrange multipliers which are associated with body $i$. $\lambda$ gives the value of the reaction forces at joints.

Equation (2.91) is a nonlinear second order Differential Algebraic Equation (DAE). Nonlinear differential equation and algebraic equations must be solved simultaneously. The unknowns are generalized coordinates, velocities, accelerations as well as Lagrange multipliers. Note that, in EoM (2.91) dependent and independent coordinates are not separated, so EoM includes also dependent generalized coordinates. The vector of Lagrange multipliers is introduced to take care of dependent coordinates. Also note that the number of dependent coordinates equals to the length of $\lambda$ vector which equals to the total number of constraints. For an MBS consisting of rigid and flexible bodies number of equations may raise to hundreds of thousands, hence, the only way for solving such a system is to use numerical methods (See section 5.11.3).

### 2.4.6. Mass matrix of a flexible body

Kinetic energy of the bodies of a multibody system is needed for deriving Equation of Motion (EoM). The mass matrix can be defined using the kinetic energy. Having the absolute velocity of an arbitrary point of the body in (2.89) the kinetic energy of body $i$ can be defined as

$$
T^i = \frac{1}{2} \int \rho^i (\dot{R}_p^i)^T \dot{R}_p^i dV^i
$$

(2.92)

Where, $T^i$ is the kinetic energy of body $i$, $V^i$ is the volume of the body and $\rho^i$ is the body density. Superscript $i$ shows that the parameter is associated to body $i$. Writing (2.89) for body $i$ and inserting it in (2.92)

$$
\dot{T}^i = \frac{1}{2} \dot{(\dot{q})}^T \int \rho^i (L^i)^T L^i dV^i \dot{q}^i = \frac{1}{2} \dot{(\dot{q})}^T M^i \dot{q}^i
$$

(2.93)

where,

$$
M^i = \int \rho^i (L^i)^T L^i dV^i = \int \rho^i \left[ I \begin{bmatrix} B^i \end{bmatrix}^T A^i S^i \right] \begin{bmatrix} I & B^i & A^i S^i \end{bmatrix} dV^i
$$

(2.94)

Multiplying the matrices and integrating each term gives

$$
M^i = \begin{bmatrix}
m_{RR}^i & m_{R\theta}^i & m_{RF}^i \\
m_{R\theta}^i & m_{\theta \theta}^i & m_{\theta F}^i \\
0 & 0 & m_{FF}^i
\end{bmatrix}
$$

(2.95)

where,
\[
\mathbf{m}^i_{rr} = \int_{\mathbf{v}_i} \rho^i I^i dV^i, \quad \mathbf{m}^i_{r\theta} = \int_{\mathbf{v}_i} \rho^i \mathbf{B}^i dV^i \\
\mathbf{m}^i_{rf} = \mathbf{A}^i \int_{\mathbf{v}_i} \rho^i \mathbf{S}^i dV^i, \quad \mathbf{m}^i_{r\theta} = \int_{\mathbf{v}_i} \rho^i (\mathbf{B}^i)^T \mathbf{B}^i dV^i \\
\mathbf{m}^i_{rf} = \int_{\mathbf{v}_i} \rho^i (\mathbf{B}^i)^T \mathbf{A}^i \mathbf{S}^i dV^i, \quad \mathbf{m}^i_{r\theta} = \int_{\mathbf{v}_i} \rho^i (\mathbf{S}^i)^T \mathbf{S}^i dV^i
\]

(2.96)

If the body is rigid, there is no elastic coordinates; hence, \( \mathbf{M}^i \) turns to

\[
\mathbf{M}^i = \begin{bmatrix} \mathbf{m}^i_{rr} & \mathbf{m}^i_{r\theta} \\ \mathbf{m}^i_{rf} & \mathbf{m}^i_{r\theta} \end{bmatrix}
\]

(2.97)

Calculation of mass matrix integrals for three dimensional body is given in [2].

2.4.6.1. Calculating mass matrix Integrals for planar motion

If it is assumed that the body moves only in \( xy \) plane, (2.72) for body \( i \) changes to

\[
\mathbf{A}^i = \begin{bmatrix} \cos \theta^i & -\sin \theta^i \\ \sin \theta^i & \cos \theta^i \end{bmatrix}
\]

(2.98)

Where, \( \theta^i \) is the angle of rotation of body \( i \) around \( z \) axis. Looking at the rotation matrix (2.98) it is more straightforward to choose \( \theta^i \) as the rotational coordinate of the body rather than the Euler parameters given in (2.73). So, in planar motion, reference coordinates are given by

\[
\mathbf{q}^i_r = [R^i_1 \hspace{1cm} R^i_2 \hspace{1cm} \theta^i]^T
\]

(2.99)

Where, \([R^i_1 \hspace{1cm} R^i_2] \) is a vector which defines the position of the origin of the body coordinate system. The partial derivative of \( \mathbf{A}^i \) with respect to the rotational coordinate \( \theta^i \) is

\[
\mathbf{A}^i_{\theta} = \begin{bmatrix} -\sin \theta^i & -\cos \theta^i \\ \cos \theta^i & -\sin \theta^i \end{bmatrix}
\]

(2.100)

Hence, (2.86) for body \( i \) becomes

\[
\mathbf{B}^i = \mathbf{A}^i_{\theta} \mathbf{u}^i_0 = \mathbf{A}^i_{\theta} (\mathbf{u}^i_0 + \mathbf{u}^i_f)
\]

(2.101)

where, \( \mathbf{u}^i_0 = [x^i_1 \hspace{1cm} x^i_2] \) is a vector describing the undeformed position of a point in body \( i \) with respect to the body system of coordinates and \( \mathbf{u}^i_f \) is the displacement of that point after deformation specified with respect to the body system of coordinates.

A set of inertia shape integrals is required to describe the inertia properties of the flexible body. These integrals are given as
\[ I_i^l = \int_{V^l} \rho^l [x_i^l, x_2^l]^T dV^l \]
\[ I_{kl}^l = \int_{V^l} \rho^l x_k^l x_i^l dV^l \]
\[ I_{kl}^i = \int_{V^l} \rho^l x_k^l S_1^l dV^l \]  \hspace{1cm} (2.102)
\[ S_{kl}^i = \int_{V^l} \rho^l (S_k^l)^T S_1^l dV^l \]
\[ \bar{S}^i = \int_{V^l} \rho^i S^i dV^l \]

where, \( k, l = 1, 2 \) for a planar motion and \( k, l = 1, 2, 3 \) for spatial motion; \( S_k^l \) denotes the \( k^{th} \) row of \( S^l \). These integrals are not functions of time or coordinates, they involve only physical and geometrical properties of the undeformed body; hence, they need to be calculated only once before solving EoM (2.91).

Using (2.97)-(2.102) after some mathematical work mass matrix integrals for a two dimensional body are calculated as follows.

\[ m_{i\theta \theta}^l = \begin{bmatrix} m_i & 0 \\ 0 & m_i \end{bmatrix} \]  \hspace{1cm} (2.103)

where, for planar motion \( I \) in (2.96) is a \( 2 \times 2 \) identity matrix. \( m_i \) is the total mass of the body.

\[ m_{i\varphi \varphi}^l = A_0^l [I_1^l + S^l q_f^l] \]  \hspace{1cm} (2.104)
\[ m_{i\varphi \theta}^l = A^l S^l \]  \hspace{1cm} (2.105)
\[ m_{i\theta \varphi}^l = S_{11}^l + S_{22}^l \]  \hspace{1cm} (2.106)
\[ m_{i\theta \theta}^l = (m_{i\varphi \varphi})_{rr} + (m_{i\varphi \theta})_{rf} + (m_{i\theta \varphi})_{ff} \]  \hspace{1cm} (2.107)

where,

\[ (m_{i\varphi \varphi})_{rr} = l_{11}^i + l_{22}^i \]
\[ (m_{i\varphi \varphi})_{rr} = 2(\bar{T}_{11} + \bar{T}_{22})q_f^l \]
\[ (m_{i\varphi \varphi})_{rr} = (q_f^l)^T m_{i\varphi \varphi}^l q_f^l \]
\[ m_{i\varphi \varphi}^l = l_{12}^i + l_{21}^i + (q_f^l)^T (S_{12}^l - S_{21}^l) \]  \hspace{1cm} (2.108)
It should be noted that in (2.104) \( I_1 \) is the moment of mass of the undeformed body around axes of body coordinate system; if the origin of the body coordinate system is chosen to be at the center of mass, this term vanishes. \( m_{ij}^T \) in (2.106) is a constant matrix (diagonal in most of cases).

### 2.4.6.2. Calculating Inertia shape Integrals

The inertia shape integrals given in (2.102) can be calculated numerically using the lumped mass technique. The idea is to assume that the body consists of lumped masses distributed at grid points of the body, thus summation can be used instead of integration. Denoting \( S_i^j \) the shape function of body \( i \) corresponding to the grid point \( j \) the shape integrals become

\[
I_1 = \sum_{j=1}^{n_j} m_{ij} \mathbf{u}_0^i j \\
I_{kl} = \sum_{j=1}^{n_j} m_{ij} \mathbf{x}_k^i j \mathbf{x}_l^i j \\
\mathbf{I}_{kl} = \sum_{j=1}^{n_j} m_{ij} \mathbf{x}_k^i j \mathbf{S}_l^i j \\
\mathbf{S}^i = \sum_{j=1}^{n_j} m_{ij} \mathbf{S}^i j \\
\mathbf{S}^i = \sum_{j=1}^{n_j} m_{ij} \mathbf{S}^i j
\]

(2.109)

where, \( k, l = 1, 2 \) for a plane motion and \( k, l = 1, 2, 3 \) for spatial motion, \( S_{kl}^i j \) denotes the \( k^{th} \) row of \( S_i^j \), \( m_{ij} \) is the mass at grid point \( j \) of body \( i \), \( n_j \) is the total number of grid points and \( \mathbf{u}_0^i j = [x_1^i j \quad x_2^i j] \) is a vector describing the undeformed position of grid point \( j \) in body \( i \) with respect to the body system of coordinates.

It is very important that the total number of the grid points where the lumped masses are placed is equal or bigger than the total number of nodal points of Finite Element mesh. For instance, in this thesis work, when the lumped masses are placed to the center of Finite Element the result mass matrix was singular. The mass matrix should have a full rank after eliminating the fixed DoFs [2]; but, the rank was equal to twice the number of elements that is twice the number of lumped masses. Mass matrix turned to full rank when the lumped masses were placed at the position of the nodal points. One advantage of placing the lumped masses at nodal points is that the \( k^{th} \) row of the shape function, \( S_{k}^i j \), becomes zero everywhere except the value corresponding to \( j^{th} \) nodal point and \( k^{th} \) DoF which equals to 1. In other words, there is no need to approximate a parameter at nodal points since the value of that parameter is already known at nodal points. This can be verified by solving (2.25) for coordinates of a nodal point. Using such a shape function makes the computations cheaper.

It should also be noted that the shape function \( S_i^j \) in (2.109) is a global shape function corresponding to nodal point \( j \); one can either use the global shape function and then sum them according to (2.109) or instead use the local shape function given in (2.25) and do summation for element nodal values, then assemble the matrices according to assembly process explained in Section 2.2 to obtain \( \mathbf{S}^i, \mathbf{S}_{kl}^i \) and \( \mathbf{I}_{kl}^i \). However, when the lumped
masses are placed at the nodal points, using (2.109) is cheaper. Lumped masses at nodal points can be found by equally distributing an element mass to its nodal points and then for every node sum the masses of the node coming from elements it belongs to.

### 2.4.7. Body stiffness matrix

In Multibody dynamics notation $K^i_{ff}$ is used for showing the global stiffness matrix of body $i$ associated with the elastic coordinates (DoFs) derived in (2.38). The same as before, the body is assumed to be linear isotropic. The body stiffness matrix used in EoM (2.91) must include reference coordinates as well. Since reference coordinates, $q^i_r$, only describe rigid body motion there is no stress or strain caused by them, so there should be no stiffness. Body stiffness matrix in an MBS is defined by

$$K^i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & K^i_{ff} \end{bmatrix}$$ (2.110)

Note that $K^i$ does not depend on the generalized coordinates $q^i$. $K^i$ is constant and needs to be calculated only once before solving EoM. However, in topology optimization of MBS, design variable changes in every iteration and since $K^i$ is a function of the design variable, it needs to be updated accordingly (See Chapter 4).

### 2.4.8. Generalized forces

There are two types of the generalized forces associated to a flexible body. The first one is generalized elastic forces. The contribution of elastic generalized forces to EoM is through the body stiffness matrix shown in (2.110). The second one is the generalized external forces. Generalized forces are associated to the generalized coordinates $q^i$; this means that the virtual work caused by the generalized external forces can be calculated multiplying a virtual change in generalized coordinates by generalized external forces:

$$\delta W^i_e = (Q^i_e)^T \delta q^i$$ (2.111)

where, $\delta W^i_e$ is the virtual change in all external forces acting on body $i$, $\delta q^i$ is the virtual change in generalized coordinates and $Q^i_e$ is the generalized external forces.

In the following two kinds of the external forces are considered: externally applied forces and Spring-Damper-Actuator element.

#### 2.4.8.1. Externally applied forces

Assume that an external force $F^i(q^i, t)$ is applied at point $P$ of the body $i$. The virtual work is defined by

$$\delta W^i_e = (F^i)^T \delta R^i_P$$ (2.112)
where, $R^b_p$ is the global position vector shown in Fig. 2.5 (but for body $i$) and is defined in (2.80); $\delta R^b_p$ is the virtual
global displacement. In the same manner shown in Section 2.4.2 for finding absolute velocity of an arbitrary point
of a flexible body, virtual global displacement can be defined as

$$
\delta R^b_p = \begin{bmatrix} I & \overline{B}^i & A^i \end{bmatrix} \begin{bmatrix} R^i_{\theta^i} \\ \theta^i \\ q^i_f \end{bmatrix} = L^i \delta q^i
$$

(2.113)

Substituting (2.113) in (2.112) and comparing to (2.111), gives

$$(Q^i_{L})^T = (F^i)^T L^i
$$

(2.114)

where,

$$L^i = \begin{bmatrix} I & \overline{B}^i & A^i \end{bmatrix}
$$

(2.115)

Expression (2.114) gives the generalized form of externally applied force $F^i(q^i, t)$.

### 2.4.8.2. Spring-Damper-Actuator element

![Diagram](image)

Fig. 2.6: AN MBS with two bodies and two Spring-Damper-Actuator (SDA) elements; SDA-2 is attached to the ground at point G;
externally applied force $F$ acts on point $A$ of body $j$. Note that at connection points there is no need to add any kind of
constraints or joints.

An MBS with two bodies and two Spring-Damper-Actuator (SDA) elements is shown in Fig. 2.6. The spring constant
is assumed to be $K$, the damping coefficient is $c$, the actuator force is $F_a$ and the undeformed length of spring is $l_o$.
First consider SDA-1. The magnitude of the force along the SDA is defined by

$$F_{SDA} = K(l - l_o) + cl + F_a
$$

(2.116)

And the SDA force vector, $F_{SDA}$, is defined by
\[ F_{SDA} = F_{SDA} \dot{l} \]  

(2.117)

where, \( l \) is the current length of the spring, \( \dot{l} \) is the rate of change of \( l \) and \( \dot{\bar{l}} \) is the unit of the vector \( \bar{l} \). \( \bar{l} \) connects point \( P^i \) to point \( P^j \) pointing towards \( P^j \), which is the same as the relative position of point \( P^i \) with respect to point \( P^j \) in global system of coordinates:

\[ l = R_{p^i} - R_{p^j} \]  

(2.118)

Hence, the current length of the spring, \( l \), and its rate of change, \( \dot{l} \), can be expressed as

\[ l = \sqrt{\bar{l}^T \bar{l}} \]  

(2.119)

\[ \dot{l} = \bar{l}^T \dot{\bar{l}} \]  

(2.120)

where, \( \dot{\bar{l}} \) is the time derivative of \( \bar{l} \). Using (2.118) and (2.89), \( \dot{l} \) can be calculated as

\[ \dot{l} = R_{p^i} - R_{p^j} = L^i \dot{q}^i - L^j \dot{q}^j \]  

(2.121)

In the same way the virtual change in \( l \) can be expressed as

\[ \delta l = \delta R_{p^i} - \delta R_{p^j} = L^i \delta q^i - L^j \delta q^j \]  

(2.122)

The virtual work on MBS due to SDA is

\[ \delta W_e^i = -(F_{SDA} \dot{l})^T \delta l \]  

(2.123)

In (2.123) the minus sign is due to the work of the SDA on MBS which opposes the work of MBS on SDA.

Substituting (2.122) in (2.123) gives

\[ \delta W_e^i = -(F_{SDA} \dot{l})^T [L^i \delta q^i - L^j \delta q^j] = (Q^i)^T \delta q^i + (Q^j)^T \delta q^j \]  

(2.124)

where,

\[ (Q^i)^T = -F_{SDA} \bar{l}^T L_i \]  

(2.125)

\[ (Q^j)^T = F_{SDA} \bar{l}^T L_j \]  

Expression (2.125) gives the generalized forces of bodies \( i \) and \( j \) due to a common SDA element. For SDA-2 shown in the Fig. 2.6 where one end is fixed to the ground, if the unit vector \( \bar{l}_{H/L}^j \) is the unit of the vector pointing from \( H^j \) to \( G \), then the generalized force is given by \( F_{SDA,2} \bar{l}_{H/L}^j L_j^i \). Thus, assuming that an externally applied force acts on body \( j \) at point \( A \), the total generalized force acting on body \( j \) due to SDA elements and externally applied forces is

\[ (Q^j)^T = F_{SDA,1} \bar{l}_{p^i/p^j}^T L_j^i + F_{SDA,2} \bar{l}_{H/L}^j L_j^i + (F^i)^T L_A^i \]  

(2.126)
Where, $\vec{p}_{ip,i}$ is the unit of the vector pointing from $P^i$ to $P^j$, $L^i_{ij}$ is the expression given in (2.90) but for body $j$ and point $P^j$, $\vec{r}_{ij}$ is the unit of the vector pointing from $H^j$ to $G$, $L^j_{ij}$ is the expression given in (2.90) but for body $j$ and point $H^j$ and $L^i_j$ is the expression given in (2.90) but for body $j$ and point $A$. 

2.4.9. Kinematical constraints

If, in an MBS, some points of a body are restricted to move on a certain trajectory or if the motion of a body is constrained to the motion of another body through joints, the body is said to have kinematical constraint. Mathematically, a constraint can be described by a nonlinear algebraic function, $C_j(q,t)$, that depends on time and the MBS generalized coordinates $q = [(q^1)^T, (q^2)^T, ..., (q^{n_b})]^T$, where, $j$ is the constraint number, $t$ is the time, $n_b$ is the total number of bodies of MBS and $q^i$ is the generalized coordinates of body $i$. A constraint equation is defined as

$$C_j(q,t) = 0 \tag{2.127}$$

Assuming that there are several linearly independent constraints in the system, a set of the constraint functions can be expressed as

$$C(q,t) = 0 \tag{2.128}$$

where, $C(q,t) = [C_1(q,t), C_2(q,t), ..., C_{n_c}(q,t)]^T$ and $n_c$ is the total number of constraints. The partial derivative of constraint functions with respect to generalized coordinates $q$ is called constraint Jacobian matrix which is a $n_c \times n$ matrix and can be expressed as

$$C_q = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n_c1} & C_{n_c2} & \cdots & C_{n_cn} \end{bmatrix} \tag{2.129}$$

where, $C_{jk} = \partial C_j / \partial q_k$ and $n$ is the total number of generalized coordinates. In the following the derivation of constraint functions and their Jacobian matrix for the revolute joints and trajectory constraints is described.

2.4.9.1. Revolute joint

A revolute joint allows only the rotation between two bodies and constrains the position of the points of the bodies where it is attached. This can be mathematically described by demanding the relative position of two points of the bodies to be zero. Assume that the revolute joint is attached to body $i$ at point $P^i$ and to body $j$ at point $P^j$, hence,

$$C = R_{pi} - R_{pj} = 0 \tag{2.130}$$

where $R_{pi}$ is the global position of point $P^i$.

Using (2.130) in the same way as in (2.89), virtual change in $C$ can be expressed as
where, \( \delta q = \begin{bmatrix} \delta q^i \\ \delta q^j \end{bmatrix} \) and \( L \) is defined in (2.90) for a general body at an arbitrary point \( P \), hence, the constraint Jacobian matrix of a revolute joint is given by

\[
C_q = \begin{bmatrix} L^i \\ -L^j \end{bmatrix}
\] (2.132)

Substituting (2.80) but for body \( i \) at point \( P^i \) and body \( j \) at point \( P^j \) in (2.130), constraint equation in explicit form is obtained as:

\[
R_{o'}^i + A^i u_{o}^i + A^i S^i q_{o}^i - R_{o'}^j + A^j u_{o}^j + A^j S^j q_{o}^j = 0
\] (2.133)

Since, in three dimensions, each point has three transitional DoFs, (2.133) gives three constraint equations; in two dimensions (2.133) gives two constraint equations.

### 2.4.9.2. Trajectory constraint

A prescribed global trajectory specified in the global system of coordinates is defined by a function in space which relates global \( x, y \) and \( z \) coordinates. Thus, the constraint equation (2.128) can be written in the form

\[
C(x, y, z) = 0
\] (2.134)

It should be noted that \( C \) gives only one constraint equation. The global position of point \( P^i \), which is constrained to move on the trajectory, is expressed as

\[
R_{p^i} = \begin{bmatrix} x_{p^i}(q^i) \\ y_{p^i}(q^i) \\ z_{p^i}(q^i) \end{bmatrix}
\] (2.135)

Inserting coordinates of \( R_{p^i} \) in (2.134) yields

\[
C(R_{p^i}) = C(x_{p^i}(q^i), y_{p^i}(q^i), z_{p^i}(q^i)) = 0
\] (2.136)

Coordinates of \( R_{p^i} \) shown in (2.135) can be found using (2.80); thus, (2.136) gives the trajectory constraint equation of point \( P^i \). To find the constraint Jacobian matrix, one may either calculate the virtual change or differentiate the constraint function. The virtual change in the constraint function yields

\[
\delta C(R_{p^i}) = C_q \delta q = \frac{\partial C}{\partial R_{p^i}} \frac{\partial R_{p^i}}{\partial q} \delta q = 0
\] (2.137)

Finally, according to (2.90), \( \frac{\partial R_{p^i}}{\partial q} = L^i \), hence,

\[
C_q = \frac{\partial C}{\partial R_{p^i}} L^i = \begin{bmatrix} \frac{\partial C}{\partial x_{p^i}} \\ \frac{\partial C}{\partial y_{p^i}} \\ \frac{\partial C}{\partial z_{p^i}} \end{bmatrix} L^i
\] (2.138)
2.4.10. Quadratic velocity

The term quadratic velocity in (2.91) can be partitioned to vectors which are associated with transitional, rotational and elastic coordinates. Thus,

\[
Q_v = [(Q_v)_{T}^T (Q_v)_\theta^T (Q_v)_f^T]^T
\]  
(2.139)

According to [2] the terms of quadratic velocity for planar motion can be calculated as

\[
\begin{align*}
(Q_v)_{T} &= (\dot{\theta}^i)^2 A^i (S^i q_f^i + I_1^i) - 2\dot{\theta}^i A_b S^i \dot{q}_f^i \\
(Q_v)_\theta &= -2\dot{\theta}^i (q_f^i)^T (m_{ff}^i q_f^i + (T_{11}^i + T_{22}^i))^T \\
(Q_v)_f &= (\dot{\theta}^i)^2 (m_{ff}^i q_f^i + (T_{11}^i + T_{22}^i))^T + 2\dot{\theta}^i (S_{12}^i - S_{21}^i) q_f^i
\end{align*}
\]  
(2.140)

where, \(S^i\), \(I_1^i, I_{11}^i, I_{22}^i, S_{12}^i, S_{21}^i\) are inertia integrals given in (2.109), \(\dot{\theta}^i\) is the angular velocity of body \(i\) which is unknown, \(A^i\) is the rotation matrix given in (2.98), \(A_b\) is defined in (2.100), \(q_f^i\) is the elastic coordinates of body \(i\) which is an unknown vector and \(m_{ff}^i\) is the mass matrix associated with the elastic coordinates which is given in (2.106).

2.4.11. Solving the Equation of Motion

2.4.11.1. Choosing a proper element type

As it was already mentioned, the Finite Element formulation is used to discretize a body to finite areas or volumes. The choice of element type is essential. In [2], beam or plate elements are used for finite element formulation. These types of elements require slopes at nodal points as an elastic coordinate (DoF). Then introducing intermediate element coordinate system would be necessary for modeling the rigid body kinematics. Moreover, using these elements, problems with large rotation cannot be solved. However, if an element type is employed which does not have slopes or rotation as element DoFs, introducing intermediate element coordinate system would be unnecessary and also there would be no problem with large rotations. Moreover, the body coordinate system (Floating Frame of Reference) is needed to be used when meshing the body in finite elements and specify undeformed nodal positions with respect to that coordinate system. This is what has been exactly done in Section 2.2.3 when the body was meshed by rectangular elements. As it was discussed in Section 2.2.3, there is no need to transform the elastic coordinate to another domain either, if the boundary lines of elements are parallel to the axes of the body coordinate system.

2.4.11.2. Reference Condition

In order to be able to describe the deformation of a body uniquely the body Floating Frame of Reference (FFR) must be rigidly attached to the body. For flexible bodies defining a rigid coordinate system for the body is not trivial. A Reference Condition (RC) is needed to be imposed in order to fix a FFR on the body. Reference Condition in a dynamical system can be seen to be the same as the Boundary Condition in a static structure problem, in the sense that both eliminate the rigid body motion of the coordinates which are described by the body coordinate
Moreover, RC in multibody dynamics also means fixing the Floating Frame of Reference to the body such that there is a unique solution for the body deformation with respect to the FFR. If RC is not imposed, the mass matrix as well as the stiffness matrix associated with the elastic coordinates \((K^e_{ij})\), shown in (2.110) are singular matrices, hence, a unique solution cannot be found. Thus, RC can be imposed by eliminating rows and columns of EoM corresponding to prescribed (fixed) elastic DoFs. More specifically, fixed elastic DoFs means that the nodal displacement of some nodes is zero in certain directions.

The choice of RC depends on the physics of the problem and the mechanical joints. It is natural, for an MBS, to not suppress body deformation in certain directions; it is important for the purpose of topology optimization of the MBS; hence the maximum number of fixed DoFs must be six in three dimensional problems and three in two dimensional problems such that the rigid body motion in all transitional and rotational directions is avoided. For instance, in a two dimensional problem, if the body has only two revolute joints, the choice of the RC is the same as the simply supported beam where the node at one joint is fixed and the node at another joint is allowed to move only in one direction. This RC is also used for a body with one revolute and one slider joint.

### 2.4.11.3. Solving DAE numerically using Sundials solver

Having found the terms of the system of Differential Algebraic Equations (DAE) (2.91) it can be seen that this system is highly nonlinear where there is coupling between reference and elastic coordinates. Solving such a system is not trivial. There are several numerical methods for solving nonlinear system of algebraic differential equations [6]. Numerical procedure of solving EoM is not in the scope of this thesis work. The reader is referred to [6] and [8] for detail explanation.

DAE (2.91) must be converted to an initial-value problem in the form

\[
RF(t, y, \dot{y}) = 0, \quad y(t_0) = y_0, \quad \dot{y}(t_0) = \dot{y}_0 \tag{2.141}
\]

where, \(RF\) is called residual function, \(t\) is the time, \(y\) is the unknown variable, \(\dot{y}\) is the time derivative of the unknown variable, \(t_0\) is the initial time, \(y_0\) and \(\dot{y}_0\) are the initial values of the unknown that need to be known at time \(t_0\). DAE (2.91) has a second order differential equation which needs to be converted to the first order. Introducing new variable \(qd^i\) which denotes the time derivative (velocity) of the generalized coordinates, DAE (2.91) can be converted to the form (2.141) as follows.

\[
F(t, y, \dot{y}) = \begin{cases} \begin{bmatrix} qd^i - q^i \\ M^i(qd^j) + K^i q^i + C^i_q \lambda - Q^e_i - \dot{Q}^e \end{bmatrix} = 0 & i = 1, 2, \ldots, n_b \\ C(q, t) = 0 \end{cases} \tag{2.142}
\]

Hence, variables \(y\) and \(\dot{y}\) are

\[
y = \begin{bmatrix} q \\ qd \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{q} \\ \dot{qd} \end{bmatrix} \tag{2.143}
\]

where, \(q = [(q^1)^T, (q^2)^T, \ldots, (q^{n_b})^T]^T\) and \(qd = [(qd^1)^T, (qd^2)^T, \ldots, (qd^{n_b})^T]^T, n_b\) is the number of bodies in MBS.
Sundials solver and IDA code (SundialsTB in Matlab and Assimulo in Python [24]) is used to solve DAE (2.142). IDA employs Backward Differentiation Formula (BDF) as a solution method with variable order and variable step size. In BDF, the time derivative of the unknown variable at time $t_n$ is approximated according to [8]

$$\dot{y}_n = \frac{\sum_{i=0}^{Q} a_{n,i} y_{n-i}}{h_n} \tag{2.144}$$

where, $Q$ is the order of method which ranges from one to five, $y_n \approx y(t_n)$, $\dot{y}_n \approx \dot{y}(t_n)$, $h_n = t_{n+1} - t_n$ is the step size and $a_{n,i}$ is a function of the step size history and the method order. Inserting the approximation of $\dot{y}_n$ to (2.142) for time $t_n$, DAE is converted to a nonlinear algebraic system of equations to be solved at each time step by Newton method:

$$F\left(t_n, y_n, \frac{\sum_{i=0}^{Q} a_{n,i} y_{n-i}}{h_n}\right) = 0 \tag{2.145}$$

IDA requires that consistent initial values which satisfy residual (2.142) are provided:

$$F(t_0, y_0, \dot{y}_0) = 0 \tag{2.146}$$

otherwise, numerical integration may fail at initial step. However, for all examples tested in this thesis work consistent initial conditions are not needed for elastic coordinates, so all given initial deflections are zero even though they may actually not be.

IDA has several options for setting the integration method and contributing variables such as Relative Tolerance, Absolute Tolerance, Maximum Number of steps, Variable Types, etc. All options have default values. The most important option variable is ‘VariableTypes’ that depends on the index of DAE.

### 2.4.11.4. DAE Index

Algebraic equations of EoM are results of constraints on bodies motion of an MBS. The constraints equations (2.132) and (2.136) are associated with the position of an arbitrary point of a body. A DAE with such algebraic equations is called Index 3 DAE. On the other hand, if the constraints are put on the velocity of an arbitrary point or on the time derivative of the generalized coordinates, the DAE is index 2.

Assume that the Index 3 constraints are given by $C(q, t) = 0$; to find Index 2 constraints differentiation of $C$ with respect to the time is needed:

$$\frac{\partial C(q, t)}{\partial t} = C_q \dot{q} + C_t = 0 \tag{2.147}$$

where, $C_t$ is the partial derivative of the vector of constraint functions with respect to the time. Note that since the trajectory and revolute constraints given by (2.133) and (2.136) do not explicitly depend on time, $C_t = 0$. For an Index 2 DAE, the algebraic equations in (2.142) must be replaced by (2.147).

IDA does not understand by itself whether the problem is Index 2 or 3. In an index 3 problem the error check must be suppressed for velocity and algebraic variables and, in an index 2 problem, it must be suppressed for only
algebraic variables. This can be done by setting the 'VariableTypes' option and then setting 'suppressAlgVars' to 'on'.

2.4.12. Modal reduction

The number of unknowns or the residual functions of EoM is very high for an MBS with deformable bodies, since the number of elastic coordinates is much larger than the number of reference coordinates. Moreover, for topology optimization, the number of Finite elements of a discretized body must be large, in order for the optimization problem to converge to an acceptable topology. Furthermore, for topology optimization of bodies within an MBS which is the subject of this thesis work, the EoM must be solved in every optimization iteration (see Chapter 4). As it was mentioned before, equation (2.145) must be solved in every time step with the Newton method; hence, evaluation of the Jacobian of the residual function is necessary once in one or several time steps. Calculating the Jacobian matrix needs residual function to be evaluated twice the number of unknowns. Thus, the simulation time increases considerably with the number of unknowns. In order to solve EoM in a reasonable time the number of unknowns must be reduced. It is possible by using a so called modal reduction or coordinate reduction method. It should be noted that the modal reduction is implemented together with a kind of substructuring method and mostly static condensation, a method so called Craig-Bampton-reduction method. There is no static condensation implemented in this thesis work. In the following modal reduction is described.

The differential equation of EoM can be partitioned for reference and elastic coordinates as follows.

\[
\begin{bmatrix}
\mathbf{m}_r^i & \mathbf{m}_{rf}^i \\
\mathbf{m}_{fr}^i & \mathbf{m}_{ff}^i
\end{bmatrix}
\begin{bmatrix}
\ddot{\mathbf{q}}_r^i \\
\ddot{\mathbf{q}}_{fr}^i
\end{bmatrix}
+ \begin{bmatrix}
0 & 0 \\
0 & \mathbf{K}_{ff}^i
\end{bmatrix}
\begin{bmatrix}
\mathbf{q}_r^i \\
\mathbf{q}_{fr}^i
\end{bmatrix}
= \begin{bmatrix}
(Q_{r}^i)_{r} \\
(Q_{fr}^i)_{r}
\end{bmatrix}
+ \begin{bmatrix}
(Q_{r}^i)_{f} \\
(Q_{fr}^i)_{f}
\end{bmatrix}
- \begin{bmatrix}
\mathbf{C}_{qr}^i \\
\mathbf{C}_{qfr}^i
\end{bmatrix}\lambda
\tag{2.148}
\end{align}
\]

Assuming that the body vibrates freely around its reference configuration, the right hand side of (2.148) vanishes which gives the homogenous form of (2.148). Considering only the elastic coordinates, the homogenous equation yields

\[
\mathbf{m}_{ff}^i \ddot{\mathbf{q}}_{fr}^i + \mathbf{K}_{ff}^i \mathbf{q}_{fr}^i = 0
\tag{2.149}
\]

A trial solution for (2.149) is given by \( \mathbf{q}_{fr,k}^i = \mathbf{a}_k^i e^{i\omega_k t} \), where, \( \omega_k \) is the vibration frequency, \( t \) is the time, \( j = \sqrt{-1} \) and \( \mathbf{a}_k^i \) is the vector of vibration amplitudes of elastic DoFs which vibrate with frequency \( \omega_k \) [2]. Substituting it in (2.149) gives an eigenvalue problem that can be solved for eigenvalues \( \omega_k \) and eigenvectors (mode shapes) \( \mathbf{a}_k^i \).

\[
(\omega_k)^2 \mathbf{m}_{ff}^i \mathbf{a}_k^i = \mathbf{K}_{ff}^i \mathbf{a}_k^i
\tag{2.150}
\]

\( k = 1, 2, ..., n_f \), where, \( n_f \) is the total number of the elastic DoFs of the deformable body \( i \). Note that the RC is already imposed so the fixed DoFs are already eliminated from \( \mathbf{m}_{ff}^i \) and \( \mathbf{K}_{ff}^i \). The exact solution for the homogenous equation (2.149) is the sum of all \( \mathbf{q}_{fr,k}^i \) for \( k = 1, 2, ..., n_f \). The reduced model can be obtained by considering only \( n_m \) eigenvectors corresponding to \( n_m \) lowest eigenvalues of (2.150), where, \( n_m < n_f \). Coordinate transformation from the physical elastic coordinates to modal elastic coordinates is expressed by

\[
\mathbf{q}_{fr}^i = \mathbf{B}_{m}^i \mathbf{p}_{fr}^i
\tag{2.151}
\]
where, $B_m^i$ is a matrix whose columns are the $n_m$ low frequency eigenvectors and $p_j^i$ is modal elastic coordinates.

$n_m$ must be selected such that the IDA solver gives a solution with an acceptable approximation. After solving EoM, the physical real coordinates can be retained using (2.151).
3. Design of the Lower A-arm of Double Wishbone suspension system in Abaqus/TOSCA using Dymola

3.1. topology optimization of a body subjected to transient loads

Structural optimization concerns static loads. The problem definition of topology optimization of a body subjected to transient loads is different, especially when the direction of the loads is changing. When a load is applied to a body, the body does not response instantaneously. It takes some time for the body to reach the steady state. If the load is transient, the body response changes accordingly before reaching the steady state; so, the response is time dependent. The transient loads result in vibrations and wave propagation in the body. This is called forced response that exhibits a phenomenon called beating which is most prominent if the damping is low and the excitation frequency is close to one of the natural frequencies. One strategy to do topology optimization on such a body under transient loads is assuming that the forces are quasi statics. This assumption means that there is enough time for the body to rest before the load changes; in other words, the body experiences several distinct load cases during operation. Hence if an optimized topology is needed to be found, all load cases must be considered. However this assumption is not realistic, since inertia forces are neglected which have an important role in parts with high accelerations, also the way of choosing the proper load cases is tricky. Finally, since the body shape and weight change in every optimization iteration, in case if the transient loads depend on the design for instance in a multibody system, the dynamic behavior and forces at joints change accordingly, hence the load cases are not valid anymore. On the other hand, assumption of quasi static loads may be valid with some approximations provided that proper load cases are selected and the transient loads do not depend on the design, and also if the effect of inertia forces is negligible. A big advantage of this method is its numerical efficiency.

To clarify the idea how to find an optimized topology considering all load cases see Fig. 3.1. Topology optimization is very sensitive to the direction of a load case. In general the magnitude is not important; the force magnitude can be scaled without changing the optimization result if only one load case is considered for optimization.

As it is shown in Fig. 3.1 the optimization result is different for different load cases. In other words, similar to optimization problem (2.1), denoting \( C_i(X,Z) \) as the objective function corresponding to load case \( i \), where, \( X \) is the design variable and \( S \) is the state variable, all \( C_i(X,Z) \) cannot have minimum at the same \( X^* \). In order to get a structure that is the ‘best’ for all load cases obviously merging all topologies would not be an option, since the volume constraint would be violated. However, there is a way to find a design so called Pareto optimality that satisfies all of the objectives and constraints ‘better’ than any other design [11]. This can be achieved by defining an optimization problem with a scalar objective function defined by

\[
C(X,Z) = \sum_{i=1}^{L} w_i C_i(X,Z) \tag{3.1}
\]

where, \( L \) is the total number of objective functions (also equals to the total number of load cases here) and \( w_i \) is a weight factor that shows the importance of the objective function. Solution of the topology optimization with the objective function defined in (3.1) gives a design which is close to the optimal design when the objective function is
a single $C_i(X, Z)$ The result of such an optimization where $C_i(X, Z)$ is the compliance of the structure corresponding to load case $i$ and $w_i = 1$ is shown in Fig. 3.1.

![Fig. 3.1:](image)

Fig. 3.1: (left) A plate with a fixed left boundary where a static load is applied on the lower right end. The uniform initial thickness is $h_0$. (right) The topology optimization result, minimization of compliance for different load cases. The final volume is 0.4 of the initial one. At the bottom left of the figure the optimized topology considering all load cases is shown; the bottom right is the optimized topology considering only the two last load cases.

Another difficulty in treating transient loads is proper selecting of the load cases. In order to select the load cases properly the transient loads acting on the part during the operation must be known. Then the maximum loads in different directions at different time instances must be selected. Each load case should correspond to only one time instant [12].
3.2. Topology optimization of the Lower A-arm of Double Wishbone suspension system

With the given introduction in section 3.1 a more practical example of topology optimization of a part subjected to transient loads can be treated. The part is the Lower Arm of Double Wishbone suspension system shown in Fig. 3.2.

![Fig. 3.2: A prototype of Lower A-arm within the Double Wishbone suspension system before applying topology optimization.](image)

The goal is to design a Lower A-arm with the given amount of material such that it shows the least deformation as the reaction to the loads applied on its joints during operation. This can be done by minimizing the compliance or strain energy of the body subject to volume or weight constraint. The problem is three dimensional. Topology optimization is done in Abaqus 6.13-4/TOSCA. The loads are collected from dynamical simulation of the Double Wishbone suspension system using Vehicle Dynamics Library (VDL) in Dymola. In the following the steps of designing the Lower A-arm is described in details.

3.2.1. Choosing the proper load cases coming from dynamical simulation of VDL

The Double Wishbone suspension system shown in Fig. 3.2 is created in Catia 2014x. It has already a primary design of the Lower A-arm. The model is exported to Dynamic Behavior Modeling (DBM) environment of Catia for dynamical simulation. DBM is in fact a built-in Dymola kernel accessible from Catia environment. Vehicle Dynamics Library (VDL) in DBM is used for rigid body dynamical simulation of the suspension system. The suspension system is simulated when the vehicle runs on an uneven road while steering and braking. The calculated transient forces at the joints of the Lower A-arm are shown in Fig. 3.3. The name of the joints is shown in Fig. 3.2. It is assumed that such information is already available. The details for calculating the forces are not the subject of this thesis report.
Fig. 3.3: The transient loads acting on the joints of the primary designed Lower A-arm. VDL is used for calculating the forces when the vehicle runs on an uneven road including steering and braking. Braking starts at time = 9 s. There is no torque at joints.

The sum of the forces in each direction is not zero. This shows that the part always experiences some acceleration which is neglected considering the assumption of semi static loads.

Since the body is assumed to be isotropic, the load cases are selected based on the pick loads in different directions. For instance, in Fig. 3.3 up to time = 9 s, nine of the loads reach their maximum value at time = 4.2 s while the other three loads reach their pick value at time = 0.9 s. Hence the loads at these two time instances before time = 9 s can be selected as two load cases. At time = 9 s the braking starts. Almost all loads reach a maximum at time = 9.3 s. Since maximum values at time = 9.3 s are global maximums, load case 3 is enough for doing topology optimization. However, in this example topology optimization is done considering all three load cases.

It should be noted that a load case is a collection of loads and boundary conditions used to define a particular loading condition [13]. The information coming from DBM contains only the load values and there is no clue about the boundary conditions. Moreover, dynamical simulation in VDL can be done only for rigid bodies; so, the calculated forces are not exact.

### 3.2.2. Defining the geometry of the first design space

Design space is defined based on the available space for the Lower A-arm in the vehicle and suspension system. The position and the design of the joints are fixed; so, they must be excluded from the topology optimization. The primary design in Catia 2014x is converted to Catia V5 version and then is exported to Abaqus. Considering the primary design the new design space is shown in Fig. 3.4. The thickness is the same as the primary design but in general it could be bigger.
3.2.3. Defining the material properties and meshing the design space

A typical setup for a mechanical problem is Abaqus needs to be done. The part is a solid, Homogeneous, 3D and flexible. The material is chosen to be elastic with the density 7800, Young’s module 2e11 and poisson’s ratio 0.3.

Abaqus provides many options for defining the mesh. The design space here is meshed to 92286 quadratic tetrahedral elements shown in Fig. 3.5. An element corresponds to a topology optimization design variable.

3.2.4. Defining sets, constraints, loads and boundary conditions

Some parts of the geometry of the design space shown if Fig. 3.4 are of special interest. These parts are the nodes or surfaces where forces act and also the geometry of the joints which must be excluded from topology optimization. Each of these parts can be specified as a set in Abaqus. These sets can be treated in special manners in setting up the problem later.

A joint force found in VDL is acting only at one point in the middle of the joint. So, a point as a set is defined at the middle of the joint and then the Degrees of Freedom of the surrounding surface which is the inner surface of the joint is constrained to Degrees of freedom of the middle point. This means that the point is rigidly attached to the
surrounding surface such that if a force acts on the point the value of the force is distributed on the nodes of the mesh on that surface. This is illustrated in Fig. 3.6. The same action needs to be done for all joints.

![Fig. 3.6: A kinematic coupling constraint between the middle point and inner surface of a joint.](image)

Another collection of sets that needs to be defined is the elements belonging to the joints geometry. These sets are shown in Fig. 3.7.

![Fig. 3.7: Sets which contains the elements belonging to the joints geometry.](image)

As it was previously mentioned there is no prescribed information about the boundary conditions of the Lower A-arm. The boundary conditions must be guessed by the aid of engineering experience. Some degrees of freedom must be fixed such that the rigid body motion is avoided. This is the tricky part of the setting the optimization problem. Topology optimization problem is very sensitive to loads direction and boundary conditions. In general various boundary conditions end up with different optimized topologies. In the current problem the forces at all joints is prescribed (weight is neglected). According to the data calculated in VDL there is no torque at joints. Hence, if it is assumed that the sum of all forces in different directions is zero and there is no torque, it can be concluded that the body is at rest. The rigid body motion can be eliminated by fixing at least 6 DoFs which are not identical. If the middle point of a joint is fixed, physically the same amount of the load must be produced at that point in order to maintain the equilibrium. Thus for a body at equilibrium with known forces at all connections it
should not matter which 6 DoFs to be fixed. Note that it is important to fix only 6 dissimilar DoFs; otherwise, deformation is certain direction might be suppressed. However, the lower A-arm is not at equilibrium since the sum of the forces is not zero showing that the body has accelerated motion. This is a drawback of assuming the transient force to be quasi static. Accepting the approximations, DoFs of Lower Ball joint shown in Fig. 3.2 is fixed as a boundary condition.

It should be noted that in Abaqus/CAE the ‘Inertia relief’ feature can be added as single load acting on the part. Doing so, introducing the boundary condition would not be needed. It is possible to be done, for instance, for a stress analysis, the result of which will be shown at the end of this chapter. However, this feature is not applicable for sensitivity based topology optimization in TOSCA.

After defining the boundary condition where a joint is fixed, the value of the forces at remaining joints found in VDL is needed to define loads in Abaqus. It is very important that the body coordinate system in Abaqus be exactly the same as the published coordinate system used in VDL for calculating the loads. Otherwise, the load values given to Abaqus are wrong resulting in a wrong topology optimization. There are three joints that are loaded in three different times so nine loads must be defined. Note that the loads at the fixed joint are not needed. A load case is a collection of loads and boundary conditions used to define a particular loading condition. Here the collection of boundary condition and three loads, which are acting at three joints at the same time, constitute one load case. The boundary condition is common between all load cases.

3.2.5. Defining the optimization task and running topology optimization

The topology optimization setup in Abaqus/TOSCA starts by defining the optimization task. Topology optimization task is chosen on the whole design space shown in Fig. 4.4 with the frozen load and boundary condition regions. The value of the Young’s module and density does not affect the topology optimization result of a static body. Because what is important in topology optimization is the relative deformation value of the elements not the value itself; this argument can be concluded from (2.57) and (1.58); but, this is not true for topology optimization of a dynamical system suggested in chapter 4 of this thesis work. In a 3D optimization problem the design variable is density (or the normalized density) of the material that changes between a value close to zero (0.001) and 1. Convergence Criteria is needed to stop optimization iteration. Two types of Convergence Criteria can be specified. First one is the change of the value of the objective function from one iteration to the next. The default value is 0.001. Second one is the average change in the design variable. The default value is 0.005.

The design responses are needed to be defined. A design response is used as an objective function or a constraint function. Two design responses are defined: strain energy and volume. Load cases must be specified while defining the strain energy.

The objective function is the design response-1, the strain energy, which needs to be minimized. The constraint is the design response-2, the volume. The volume of the final topology must not be bigger than 0.45 of the initial volume. The choice of the fraction of the initial value depends on the amount of the material that needs to be removed.

The sets of the elements belonging to the joints geometry defined previously should be used to specify the geometry restriction of the topology optimization. Frozen elements of the body remain intact during topology
optimization. There are different options for geometry restrictions. The manufacturing restriction or Demold control can also be employed.

For further information about setting up the optimization task see [13].

After finishing the optimization setup the optimization task can be submitted to the job and then can be run. The optimization will proceed and the iterations will stop if one of the convergence criteria is satisfied. The optimization result after 40 iterations is shown in Fig. 3.9. The history of the volume fraction and the objective function, strain energy, is depicted in Fig. 3.10.

![Optimization Result](image)

**Fig. 3.9:** The optimization result of Lower A-arm after 41 iterations. 3 load cases are considered. The color illustrates the value of the normalized material property that is the same as the design variable. It changes between 0.001 and 1.

![Volume Fraction and Objective Function](image)

**Fig. 3.10:** The history of the volume fraction, on the left, and the objective function, strain energy, on the right; volume converges to 0.45 of the initial volume and strain energy is minimized.

The optimized topology shown in Fig. 3.9 is the 'best' topology considering all 3 load cases. For a single load case the optimized topology will be different. For instance, if only the first two load cases shown in Fig. 3.3 are considered meaning that the braking of the vehicle is not included, the result would be much different. Such an optimized topology is shown in Fig. 3.11.
Fig. 3.11: The optimization result of Lower A-arm. Only the first two load cases are considered.

It should be noted that if the thickness of the design space had been chosen bigger, most of the material would have been removed from the inner part of the body giving different topology from what is shown in Fig. 3.9.

The optimization can be continued by defining a new design space based on the found topology shown in Fig. 3.9 and by repeating all the steps of setting up the optimization task. The new design space is shown in Fig. 3.12, on the right, where the body boundaries are frozen in addition to the joints geometry. This time the volume of the final topology must not be bigger than 0.65 of the initial volume. Optimized topology is depicted in Fig. 3.12, on the left.

Fig. 3.12: on the right, a new design space where the body boundaries are frozen in addition to the joints geometry; on the left the optimal topology; All 3 load cases are considered.

3.2.6. Convert the result to a real part (verification model)

The results of topology optimization shown in Fig. 3.9 – 3.12 contain only information about the geometry through elements. In order to be able to verify the model and to do, for instance, stress analysis they need to be converted
to an actual part so called verification model [14]. In stand-alone TOSCA it is possible to convert the optimal topology directly to the verification model without altering the problem setup. This is not implemented in Abaqus/TOSCA yet. Instead, the surface of the part can be smoothened and exported to ‘stl’, ‘inp’ or ‘3dxml’ files in Abaqus/TOSCA and then can be imported again to Abaqus. Then the problem setup can be started form beginning to be able to do the stress analysis. Fig. 3.13 shows the stress analysis on the verification model where only load case 3 is acting on the part; the inertia relief is used so that no boundary condition needs to be applied. The max stress is 473.3 Mpa.

![Stress analysis on the verification model](image1)

**Fig. 3.13:** The stress analysis on the verification model where only load case 3 is acting on the part; the inertia relief is used so that no boundary condition is needed to be applied. The max stress is 473.3 Mpa.

### 3.2.7. Make an approximated part based on the available drawing commands

Finally, an approximated part which is created based on the optimal topology using available drawing commands or the smoothened optimal topology itself can be inserted back into the dynamical system, see Fig. 3.14 and 3.15.

![Approximated part](image2)

**Fig. 3.14:** An approximated part which is created based on the optimal topology using available drawing commands.
3.2.8. Comments on the method

- By making the assumption of quasi static loads, inertia forces are neglected which have an important role in parts with high accelerations in a dynamical system.
- The way of choosing the proper load cases is a problematic task [12].
- In order to introduce the boundary condition it was assumed that the body is in equilibrium which is a rough approximation. Since the body is not in total equilibrium, a change in boundary condition for instance, fixing a joint other than the Lower Ball joint, results in a different optimal topology.
- Since the body shape and weight change in every optimization iteration, in case if the transient loads depend on the design, for instance, in a Multibody system, the dynamic behavior and forces at joints change accordingly, hence the load cases are not valid anymore. For example, rigid body simulation of the suspension system shown in Fig. 3.15, where the optimal A-arm is used, will not give the same forces at joints as the system shown in Fig.3.2.
- Design process involves using DBM and TOSCA which needs exchange of data. For example, using the same coordinate system in both environments and file exchange between the two environments are central.
- Doing the whole design process needs user’s good knowledge and engineering skills. The design process must be done step by step working in different environments.
4. Topology Optimization of Multibody Systems

The Topology Optimization (TO) of static structures with fixed loading is a very attractive topic in structural mechanics that have found many applications in industrial design tasks. The extension of the theory to dynamic loading for designing a multibody system with bodies which are lighter and stronger can be of great interest. The objective of this thesis work is to investigate one of the possible ways of extending the static structural Topology Optimization to Topology Optimization of dynamical bodies embedded in a Multibody system. The SIMP method is introduced for TO of static parts. As an example of using SIMP with multi-criteria objective function, the design of a body within an MBS, the lower A-arm of a Double Wishbone suspension system, is shown in Chapter 3. The design was made by assuming that the reaction forces at connections are static. This assumption made it possible to use SIMP explained in Chapter 2 directly just by altering the objective function. Important load cases are extracted from the transient reaction loads calculated by rigid body simulation of MBS. However, such a method has some drawbacks which were the motivation of introducing another more systematic approach. A more systematic approach is to do TO of flexible bodies while they are operating in an MBS considering all transient reaction forces as well as inertia forces acting on the bodies during operation time. Direct applying SIMP to such a problem results in numerical difficulties and non-convergence. Moreover, calculating the sensitivity is numerically expensive. There has not been much research on this topic so far. In [15] almost the same approach is used with two different regime of stiffness penalization. The switching criteria between two regimes might be different for different problems, thus the presented formulation in [15] is not always valid. [7, 20] suggests numerical calculation of sensitivity using generalized alpha method; the example used in [7, 20] is size optimization with low number of design variables; so the problem of non-convergence which is observed only in topology optimization does not occur. Also in [9] only an example of shape optimization is presented. In general, no systematic approach is developed yet in literature to handle the problem of instability and non-convergence of topology optimization of the structure under dynamic loadings or topology optimization of multibody systems. A modification to SIMP is proposed here that helps to get convergence in any TO of MBS problem. In addition, sensitivity is approximated by eliminating terms which have low order of magnitude but numerically expensive to calculate; so, the optimal design can be found in a reasonable computation time in a problem with large number of design variables.

The idea is to solve the nonlinear equation of motion of MBS at each optimization iteration with the reduced coordinates, then retain the physical real coordinates and approximate the sensitivities and then minimize the objective function at iteration $k$ using the modified SIMP method, after that, update the design variables and solve the nonlinear equation of motion again but with the updated design variables. This process must be repeated until the convergence criterion is satisfied. This approach is described in details in following sections.

4.1. Defining and approximating the optimization problem

In Topology Optimization of a multibody system, the objective function can be defined in the same way as in (4.1) but considering all times:
\[ C(X, S) = \frac{1}{t_s} \int_{0}^{t_s} C(X, S, t) \, dt \] (4.1)

where, \( C \) denotes the objective function, \( X \) is the design variable, \( S \) is the state variable, \( t \) is the time and \( t_s \) is the upper limit of the integral that is the same as the simulation end time (stopping time) of MBS. Solving (4.1) numerically by discretizing the time interval to \( s \) steps yields

\[ C(X, S) \approx \frac{1}{(s + 1)} \sum_{i=0}^{s} C(X, S, t_i) \] (4.2)

where, \( t_i \) denotes \( i^{th} \) time step. When solving the nonlinear equation of motion with IDA solver, \( s \) is the number of the communication points. Noting that the solver uses variable step sizes, the communication points are the time steps when the solution of the EoM is demanded from the solver.

In Topology Optimization of a multibody system, equilibrium constraint means satisfying the nonlinear equation of motion. Considering the new objective function and the equilibrium constraint the optimization problem defined for static parts in (2.2) changes to the following problem for body \( j \).

\[
\begin{align*}
    \min_{X^j, q_{q}^j} \quad & C^j(X^j, q_{q}^j) = \frac{1}{(s + 1)} \sum_{i=0}^{s} C^j(X^j, q_{q}^j, t_i) \\
    \text{s. t.} \quad & M^j \ddot{q}^j + K^j q^j + C_{q}^j \lambda = Q_{e}^{j} + Q_{v}^{j} \quad i = 1, 2, \ldots, n_b \\
    & C(q, t) = 0 \\
    & g_i^j(X^j) = \int_{A^j} X^j \, da - V_{\text{max}}^j \leq 0 \\
    & X^j \in X^j = \{X^j, \text{min} \leq X^j \leq X^j, \text{max}\}
\end{align*}
\] (4.3)

where, the superscript \( j \) shows that the variable belongs to body \( j \); if the objective function is chosen to be the compliance, \( S = q_{q}^j \) is the elastic degrees of freedom of body \( j \) which are always chosen to be the nodal displacements. The best would be to let \( j = 1, 2, \ldots, n_b \) but there is no restriction to optimize just one body at a time which introduces some approximation.

The numerical solution of the nonlinear equation of motion gives \( q_{q}^j \) as a function of \( X^j \), which makes it possible to write the objective function as a function of only the design variable \( X^j \); doing so the optimization problem becomes:

\[
\begin{align*}
    \min_{X^j} \quad & C^j(X^j) = \frac{1}{(s + 1)} \sum_{i=0}^{s} C^j(X^j, t_i) \\
    \text{s. t.} \quad & g_i^j(X^j) = \int_{A^j} X^j \, da - V_{\text{max}}^j \leq 0 \\
    & X^j \in X^j = \{X^j, \text{min} \leq X^j \leq X^j, \text{max}\}
\end{align*}
\] (4.4)
Using the same approach explained in Chapter 2 to solve the optimization problem the objective function is approximated applying the Optimality Criteria (OC) method. In the same way as in (2.45), for ease of notation we drop the time argument in the expressions.

\[ C_i^j(X^j) \approx C_i^j(X^{j,k}) + \sum_{e=1}^{n} \frac{\partial C_i^j(X^{j,k})}{\partial Y_e^j(X^j)} \left( Y_e^j(X^j) - Y_e^j(X_e^{j,k}) \right) \]  

(4.5)

or,

\[ C_i^j(X^j) \approx C_i^j(X^{j,k}) + \sum_{e=1}^{n} \frac{\partial C_i^j(X^{j,k})}{\partial X_e^{j,k}} \frac{\partial X_e^{j,k}}{\partial Y_e^j} \left( Y_e^j(X^j) - Y_e^j(X_e^{j,k}) \right) \]  

(4.6)

where, according to OC method,

\[ Y_e^j(X_e^j) = \begin{cases} 
X_e^j & \text{if } \frac{\partial C_i^j(X^j)}{\partial X_e^j} > 0 \\
(X_e^j)^{-\alpha} & \text{if } \frac{\partial C_i^j(X^j)}{\partial X_e^j} \leq 0
\end{cases} \]  

(4.7)

where, \( X^{j,k} \) is the vector of design variables of body \( j \) at iteration \( k \) and \( X_e^j \) is the design variable of element \( e \) of the body \( j \).

### 4.2. Sensitivity analysis

In order to evaluate (4.6) it is necessary to compute the sensitivity of the objective function with respect to the design variable at iteration \( k \) at time \( t_1 \), \( \frac{\partial C_i^j(X^{j,k})}{\partial X_e^{j,k}} \). Calculating the sensitivity numerically is very expensive. In this thesis work, it is attempted to approximate the sensitivity by eliminating some terms of the analytical expression which are very difficult to find analytically or are expensive to calculate numerically but intuitively have minor effect on the total sensitivity. Doing so, the sensitivity of the dynamic response problem is similar to that in static response problem. Another alternative is to use Adjoint method [25] which is left for future investigations about TOMBS.

If \( C_i^j \) is the compliance of body \( j \) at time \( t_1 \):

\[ C_i^j(X^j) = \left( F_{f,i}^j(X^j) \right)^T \cdot q_{f,i}^j(X^j) \]  

(4.8)

the sensitivity can be calculated as follows
\[
\frac{\partial C_i^j(X^j)}{\partial X_e^j} = \frac{\partial}{\partial X_e^j} \left( (F_{ij}^j(X^j))^T \cdot q_{ij}^j(X^j) \right) \\
= \frac{\partial}{\partial X_e^j} \left( F_{ij}^j(X^j) \right)^T \cdot q_{ij}^j(X^j) + \left( F_{ij}^j(X^j) \right)^T \frac{\partial q_{ij}^j(X^j)}{\partial X_e^j} 
\]  
(4.9)

Where, \( F_{ij}^j(X^j) \) denotes equivalent static loads including the inertia forces acting on the nodal points of the body \( j \) at time \( t_i \) which produces the same displacement field as the dynamic loads; these loads are functions of the design variable \( X^j \); \( q_{ij}^j(X^j) \) is the vector of the nodal displacements of the body \( j \) at time \( t_i \) which is also a function of the design variable \( X^j \). \( \frac{\partial q_{ij}^j(X^j)}{\partial X_e^j} \) can be found by differentiating the equilibrium constraint with respect to the design variable. The differential equation of EoM of an MBS is given by (2.91):

\[
M^i \ddot{q}^i + K^i q^i + C_{q}^{iT} \lambda = Q_e^i + Q_v^i \quad i = 1,2, ..., n_b 
\]  
(4.10)

Differentiating (4.10) with respect to \( X^j \) gives:

\[
\frac{\partial M^i}{\partial X_e^j} \ddot{q}^i + \frac{\partial M^i}{\partial X_e^j} \dot{q}^i + K^i \frac{\partial q^i}{\partial X_e^j} + K^i \frac{\partial q^i}{\partial X_e^j} + C_{q}^{iT} \frac{\partial \lambda}{\partial X_e^j} = \frac{\partial Q_e^i}{\partial X_e^j} + \frac{\partial Q_v^i}{\partial X_e^j} \quad i = 1,2, ..., n_b 
\]  
(4.11)

Rearranging the terms of (4.11) yields

\[
M^i \ddot{q}^i + K^i \dot{q}^i + C_{q}^{iT} \lambda = F_p^i \quad i = 1,2, ..., n_b 
\]  
(4.12)

where, \( F_p^i \) is set to

\[
F_p^i = \frac{\partial Q_e^i}{\partial X_e^j} + \frac{\partial Q_v^i}{\partial X_e^j} - \frac{\partial M^i}{\partial X_e^j} q^i - \frac{\partial K^i}{\partial X_e^j} \dot{q}^i - \frac{\partial C_{q}^{iT}}{\partial X_e^j} \lambda \quad i = 1,2, ..., n_b 
\]  
(4.13)

Differential equation (4.12) together with the constraint equations \( C(q, t) = 0 \) has the same form as the EoM (2.91) which can be solved using Sundial’s integrator IDA with the same settings provided that an explicit expression of the \( F_p^i \) is found. However, Equation (4.12) is needed to be solved for all times \((s+1)\) and for every element of the body. For topology optimization the number of design variables or finite elements often is large. Large number of design variables and time steps make finding the sensitivity very expensive. To overcome this problem the sensitivity must be approximated by eliminating small terms in (4.12) and (4.13). These terms are \( \frac{\partial \lambda}{\partial X_e^j} \), \( \frac{\partial q_{e}^i}{\partial X_e^j} \), \( \frac{\partial q_{v}^i}{\partial X_e^j} \), \( \frac{\partial M^i}{\partial X_e^j} \), \( \frac{\partial K^i}{\partial X_e^j} \), \( \frac{\partial C_{q}^{iT}}{\partial X_e^j} \). Assuming that these terms are small and also assuming that the sensitivity of the elastic coordinates of a body, for instance body \( j \), does not depend on other bodies \( \frac{\partial X_e^i}{\partial X_e^j} = 0 \) and \( \frac{\partial q_{e}^i}{\partial X_e^j} = 0 \) if \( i \neq j \), then (4.12) is simplified to
\[ \frac{\partial K^j_{ff}}{\partial X^j_e} \mathbf{q}^j_f + K^j_{ff} \frac{\partial \mathbf{q}^j_f}{\partial X^j_e} = \mathbf{0} \tag{4.14} \]

where, \( K^j_{ff} \) is the global stiffness matrix of body \( j \) associated with the elastic coordinates. Equation (4.14) is the same as (2.49) where the static equilibrium equation was differentiated with respect to the design variable but when forces are not design dependent.

Solving (4.14) for \( \frac{\partial \mathbf{q}^j_f}{\partial X^j_e} \) gives:

\[ \frac{\partial \mathbf{q}^j_f}{\partial X^j_e} = -K^j_{ff}^{-1} \frac{\partial K^j_{ff}}{\partial X^j_e} \mathbf{q}^j_f(X^j) \tag{4.15} \]

Substituting (4.15) in (4.9) gives an expression for the sensitivity of the objective function, compliance, with respect to the design variable.

\[ \frac{\partial C^j_i(X^j)}{\partial X^j_e} = \frac{\partial (F^j_{f,t}(X^j))^T}{\partial X^j_e} \mathbf{q}^j_f(X^j) + (F^j_{f,t}(X^j))^T \left( -K^j_{ff}^{-1} \frac{\partial K^j_{ff}}{\partial X^j_e} \mathbf{q}^j_f(X^j) \right) \tag{4.16} \]

where, \( \mathbf{q}^j_{f,t}(X^j) \) is the elastic coordinates or nodal displacements of the body \( j \) with respect to the body coordinate system at time \( t \), which is a function of the design variable. With the above assumptions the first term of the right hand side of (4.16) is close to zero. As it was mentioned before \( F^j_{f,t}(X^j) \) can be assumed to be all the forces including the inertia forces acting on the nodal points of the body \( j \) at time \( t \); it can be defined as follows.

\[ \begin{bmatrix} F^j_{r,t} \\ F^j_{f,t} \end{bmatrix} = \mathbf{Q}^j_t + \mathbf{Q}^j_t \dot{\mathbf{q}}^j_t - \mathbf{C}^j_{q,i} \lambda \tag{4.17} \]

Considering that the rows and columns of the stiffness matrix corresponding to the reference coordinates in (4.10) are zero, Equation (4.10) for body \( j \) and for elastic coordinates can be written as

\[ \mathbf{q}^j_{f,t}(X^j) = K^j_{ff}^{-1} F^j_{f,t}(X^j) \tag{4.18} \]

Using (4.18) in (4.16) the approximated sensitivity is found to be

\[ \frac{\partial C^j_i(X^j)}{\partial X^j_e} = -\mathbf{q}^j_{f,t} \dot{\mathbf{q}}^j_t \frac{\partial K^j_{ff}}{\partial X^j_e} \mathbf{q}^j_{f,t}(X^j) \tag{4.19} \]

Another way to derive (4.19) is by defining the compliance of the body \( j \) at time \( t \), according to (4.8) and (4.18)', in the form

\[ C^j_i(X^j) = \mathbf{q}^j_{f,t} \dot{\mathbf{q}}^j_t K^j_{ff} \mathbf{q}^j_{f,t}(X^j) \tag{4.20} \]

where, the symmetry of \( K^j_{ff} \) is considered; differentiating (4.20) with respect to \( X^j_e \):

69
\[ \frac{\partial c_i^j(X_i)}{\partial x_e} = \frac{\partial q_{j,1}^i(X_i) \cdot K_{ff}^i \cdot q_{j,1}^i(X_i)}{\partial x_e} + q_{j,1}^i(X_i) \cdot \frac{\partial K_{ff}^i}{\partial x_e} \cdot q_{j,1}^i(X_i) \]

(4.21)

And finally substituting (4.15) in (4.21) for the time \( t_i \) gives the same expression for the sensitivity as given in (4.19).

\( K_{ff}^i \) is built by assembling the element stiffness matrices \( k_e^j \) of the body \( j \) shown in (2.41); hence

\[ \frac{\partial K_{ff}^i}{\partial x_i} = \begin{cases} k_{e,0}^i & \text{if } i = e \\ 0 & \text{if } i \neq e \end{cases} \]

(4.22)

where, \( k_{e,0}^i \) is the same as \( k_e^i \) given in (2.42) but with global numbering which is a matrix with all entries to be zero except the entries corresponding to the global nodal numbers of the element \( e \). Finally,

\[ \frac{\partial c_i^j(X_i)}{\partial x_i} = -q_{j,1}^i \cdot K_{ff}^i \cdot q_{j,1}^i(X_i) \]

(4.23)

### 4.3. Optimality Criteria approximation of the compliance of a dynamic body

Because of the assumption of linear elasticity \( k_e^{i,0} \) is a constant positive definite matrix, hence, (4.23) gives a negative value and according to (4.7) the choice of the intermediate variable is \( (X_e^j)^{-\alpha} \). Substituting (4.23) and (4.7) into (4.6) and eliminating the constant terms gives

\[ C_i^{i,k,OC}(X_i) = \sum_{l=0}^{n} \sum_{e=1}^{n} q_{j,1}^l \cdot (X_{e}^{i,k}) \cdot k_{e,0}^i \cdot q_{j,1}^i(X_{e}^{i,k}) \cdot (X_{e}^{i,k})^{1+\alpha} \cdot (X_{e}^{i})^{-\alpha} \]

(4.24)

This is a desired approximation of (4.4). The equivalent expression to (4.24) for static parts is given in (2.56). (4.24) can be written in the form

\[ C_i^{i,k}(X_i) \approx C_i^{i,k,OC}(X_i) = \sum_{e=1}^{n} \sum_{l=0}^{n} b_e^k(X_e^l)^{-\alpha} = \sum_{e=1}^{n} b_e^k(X_e^l)^{-\alpha} \]

(4.25)

where,

\[ b_e^{i,k} = q_{j,1}^l \cdot (X_{e}^{i,k}) \cdot k_{e,0}^i \cdot q_{j,1}^i(X_{e}^{i,k}) \cdot (X_{e}^{i,k})^{1+\alpha} \cdot (X_{e}^{i})^{-\alpha} \]

(4.26)

and

\[ b_e^k = \sum_{l=0}^{n} b_e^{i,k} \]

(4.27)
The sub-problem approximation of the optimization problem (4.4) now can be written in the form:

\[
\begin{align*}
\min_{\chi^j} \quad & C_{j,k,OC}(\chi^j) = \sum_{e=1}^{n} b_{e}^k (\chi_{e}^j)^{-a} \\
\text{s. t.} \quad & g_{e}^j(\chi^j) = \sum_{e=1}^{n} a_{e}^j \chi_{e}^j - V_{max} \leq 0 \\
\chi^j \in & \chi^j = \{\chi_{j,min} \leq \chi^j \leq \chi_{j,max}\}
\end{align*}
\] (4.28)

The optimization problem given by (4.28) for a dynamic body has the same form as the one for a static body defined in (2.58) but with different \( b_{e}^k \); see (4.25). This problem can be solved using Lagrangian Duality method and SIMP; see sections 2.3.2 and 2.3.3. SIMP changes \( b_{e,k}^l \) defined in (4.26) to

\[
b_{e,k}^l = q_{f,l}^{\perp}(X^{l,k}) q (X_{e}^{l,k})^{q-1} k_{e}^{l} q_{f,l}^{\perp}(X^{l,k}) (X_{e}^{l,k})^{1+a} \quad (4.29)
\]

### 4.4. Modification to SIMP method in TOMBS

In section (4.3) it was shown that the problem of the Topology Optimization of a Multibody System (TOMBS) can have the same form of the Topology Optimization problem of a static body, if some approximations are made for calculating the sensitivity. TOMBS problem can be solved using Lagrangian Duality method and SIMP. Using SIMP directly in TOMBS may result in non-convergence of the optimization algorithm in MBS. The reason of non-convergence is appearing an effect which is called ‘Flying Elements’ in this thesis work; see Section (5.1.1) for an illustrative example. In the following the reason of non-convergence is explained in details.

Imagine two harmonic oscillators each consisting of a spring and a mass attached to it. Each system has one degree of freedom. The mass for both systems is the same. The first spring is soft and the other one is stiff. If both systems are disturbed with the force with the same magnitude the first system vibrates with higher amplitude than the second one. This is what is happening in TOMBS when the elements’ stiffness is penalized in SIMP. In this case, the stiffness of an element might be different from the neighboring elements; hence, some elements may experience higher displacement than the other elements caused by the inertia forces. Such an element with high displacement is called ‘Flying Element’ in this thesis work. In flexible multibody model used in this thesis work the uniform mass distribution is converted to the lumped mass distribution. An alternative explanation is via lumped masses. The lumped masses are located in nodal points of the finite element mesh. Schematically the lumped masses are connected with springs shown in Fig. 4.1. By penalizing the stiffness of the elements around a lumped mass the lumped mass is not strongly attached to the body any more. Thus when the body experiences acceleration the mass does not follow the body’s trajectory. In the following it is explained why this effect cause non-convergence of the optimization algorithm and how to overcome this problem.
Fig. 4.1: The uniform mass distribution is converged to the lumped mass distribution. In finite element model with four-node rectangular elements the lumped masses are connected with springs as is shown in the figure.

It was shown in section 2.3.2 that in the optimization algorithm the convergence criterion can satisfy if (superscript \( j \) is omitted for simplification of the notations)

\[
X^{k+1}_e = X^k_e \quad \text{for } e = 1, \ldots, n
\]  

(4.30)

In case of disabling the box constraint and using equations (2.61) and (2.62), it can be shown that

\[
X^{k+1}_e = \left( \frac{\alpha b^k_e}{\lambda a_e} \right)^{\frac{1}{1+\alpha}}
\]  

(4.31)

Substituting \( b^k_e \) from (4.27) in (4.31) and considering the SIMP and (4.30) at convergence,

\[
q \left( X^{j,k}_e \right)^{q-1} \sum_{i=0}^e q_{f,i}^T (X^{j,k}_e) k_e^{j,0} q_{f,i}(X^{j,k}_e) = 1
\]  

(4.32)

rearranging the terms

\[
\sum_{i=0}^e q_{f,i}^T (X^{j,k}_e) k_e^{j,0} q_{f,i}(X^{j,k}_e) = \frac{\lambda a_e}{q \left( X^{j,k}_e \right)^{q-1}} = S_e
\]  

(4.33)

If there is no box constraint, the convergence occurs when (4.33) is nearly satisfied. The left hand side of (4.33) can be regarded as the sum of the strain energies per volume of the element \( e \) of the body \( j \) obtained at different time instances during the operation. It can be shown that

\[
\begin{cases} 
S_e \approx \frac{\lambda a_e}{q} & \text{if } X^{j,k}_e \approx 1 \\
S_e \gg \frac{\lambda a_e}{q} & \text{if } X^{j,k}_e \approx 0
\end{cases}
\]  

(4.34)

It can be concluded that at the first iteration \( (k = 1) \) all \( X^{j,k}_e \) has the same value and accordingly the same \( S_e \). Elements with high displacement (where the sum of strain energies is higher than \( S_e \)) must become thicker at the
next iteration in order to reduce the displacement to try to satisfy (4.33); similarly, elements with low displacement (where the sum of strain energies is lower than $S_e$) must become thinner in order to increase the displacement at the next iteration to try to satisfy (4.33). Existence of the term $(X_{e,k}^j)^{q-1}$ in the denominator of (4.33) introduced by SIMP helps the thickness to increase or to decrease more in one iteration resulting in a faster convergence. At the second iteration $S_e$ is not the same for all elements. It has a higher value for thinner elements, so thin elements need to become again thinner in order to try to satisfy (4.33). On the other hand, thick elements have small $S_e$, hence, they become again thicker at the next iteration. This process can be continued until all elements reach the box limits. In an MBS displacement depends on the stiffness as well as the mass and inertia forces. Penalizing $k^j_i$ in SIMP causes thin elements to have very high elasticity or low stiffness. Thus, highly elastic elements with a constant mass experience high displacement. In case of having high acceleration and inertia forces the effect of ‘Flying Elements’ is observed. High displacement of the thin elements results in high strain energy even higher than $S_e$, thus, the thickness is increased instead of being decreased at the next iteration causing oscillation of the thickness continuously and hence non-convergence. This problem can be cured by penalizing not only the element stiffness but also the element mass (or lumped masses) or can be reduces by increasing the lower bound of the thickness or penalizing the stiffness more slightly for thin elements. It will be shown in Chapter 5 that penalizing the lumped masses gives a better descend in objective function during minimization. Thus the modification to SIMP which is done in this thesis work is

$$E'^j_e = (X^j_e)^{q-1}E^j_i \quad \text{and} \quad M'^j_p = \min \left( (X^j_{e\in N_p})^{q-1}M'^j_{p,0} \right)$$  \hspace{1cm} (4.35)$$

where, $E^j_i$ is the initial Young’s modulus shown in (2.10), $E'^j_e$ is the penalized Young’s modulus, $M'^j_{p,0}$ is the initial lumped mass of the node number $p$, $M'^j_p$ is the penalized lumped mass, $q$ is the penalization factor and $N_p$ is a set containing the elements numbers which share the node $p$. With this modification $X^j_e$ in both element mass and element stiffness matrices changes to $(X^j_e)^q$. Instead of lumped masses element masses can be penalized:

$$E'^j_e = (X^j_e)^{q-1}E^j_i \quad \text{and} \quad M'^j_e = (X^j_e)^{q-1}M'^j_{e,0}$$  \hspace{1cm} (4.36)$$

where, $M'^j_{e,0}$ is the initial mass of the element and $M'^j_e$ is the penalized element mass. However, according to the EoM of an MBS (Eq. 2.91), the relation between the lumped (or element) masses and element stiffness is not linear, thus, when high accelerations present in the system, penalizing the element or lumped masses is not the best solution for avoiding flying elements; on the other hand, the modification helps convergence of the algorithm.

Modification (4.35) can be seen as a filtering process of the element masses to obtain nodal or lumped masses. This formulation must be improved to maintain the overall mass of the body.

4.5. Convergence criterion

Due to the nonlinearity of the relation between the mass and the stiffness in a dynamic system the effect of flying elements is still slightly present despite implementation of (4.35). As a result using the same convergence criteria as in the static case, for example, the change of the value of the objective function from one iteration to the next or the average change in the design variable might be unsuccessful. To overcome this problem the unstable flying
elements must be excluded from the convergence criteria. The convergence criterion which is introduced in [21] is also appropriate to be used here: stop the iterations if the number of the design variables that change considerably from one iteration to the next does not exceed a small portion of the total number of design variables.

4.6. Some comments on the method

- The stopping time $t_s$ used in (4.1) is a function of the design. Since, the design changes at every iteration, the stop time of the simulation must be changed accordingly to include all important deformations that contribute to the topology optimization. The dependence of a proper stopping time to the design is more sensible when the mass is penalized.
- During the iterations neither the topology nor the dynamics of the system is physically correct since the stiffness and mass of the bodies change significantly; but, it converges to a correct result both for the topology and the dynamical behavior assuming that the thickness of the elements of the final result is either one or zero.
- For performing TOMBS some approximations in addition to the approximations in TO of the static structures are made, which are source of errors, each subject to more investigations. The important ones are modal reduction and sensitivity approximation and the approximations necessary in using Backward Differential formula for solving EoM. The TOMBS is sensitive to some of MBS simulation parameters such as number of considered modes, relative and absolute errors and stopping time.
- The choice of the Reference Condition is not an easy task when simulating the MBS behavior.
- MBS simulation is wrong during iterations since the stiffness of the elements with the thickness less than 1 is considerably reduced giving a deformation of the body that is actually more than the case when all elements have either minimum normalized thickness or one. If after convergence the body reaches the state where only maximum and minimum values of the thickness are present then the last iteration gives correct deformation and consequently correct dynamical simulation. But this is not the case at convergence and always some intermediate values of the thickness exist. Hence, for verification model of dynamical system it is needed to either round off the intermediate values to one or zero, or keep the intermediate values as they are (but this is not true for pure topology optimization especially in 3D, since the design variable is density), then redo the simulation such that the stiffness of no elements is reduced or penalized. This accompanies with making a small approximation in topology optimization since the body becomes more stiff (which actually is the goal of maximization the stiffness problem or minimization the compliance) and the forces at joints may change making optimal topology not completely the correct one for the applied forces. Note that this problem also happens for static topology optimization but in less magnitude since forces are constants in most of the cases. So, in conclusion, if the topology converges to a pure hole and material state then the dynamical behavior also converges to the correct behavior.
- The simulation must be performed such that it includes all major deflections of bodies during the operation and different working conditions.
5. Examples of Topology Optimization of Multibody Systems

A code has been written both in Matlab and Python to do Topology Optimization of the bodies embedded to an arbitrary two dimensional MBS which can be described by the features explained in Section 2.3. First, the MBS must be defined starting from defining bodies’ physical and mechanical properties, joints, constraints, forces and reference conditions. All bodies have initially a rectangular shape (the design area) which is meshed with Four-node rectangular elements. The joints between bodies are a revolute joint. The prismatic joints and a revolute joint between a body and the ground can also be defined using trajectory constraints. The code has an option to choose whether a body is rigid or flexible or if the simulation must be done with modal reduction or not. TO can be done only on the flexible bodies. One or several bodies can be chosen for performing TO in a time. The simulation of the behavior of the MBS with rectangular bodies also can be done without performing any TO. In the figure below the general procedure of performing TOMBS is repeated.

![Diagram of the general procedure of performing TOMBS](image)

**Fig. 5.1:** Illustration of the general procedure of performing TOMBS.

For all bodies in next examples the gravitational force is neglected and the body coordinate system is defines as shown in the figure below.
5.1. Slider Crank

5.1.1. Testing TOMBS on a Slider Crank system

As the first example a simple Slider Crank system shown in Fig. 5.3 is used for testing the idea of TOMBS. The result is shown in Fig. TO is performed for both bodies at a time. The input data needed for defining the problem is given in Table 1 in Appendix.

Fig. 5.2: The body system of coordinates.

Fig. 5.3: A schematic of a simple slider crank system.

Fig. 5.4: Results of the TOMBS; TO is performed simultaneously for both bodies. On the top: optimized MBS, Number of iterations is 100 and the stopping time is 0.712 s. Number of elements is 22500. The color bar shows the normalized thickness.
Some of the properties of TOMBS are investigated here on this system. The TO result depends on the stopping time of the simulation. The reason is that a longer simulation time includes more details and deflections in the behavior of the MBS which might be missed before. If the MBS behavior is periodic the best is to set the simulation time to be the same as the period time. The result of TOMBS for two different stopping time, $t_s$, is shown in Fig. 5.5. TO is performed for the Body 2.

$$t_s = 0.078 \text{s} \quad \text{and} \quad t_s = 0.624 \text{s}$$

![Fig. 5.5: Results of the TOMBS; the connecting rod (body 2) for two different stopping time, $t_s$; number of iterations is 180 and the number of elements is 4500. The compliance is given by $\sum_{x=1}^{n} \sum_{f=0}^{q} q^{ij}_{f,x} (x^{i}) k^{j,0}_{f,x} q^{ij}_{f,x} (x^{j}) q^{q}$, which can be averaged in time.](image)

Depending on the input data such as the BDF order, the absolute and relative tolerances and the number of eigenmodes the computer computation time may vary. It should be noted that if eigenvectors corresponding to the high eigenvalues are included the problem becomes stiffer and thus computation time is longer.

Also it should be noted that the final result might be different if all bodies in the MBS are considered to be flexibel. The reason is that the deflection in one body affects the whole system and thus its Topology Optimization.

The modification of the SIMP method suggested in Chapter 4 made it possible to get convergence of the optimization algorithm; otherwise the effect of flying element occurs which results in the non-convergence of the algorithm. This effect is shown in Fig. 5.6.
Fig. 5.6: Flying Element effect; (left) High displacement (norm) of a very elastic element when the mass is not penalized; (right) non-convergence of the body 2 after 100 iteration; the traditional SIMP is used with the lower thickness $10^{-3}$; see section 4.4.

In Fig. 5.7 the compliance history for four different cases are compared. The first case is when the traditional SIMP is used with no mass penalization; in the second case the SIMP is modified as in [15] with no mass penalization:

$$E_e^I = \begin{cases} (X_e^I)^q E^I & \text{if } 0.1 \leq X_e^I \leq 1 \\ E^I / 100 & \text{if } X_e^I < 0.1 \end{cases}$$ (5.1)

where, $E^I$ is the initial Young’s modulus shown in (2.10), $E_e^I$ is the penalized Young’s modulus. The third case is when the element masses are penalized as in (4.36) and finally, in the fourth case the lumped masses at nodal points are penalized according to (4.35).

In the Case 1 the effect of the flying elements totally destroys the convergence pattern; in the Case 2, this effect is a little attenuated which is enough for the most of the elements to converge to a stable state during optimization, but the overall optimum topology is questionable since there is a high fluctuations in compliance history. A question that may rise is why the algorithm converges when there are still high fluctuations in objective function. The answer is that the algorithm minimizes the objective function for a single element separately at an iteration. Thus those elements that are stable might converge as far as the effect of flying elements on them is not very considerable. In the Case 3; the effect of the flying elements still presents, but in less magnitude than in the Case 2; and finally, the Case 4 shows the best descend in compliance history and the effect of the flying elements is very small. Again, it should be noted that according to the EoM the relation between the stiffness and mass is not linear, hence direct penalizing the element or lumped masses is not the best solution, however, it is easy to implement. Also note that the sensitivity approximation is rough that might be a reason for not having a smooth minimization of the objective function. It can be also argued from the compliance history shown in Fig. 5.5 that if the system does not experience the high acceleration or if the high vibration frequencies are not excited the compliance history is smooth, which, is the case when the stopping time is 0.078s. In Fig. 5.8 the volume history for all above cases is illustrated.
Case 1

Case 2

Case 3

Case 4

Fig. 5.7: The compliance history for four different cases; the Case 1 is when the traditional SIMP is used with no mass penalization; in Case 2 the SIMP is modified as in (5.1) with no mass penalization; Case 3 is when the element masses are penalized as in (4.36); in Case 4 the lumped masses at nodal points are penalized according to (4.35). Number of iterations is 135 and the Number of elements is 4500. The compliance is given by

\[ \sum_{\varepsilon = 1} \sum_{\ell = 0} q_{\varepsilon,\ell}^{i,j} (x_{\varepsilon}^i)^{k_{\varepsilon,\ell}^{i,j}} (x_{\varepsilon}^{i,j})^{q_{\varepsilon,\ell}^{i,j}} \]

which can be averaged in time.

Fig. 5.8: The volume history for all cases in Fig. 5.7.
In Fig. 5.9 the norm of the maximum nodal deflection of the body 2 is compared between two different designs. The first is the initial design where the mass is uniformly distributed in the design space, and the second is the optimized topology with the same mass as in the initial design. Obviously it is expected that the optimized design has a less deflection.

![Graph](image)

Fig.5.9: Norm of the maximum elastic deformation of the connecting rod for two different designs, non-optimized and optimized. The weight of both systems is kept the same;

5.1.2. Topology optimization of the connecting rod using important static load cases

In Chapter 3, an example of TO on a multibody system was shown; where TO was performed on a dynamic body based on the multiple static load cases collected from the rigid body simulation of the MBS in Dymola. Some major approximations were necessary for using the method. In this section it is tried to compare the result of TO performed based on multiple static load cases with TOMBS introduced in Chapter 4. In the same way the loads acting on the body 2 (connecting rod) are collected from the rigid body simulation of the initial design. The initial design is a plate with the same desired volume of the final design.

\( \lambda^i \), the vector of Lagrange multipliers, gives the value of the reaction forces at joints of the body \( i \) in global system of coordinates. The left joint of the body \( i \) is the revolute joint which introduces two constraints in two dimensions one in \( x \) and one in \( y \) direction; on the right end of the body 2 there is a trajectory constraint similar to a slider joint which introduces a constraint always in global \( y \) direction. Hence there are three constraints imposed on the body 2 and thus three Lagrange multipliers, where the first two Lagrange multipliers belong to the revolute joint, and the third one, to the slider joint. Note that Lagrange multipliers are functions of time. The other force acting on the body 2 is by the spring attached to the right end which is always in the global \( x \) direction. To perform Topology Optimization these forces must be transformed to the body coordinate system. It can be done using the inverse of the transformation matrix \( A \) at a time \( t_i \) given in (2.72) but in two dimensions as in (2.98):

\[
F'_{Left\, joint} = A^{-1} \begin{bmatrix} \lambda_1^{2t_i} \\ \lambda_2^{2t_i} \end{bmatrix}
\]

(5.2)
Where, \( F_{\text{Left joint}} \) is the force vector acting on the left joint of the body 2 given in the body coordinate system, \( \lambda_1^{2,t_l} \) is the first Lagrange multiplier of the vector \( \lambda^{2,t_l} \), the vector of Lagrange multipliers associated to the body 2 at the time \( t_l \) and \( F_{\text{spr}}^{2,t_l} \) is the spring force acting on the right end of the body 2 at time \( t_l \).

Fig. 5.10 shows the forces acting on the body 2 given in the body coordinate system and the chosen loads for performing TO.

![Fig. 5.10: The forces acting on the body 2 given in the body coordinate system obtained by rigid body simulation; the chosen loads for performing TO are also shown.](image)

To build the load cases it is also needed to define the boundary conditions. There are no forces other than those at the joints. There are three options for defining boundary conditions that is shown in Fig. 5.11.

![Fig. 5.11: Three options for defining boundary conditions of the body 2 in static Topology Optimization.](image)
Simply supported boundary condition is used as the Reference Condition for defining the body coordinate system of the body 2 in MBS. Obviously this boundary condition uniquely is not enough to be used together with the loads shown in Fig. 5.10 to define load cases needed for TO, since it suppresses all loads except the load exerted on the right end in body x direction. Thus, other boundary conditions also need to be considered. Two more boundary conditions are shown in Fig. 5.11. Clumped ends introduce torque; hence both ends are chosen to be clumped in different boundary conditions to reduce the effect of the torque when doing TO. Therefore, five load cases are chosen for performing TO. Just Load 2 is active for simply supported boundary condition; these two together constitute one load case. The other two boundary conditions with two loads constitute four more load cases. The result of static Topology Optimization considering five load cases is shown in Fig. 5.12.

Fig. 5.12: The result of static Topology Optimization of the body 2 considering five load cases. 200 iterations are made.

Fig. 5.13: The norm of the maximum nodal deflection of the three designs during the operation; the objective function with no penalization for initial MBS is 4.28e-4, for TOMBS is 2.95e-4 and for Quasi-static TO is measured 3.52e-4; the weight is maintained the same for all designs.

One of the objectives of this thesis work is to prove that TOMBS gives a better design than the TO based on multiple static loads. The norm of the maximum nodal deflection of three designs during the operation is shown if Fig. 5.13. According to Fig. 5.13 the design obtained by TOMBS is the stiffest one which shows less deflection
during operation. In the initial design the mass is uniformly distributed in the design space. The total mass or volume is maintained the same for all three designs.

5.1.3. The effect of number of considered modes on MBSTO

The effect of the selected number of the first eigenmodes on the final optimized design of the body 2 is illustrated in Fig. 5.14. To remove the symmetry of the final result the stopping simulation time is chosen to be 0.078s; however the same was tested for longer stopping time and the conclusion is the same.

![Fig. 5.14: the effect of the chosen number of the first eigenmodes on the final optimized design of the body 2. The number of iterations is 150; the norm of the eigenmodes in 100; only body 2 is flexible in MBS; the rest of the data is according to the Table 5.1.](image)

It can be concluded form Fig. 5.14 that the optimized design does not change significantly if the number of the first eigenmodes to be ten or more. The number of unknowns in the reduced EoM is the sum of the number of the reference coordinates and the chosen eigenmodes for each body (in this example: 6+10, if only the body 2 is flexible); hence, less number of eigenmodes results in the faster simulation. Moreover, the highest selected eigenmode corresponds to the highest natural frequency of the system that is allowed. High frequencies make the problem stiff and thus the longer simulation time. In addition, if damping is also considered, high frequencies which contribute to the mechanical deformation damp quickly.

It can also be seen in Fig. 5.14 that the result of the TO, if only first eigenmode is chosen, is totally symmetric. The symmetry in x axis is because the MBS behaves symmetrically in x direction depending on the chosen stopping time; but, the symmetry in y axis is because only the first eigenmode is used. As the number of the eigenmodes increase more details of the true MBS behavior is accounted for and the symmetry in y axis disappears, reaching to the state where any increase in the level of details does not change the optimization result significantly. Note that the result of TOMBS when the coordinates are reduced may be different from the result of non-reduced
coordinates; however, as the number of the unknowns in an EoM with real coordinates is tens of thousands the simulation time is very long, forcing one to accept the modal reduction approximation. The number of needed eigenmodes also depends on the MBS operation frequencies and must be such that corresponding eigenvalues (MBS natural frequencies) be bigger than the operation frequencies.

5.2. **Nowait train**

Three screen shots of a Nowait train motion is shown in Fig. 5.15. The rails are diverging while the train is going up to the station causing reduction of the velocity from 10 m/s to about 1 m/s without braking. The objective is to find the optimal topology of the middle connecting rod which connects two wagons. The TOMBS data is given in Table 2. TOMBS is applied to the only first connecting rod and the result is replicated for the second one.

![Fig. 5.15: Three screen shots of a Nowait train motion; the velocity of the first wagon is shown.](image)

The MBS consists of 11 bodies; in Table 2 of the Appendix information of only 7 bodies are given. Bodies 1 and 3 are the wagons, body 2 is the connecting road and bodies 4-7 are the wheels. The only flexible body is the body 2. The result of TOMBS is shown in figures 5.16 and 5.17.

![Fig. 5.16: The results of the TOMBS of the Nowait connecting rod; Number of iterations is 255.](image)
5.3. A seven-body MBS

To show the power of the TOMBS method an MBS with seven bodies ([19], Schiehlen 1990) is tested. Three bodies are considered to be flexible. The schematic of a seven-body MBS called Squeezer is shown in Fig. 5.18.

The result of TOMBS is shown in Fig. 5.19. TOMBS data is given in Table 3 in Appendix. In Fig. 5.20 the result of two different design is presented where the Body 3 is optimized. In this figure one design corresponds to the system where all bodies other than the body 3 are considered to be rigid and the other design is obtained when 3 bodies in the system are flexible. It should be noted that the optimal topology is different because of the change in system overall response.
**Fig. 5.19:** The result of TOMBS on the Squeezer MBS. The stopping simulation time = $2.2\times10^{-3}$ s; (right) the compliance history of the body 2. Three bodies are optimized.

**Fig. 5.20:** The result of TOMBS on the Squeezer MBS. (right) three bodies are flexible in the system; (left) only the Body 3 is optimized and the other bodies are rigid;
6. The structure of the code

The structure of the code is similar to the steps shown in Fig. 5.1. First, the flexible MBS must be defined. Then based on the given information the finite element model of the flexible bodies is built and then the stiffness matrix, the mass matrices, inertial shape integrals, constraints, forces and generalized velocities are calculated. To keep this chapter short the focus is only on that part of the code which is needed for a user to define the TOMBS problem.

The structure of the Python code is explained here. There are three levels of calculations. Those calculations that do not need to be updated or recalculated neither in every time step of the simulation nor in optimization iterations are categorized in level one. The level two are those that need to be updated in every optimization iteration; finally, the calculations in level three are those that are repeated both in every time step and optimization iteration. Doing the calculations in the correct level helps to reduce the overall computation time. See Table 6.1 for short description on classes and functions.

<table>
<thead>
<tr>
<th>Classes</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlanarBody</td>
<td>Contains the default values of the body and override them by the new values coming from the function <code>Name_of_the_MBS</code>. (level one)</td>
<td></td>
</tr>
<tr>
<td>init_MBS</td>
<td>Initializes the MBS based on the information coming from the function <code>Name_of_the_MBS</code>. (level one)</td>
<td></td>
</tr>
<tr>
<td>MBS</td>
<td>A subclass of the Assimulo <code>Implicit_Problem</code> class which uses an instantiation of the class <code>init_MBS</code> as an input. This class contains two properties <code>MBS.res</code> and <code>MBS.MBS_solve</code>. (instantiation of the class is done in level one)</td>
<td></td>
</tr>
<tr>
<td>TOMBS</td>
<td>Initializes and solve the TOMBS problem. This class contains one property <code>TOMBS.solve</code>. (level one)</td>
<td></td>
</tr>
<tr>
<td>MBS.res</td>
<td>Returns the residuals of the EoM. (level three)</td>
<td></td>
</tr>
<tr>
<td>MBS.MBS_solve</td>
<td>Solves the EoM in Assimulo using IDA solver.(level three)</td>
<td></td>
</tr>
<tr>
<td>TOMBS.solve</td>
<td>Solves the TOMBS problem. (level one)</td>
<td></td>
</tr>
<tr>
<td>Name_of_the_MBS</td>
<td>Defines the MBS properties and TO options (the values that need to be set by the user). (level one)</td>
<td></td>
</tr>
<tr>
<td>PlatBody</td>
<td>Uses functions <code>mesh1</code> and <code>k_local</code> to mesh the body and generate the local specific stiffness matrices and the mesh information. (level one)</td>
<td></td>
</tr>
<tr>
<td>mesh1</td>
<td>Meshes the body with four-node rectangular elements. (level one)</td>
<td></td>
</tr>
<tr>
<td>k_local</td>
<td>Calculates the local specific stiffness matrix of an element. (level one)</td>
<td></td>
</tr>
<tr>
<td>FindNode</td>
<td>Finds the number of a node, and its DoFs, on the body that is the closest to the given local position. (level three)</td>
<td></td>
</tr>
<tr>
<td>Kiff</td>
<td>Calculates the global stiffness matrix of the body. (level two)</td>
<td></td>
</tr>
<tr>
<td>ModalVectors</td>
<td>Calculates the first n eigenvectors of the homogenous part of EoM. (level two)</td>
<td></td>
</tr>
<tr>
<td>Inertia_integrals</td>
<td>Calculates the inertia shape integrals. (level two)</td>
<td></td>
</tr>
<tr>
<td>Function Name</td>
<td>Description</td>
<td>Level</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Mass_Matrix2D</td>
<td>Calculates the mass matrix. (level three)</td>
<td></td>
</tr>
<tr>
<td>RevolutJoint2D</td>
<td>Calculates the residual of the Revolute joint constraint equations and the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>constraint Jacobian matrix. (level three)</td>
<td></td>
</tr>
<tr>
<td>Nodal_ex_Generalized_F</td>
<td>Gives a column vector containing the generalized external forces acting on</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a node of the mesh of the body. (level three)</td>
<td></td>
</tr>
<tr>
<td>Nodal_ex_Generalized_F_SDA_Fixed</td>
<td>Gives a column vector containing the Generalized external spring-damper-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>actuator force acting between a node of the mesh of the body and a fixed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>point in the space. (level three)</td>
<td></td>
</tr>
<tr>
<td>TrajectoryConstraint</td>
<td>Calculates the residual of the trajectory constraint equations and the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>constraint Jacobian matrix. (level three)</td>
<td></td>
</tr>
<tr>
<td>QuadraticVelocity2D</td>
<td>Calculates the quadratic velocity vector of the body. (level three)</td>
<td></td>
</tr>
<tr>
<td>Lam_Acc_CIC</td>
<td>Finds Consistent Initial Condition for Lagrange multipliers and nodal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>accelerations given initial positions and velocities. (level two).</td>
<td></td>
</tr>
<tr>
<td>Visualize</td>
<td>Animates the result of the simulation.(level one)</td>
<td></td>
</tr>
<tr>
<td>q_extract</td>
<td>Extracts the position (and also the velocity and the acceleration if needed)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>from the solution vector; retains the real coordinates if modal reduction</td>
<td></td>
</tr>
<tr>
<td></td>
<td>is used. (level three)</td>
<td></td>
</tr>
<tr>
<td>Fn</td>
<td>Defines the external force acting on the body in global coordinate system.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(level three)</td>
<td></td>
</tr>
<tr>
<td>Xstar</td>
<td>Specifies whether the design variable reached the box limit. (level three)</td>
<td></td>
</tr>
</tbody>
</table>

**Table 6.1**: Description about the classes and functions used in Python code.

### 6.1. Defining the TOMBS problem

The TOMBS problem can be defined by setting bodies’ physical and mechanical properties, joints, constraints, forces, reference conditions, finite element model data, solver options, EoM data and TO data. The needed data is shown, for instance, in Table 2 in Appendix. All bodies have initially a rectangular shape. Some of the properties have default values; by setting the value of the property the default value is overridden. Defining the TOMBS is done in class ‘Name_of_the_MBS’. The code has an option to choose whether a body is rigid or flexible or if the simulation must be done with modal reduction or not. TO can be done only on the flexible bodies, otherwise, an exception will raise. One or several bodies can be chosen for performing TO in a time.

#### 6.1.1. Defining a body

Each body of the MBS must be defined by setting the body properties. In this thesis work only a rectangular planar body is allowed. Defining a body lies in the level one of the calculations. The body properties or options are as follows:

- **Mechanical properties**: modulus of elasticity, Poisson ratio, initial thickness, density, x size (length), y size (width).
- **Finite Element Model options** (is used if the body is flexible): element type, number of elements in x direction, number of elements in y direction.
- TO options (is used if the body is flexible and if TOMBS is needed to be performed): maximum normalized thickness, minimum normalized thickness, $\alpha$ in OC method, $q$ factor in SIMP, filter radius, fraction of the initial volume.

These options are defined in class ‘\texttt{PlanarBody}’ and will be overridden if a new value is defined in class ‘\texttt{Name_of_the_MBS}’. For instance:

```python
# Some properties of Body 3. Other properties have default values. See class ‘PlanarBody’
B3_options = ['x size', 20e-3,
              'y size', 35e-3,
              'Initial thickness', 5e-3,
              'Min Thickness', 1e-5,
              'Modulus of Elasticity', 2e10,
              'Body_is_Deformable', 1,
              'Num x elements', 80,
              'Num y elements', 140]

Bodies_Options = [B1_options, B2_options, B3_options, B4_options, B5_options, B6_options, B7_options]

#All information about the bodies are gathered in the list ‘Bodies_Options’
```

### 6.1.2. Defining the joints, constraints and external forces

After defining the bodies they need to be assembled in the MBS. It is done by defining the joints and constraints between bodies and the ground. A revolute joint is between two bodies, thus the number of the bodies and the local position of the joint on each body must be specified. A trajectory constraint needs information about the residual of the trajectory function in the space, its partial derivatives with respect to $x$ and $y$, the number of body and the local position of the body where the constraint is applied. A revolute joint between a body and ground is defined by constraining the joint point of the body both in $x$ and $y$ directions, so the only DoF is the reference orientation. It should be noted that the code automatically applies the constraint on the nearest nodal point of the mesh to the given position. For instance:

```python
Joints = ['Revolute2D', [[[1, 7e-3, 0], [2, 28e-3, 0]],
                       [[2, 0, 0], [6, 20e-3, 0]]]

#First input in each inner list is the number of the body the second and the third are the local position of the rev joint on the body. Here two revolute joints are defined; one between the bodies 1 and 2, and one between 2 and 6.

Constraints = ['Trajectory', [[1, 0, 0, 'x', 't', '0'],
                             [1, 0, 0, 'y', '0', '1']]]

#Each list defines the number of the body and the undeformed position with respect to the body floating frame of reference, where the constraint acts. The third input is a string that is a function (A string variable for each constraint of the node) in terms of $x$, $y$, $t$ that describes the constraint (residual) which must be evaluated. For example ‘$x + y - 1$’. The other 2 strings are the partial derivative of the constraint function with respect to $x$ and $y$ respectively.

In the same way the needed information for defining external forces must be specified. For instance:
6.1.3. Defining the solution method

The information such as whether the body is flexible, modal reduction must be done or not, the reference condition of each flexible body, the stopping time and the index of the EoM are specified in this part. For instance:

```plaintext
Forces = ['External Force', [F1, 1.7e-3/2, 0], 'SDA Fixed_Body', [[4530, 0.07785, 0, 0], [3, 20e-3, -0.00124], [0.014, 0.072]]]
```

#'External Force' is a externally applied force on a point on the body. The first input is the name of the force function, the second is the body number, the third and fourth are the local position where the forces acts. ‘SDA Fixed_Body’ is a Spring Damper Actuator element between a fixed position in the space and the a point on the body. The first list gives the SDA data; the spring constant, undeformed length, damping constant and the magnitude of the actuator force respectively; the second list contains the number of the body, local x coordinates and y coordinates where on end of the SDA is attached to the body; the third list specifies the global position in the space where the other end of the SDA is attached.

6.1.4. Defining a consistent initial condition

In order to reduce the risk for the solver to fail at initial step a consistent initial condition must be provided by the user. It can be done by specifying the reference coordinates of each body, the global position of the origin of the

```
```

Type ‘yes’ in the list after ‘Modal reduction’ to solve the problem with modal reduction; the next list specifies the number of first modes for each body that are used for modal reduction; for rigid bodies these numbers are skipped.

The reference conditions are defined for flexible bodies. If the body is rigid the information about the reference condition is skipped. By applying the reference condition some DoFs of the elastic coordinates with respect to the body floating frame of reference are fixed. ‘Pinned’ means that the both DoFs of the nearest node to the given position in x and y directions are fixed. ‘X Slider’ means that the nearest node to the given local position is free to slide only in local x direction (corresponding elastic coordinate in y direction is fixed); in contrast, ‘Y Slider’ means that the nearest node to the given local position is free to slide only in local y direction (corresponding elastic coordinate in x direction is fixed). Also a ‘Clumped’ reference condition can be applied by pinning two or several nodes at the boundary. Obviously, several reference conditions can be defined for a body.

6.1.4. Defining a consistent initial condition

In order to reduce the risk for the solver to fail at initial step a consistent initial condition must be provided by the user. It can be done by specifying the reference coordinates of each body, the global position of the origin of the
body system of coordinates and the rigid body orientation, at the initial time. If the initial velocities are not zero, also a consistent value of the initial velocities must be provided. For instance:

\[
\begin{align*}
q_{01} &= \text{zeros}(3,1) \quad \# \text{reference position and angle of body 1} \\
q_{02} &= \text{zeros}(3,1) \quad \# \text{reference position and angle of body 2} \\
q_{01}[0] &= 0 \quad \# \text{x-position [m]} \\
q_{01}[1] &= 0 \quad \# \text{y-position [m]} \\
q_{01}[2] &= -0.062 \quad \# \text{angle [rad]} \\
q_{02}[0] &= 0.021 \times \cos(-0.062+\pi) \quad \# \text{x-position [m]} \\
q_{02}[1] &= 0.021 \times \sin(-0.062+\pi) \quad \# \text{y-position [m]} \\
q_{02}[2] &= -0.062 \quad \# \text{angle [rad]}
\end{align*}
\]

Consistence\_IC = [q_{01}, q_{02}] \quad \# \text{IC for reference coordinates}

ConsistenceV\_IC = [qv_{01}, qv_{02}] \quad \# \text{IC for reference velocities}

### 6.2. Reducing the computation time

It is tried to reduce the computation time of the TOMBS algorithm in different ways. For example, using the sparse matrices; reducing the number of the communication points needed for retaining the real coordinates; distributing the lumped masses in nodal points such that the shape functions convert to unity when calculating the inertia shape integrals; using the best function that calculate the eigenmodes of large matrices in a fast way; categorizing the calculations in three levels to avoid re-computation as much as possible; applying the constraints and forces only at nodal positions in order to avoid extra interpolations; the correct choice of the number of the elements, stopping time and number of the first eigenmodes; finally, finding and using an analytical expression for the integration of the elements stiffness matrices such that no approximations or more numerical calculation is needed.

The most time consuming parts of the computations are calculating the inertia shape integrals, solving the eigenvalue problem, evaluating the residual function and retaining the real physical coordinates; however, evaluating the residual function belongs to the level three of the calculations which must be performed many times in only one time step while the other three belong to the level two which must be done only in every optimization iteration. If the modal reduction is used, a reduced form of the residual function must be evaluated, thus the residual function evaluation and the 'IDA' solver simulation time does not actually depend on the number of the real physical elastic DoFs or the number of the elements; on the other hand, the needed time for calculating the inertia shape integrals, solving the eigenvalue problem and retaining the real physical coordinates increase considerably if the finite element mesh is refined.

The code is not printed in the report but it is available upon request.
7. Conclusion and future works

There are two approaches in literature for topology optimization of the bodies under dynamic loading. The first is called component-based approach where multiple static load cases are selected from the transient loads acting on the isolated body. Using this approach it is possible to implement the traditional multi-criteria structural optimization methods based on the static response to suggest a conceptual topology. However the final topology is very questionable. The main reason is that the coupling between rigid and elastic coordinates is omitted and the effect of the inertia forces is neglected, thus the static response cannot mimic the true displacement field and stresses. The second approach is based on Equivalent Static Loads Method (ESLM) where a static load vector at every time step is found such that it produces the same displacement field as the one in dynamic loading. Then again it would be possible to use the traditional multi-criteria structural optimization methods based on the static response to suggest a conceptual topology. However, this method is only well suited for size and shape optimization. Using the method for topology optimization causes instability and non-convergence of the optimization algorithm. In literature this problem is attenuated by removing some of the elements and updating the grid data in every optimization process. However, in this approach the design area is restricted and the possibility of elements’ revival in later iterations is neglected. Moreover, the element removal needs post processing of the data which is not unique for different problems. In addition constraints and the objective function cannot be defined based on overall system response. To overcome all these problems another systematic approach, the TOMBS method, is suggested in this thesis work.

It is shown that the TOMBS gives a design of a body embedded in MBS better than the design obtained based on the multiple static load cases, even though some approximations are made for calculating the sensitivity and reduction of coordinates. Moreover, several bodies can be optimized simultaneously; whereas, in the other methods each body must be optimized uniquely separated from the other bodies of the MBS while the final design still is not the best. The effect of the flying element which is the origin of instability and non-convergence of the topology optimization is considerably reduced by penalization of element masses or lumped masses. This approach is general and mathematically rigorous for topology optimization of all kind of MBS. However, the proposed mass penalization is not the prefect one since the relation between stiffness and mass is not linear according to the equation of motion. The TOMBS approach can find many applications in designing vehicle systems, high-speed robotic manipulators, airplanes and space structures.

A general static topology optimization is sensitive to some parameters such as filters used, number of design variables and SIMP factor; in addition, TOMBS is sensitive to the number of the eigenmodes and stopping time and also some parameters that influence the DAE solver performance such as the norm of the eigenmodes, absolute and relative tolerances. As it was discussed in Chapter 5 the final optimized design does not change significantly if the number of the first eigenmodes used for coordinate reduction is bigger than some threshold number. In the tested example the threshold number was ten, but it can be chosen bigger. In many applications 14 first eigenmodes can be enough. The choice of the stopping time depends on the behavior of the MBS. The stopping time must be chosen such that it includes all major deflections of bodies during the operation. If the MBS behaves periodically, at least one period can be enough; if it does not, the simulation time must be long enough or be one or several time intervals when the major deflections of MBS happens; thus for choosing a proper stopping time the MBS of the initial design must be at least one time simulated before starting setting up TOMBS. The dependence of the result to DAE solver performance can be investigated by using another DAE solvers in an MBS commercial simulators.
One disadvantage of TOMBS might be the longer computation time; however, computation time could be much longer if sensitivity approximation and modal reduction are not done. Also, using another DAE solver in an MBS commercial simulators which build and solve the EoM in a faster way can reduce the overall computation time of TOMBS. Another disadvantage is that solver may not give a solution if the problem is stiff. This problem also presents when using ESLM.

The possible future works can be listed as follows:

- A major attempt done in this thesis work was writing a code for building and solving an arbitrary two dimensional flexible multibody system; then the result was used in the loop of TOMBS and it was updated in every iteration. The general procedure of performing TOMBS is shown again in Fig. 8.1. The green steps can be replaced by an existing flexible MBS simulator such as Dymola flexible body library or any other. Replacing the green steps by more sophisticated simulators helps to reduce the simulation time and also makes it possible to do the TOMBS for a three dimensional MBS since the blue steps are general for two or three dimensions.

- Flexible Multibody Dynamics Library in Dymola builds the first input files, [17] and [18], and gives it to the Finite Element simulator ABAQUS and then takes back a Standard Input Data (SID) file [16]. SID is used for simulating the MBS behavior. In SID, only two sets of the coordinates which are called connection and shape nodes are retained; but for performing TOMBS all elastic coordinates must be retained from the reduced ones. Thus, performing TOMBS requires changing the content of the SID files such that it would be possible to retain all elastic coordinates.
- Investigating the error introduced to TOMBS by approximating the sensitivity and reducing the coordinates, to find ways of reducing the error or finding the methods of sensitivity analysis that have the least error; also to investigate if it is possible to compensate the error caused by modal reduction, for example by adding static modes correction.

- TOMBS can be extended to have multi-objective functions or different objective functions for different bodies. For instance, if several time interval of MBS behavior must be considered; or, if some bodies of the MBS need to be stiff while the others must be highly flexible; or also, if for some of the bodies the constraint is the weight and for some others the constraint is the displacement. The sensitivity approximation needs to be investigated for different objective functions.

- Try to investigate the possibility of using non-gradient based optimization which may affect the computation time significantly.

- TOMBS, where the MBS experiences impacts and discontinuities. The examples of such system are contact problems found in mechanical watches and construction tools.

- Adding damping to the system in order to reduce the effect of high frequencies.

- Investigate the best relation between element stiffness penalization and the lumped mass penalization to get a smooth minimization of the objective function. Such a relation is nonlinear according to the EoM.
References


[18] “Generating Flexible Body Standard input Data from an Abaqus Substructure”, Abaqus 6.11 documentation


# Appendix

## Table 1
TOMBS data of the Slider Crank system.

<table>
<thead>
<tr>
<th>Bodies</th>
<th>Body 1 (Crank Shaft)</th>
<th>Body 2 (connecting rod)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body is flexible?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Modulus of elasticity, $E$ [GPa]</td>
<td>$4 \times 10^9$</td>
<td>$4 \times 10^9$</td>
</tr>
<tr>
<td>Poisson ratio, $\nu$</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>Initial thickness [m]</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Density [$kg/m^3$]</td>
<td>7800</td>
<td>7800</td>
</tr>
<tr>
<td>$x$ size [m]</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td>$y$ size [m]</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Initial position of the origin of the Body coordinate system with respect to the global coordinate system</td>
<td>(0, 0)</td>
<td>(−0.0467, 0.0884)</td>
</tr>
<tr>
<td>Initial rigid body orientation [rad]</td>
<td>2.0563</td>
<td>5.9839</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revolute Joints</th>
<th>Revolute Joint 1</th>
<th>Revolute Joint 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between bodies</td>
<td>1, 2</td>
<td>-</td>
</tr>
<tr>
<td>Local position on the first body</td>
<td>(0,1,0)</td>
<td>-</td>
</tr>
<tr>
<td>Local position on the second body</td>
<td>(0,0)</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trajectory constraints</th>
<th>Trajectory Constraint 1</th>
<th>Trajectory Constraint 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of the body</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Local position on the body</td>
<td>(0,0)</td>
<td>(0,3,0)</td>
</tr>
<tr>
<td>Residual functions that define the trajectory</td>
<td>$x = 0, y = 0$ ($x, y$ are the global positions)</td>
<td>$y = 0$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Forces</th>
<th>Extremly applied force</th>
<th>Spring-Damper-Actuator element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of the body</td>
<td>1</td>
<td>Between 2 and ground</td>
</tr>
<tr>
<td>Local position on the body</td>
<td>(0.05,0)</td>
<td>(0.3,0), (0.4,0)</td>
</tr>
<tr>
<td>Force data</td>
<td>$-1000 \ N$ always in $x$ direction</td>
<td>$K = 1000 \frac{N}{m}, I_0 = 0.2 \ m, c = 0,$ $F_{na} = 0$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference Conditions (only for flexible bodies)</th>
<th>Body 1</th>
<th>Body 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body 1</td>
<td>Local position (0,0) pinned ($x, y$ DoFs are fixed, in other words corresponding elastic coordinates in $x$ and $y$ directions are fixed)</td>
<td>Local position (0.1,0) free to slide only in local $x$ direction (corresponding elastic coordinates in $y$ direction is fixed)</td>
</tr>
<tr>
<td>Body 2</td>
<td>Local position (0,0) pinned</td>
<td>Local position (0.3,0) free to slide only in local $x$ direction (corresponding elastic coordinates in $y$ direction is fixed)</td>
</tr>
<tr>
<td></td>
<td>Body 1</td>
<td>Body 2</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td><strong>Finite Element Model data (only for flexible bodies)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Element type</td>
<td>Four-node rectangular element</td>
<td>Four-node rectangular element</td>
</tr>
<tr>
<td>Number of elements in x direction</td>
<td>150</td>
<td>100</td>
</tr>
<tr>
<td>Number of elements in y direction</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td><strong>Modal (Coordinate) reduction data (only for flexible bodies)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of the first modes used</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td><strong>IDA Solver and EoM data</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RelTol</td>
<td>1.e-6</td>
<td></td>
</tr>
<tr>
<td>AbsTol</td>
<td>1.e-6</td>
<td></td>
</tr>
<tr>
<td>suppressAlgVars</td>
<td>On</td>
<td></td>
</tr>
<tr>
<td>MaxNumSteps</td>
<td>4000</td>
<td></td>
</tr>
<tr>
<td>Stopping time</td>
<td>0.624 s (body 1), 0.468 s (body 2)</td>
<td></td>
</tr>
<tr>
<td>Number of the communication points</td>
<td>312 (body 1), 234s (body 2)</td>
<td></td>
</tr>
<tr>
<td>EoM Index</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>TO data (only for flexible bodies)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum thickness</td>
<td>1e-3</td>
<td>1e-3</td>
</tr>
<tr>
<td>Maximum thickness</td>
<td>1 m</td>
<td>1 m</td>
</tr>
<tr>
<td>$\alpha$ in OC method</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$q$ factor in SIMP</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Filter radius</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>Fraction of the initial volume</td>
<td>0.4</td>
<td>0.45</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>180</td>
<td>180</td>
</tr>
</tbody>
</table>
Table 2

TOMBS data of the Nowait train.

<table>
<thead>
<tr>
<th>MBS data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bodies</strong></td>
</tr>
<tr>
<td>Body 1</td>
</tr>
<tr>
<td>Body is flexible?</td>
</tr>
<tr>
<td>Modulus of elasticity, $E$ [GPa]</td>
</tr>
<tr>
<td>Poisson ratio, $v$</td>
</tr>
<tr>
<td>Initial thickness [m]</td>
</tr>
<tr>
<td>Density [$kg/m^3$]</td>
</tr>
<tr>
<td>$x$ size [m]</td>
</tr>
<tr>
<td>$y$ size [m]</td>
</tr>
<tr>
<td>Initial position of the origin of the Body coordinate system with respect to the global coordinate system</td>
</tr>
<tr>
<td>Initial rigid body orientation [rad]</td>
</tr>
<tr>
<td><strong>Revolute Joints</strong></td>
</tr>
<tr>
<td>Rev 1</td>
</tr>
<tr>
<td>Between bodies</td>
</tr>
<tr>
<td>Local position on the first body</td>
</tr>
<tr>
<td>Local position on the second body</td>
</tr>
<tr>
<td><strong>Trajectory constraints</strong></td>
</tr>
<tr>
<td>TC 1</td>
</tr>
<tr>
<td>Number of the body</td>
</tr>
<tr>
<td>Local position on the body</td>
</tr>
<tr>
<td>Residual functions that define the trajectory</td>
</tr>
<tr>
<td><strong>Forces</strong></td>
</tr>
<tr>
<td>External force 1</td>
</tr>
<tr>
<td>Number of the body</td>
</tr>
<tr>
<td>Local position on the body</td>
</tr>
<tr>
<td>Force data</td>
</tr>
</tbody>
</table>

**Reference Conditions (only for flexible bodies)**

Body 2
- Local position (0, 0) pinned ($x$, $y$ DoFs are fixed, in other words corresponding elastic coordinates in $x$ and $y$ directions are fixed)
- Local position (0, 1) free to slide only in local $x$ direction (corresponding elastic coordinates in $y$ direction is fixed)

**Finite Element Model data (only for flexible bodies)**

<table>
<thead>
<tr>
<th>Body 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element type</td>
</tr>
<tr>
<td>Number of elements in $x$ direction</td>
</tr>
<tr>
<td>Number of elements in $y$ direction</td>
</tr>
</tbody>
</table>

**Modal (Coordinate) reduction data (only for flexible bodies)**

99
<table>
<thead>
<tr>
<th>Number of the first modes used</th>
<th>14</th>
</tr>
</thead>
</table>

**IDA Solver and EoM data**

<table>
<thead>
<tr>
<th>RelTol</th>
<th>$1. e^{-6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbsTol</td>
<td>$1. e^{-6}$</td>
</tr>
<tr>
<td>suppressAlgVars</td>
<td><em>on</em></td>
</tr>
<tr>
<td>MaxNumSteps</td>
<td>20000</td>
</tr>
<tr>
<td>Stopping time</td>
<td>15 s</td>
</tr>
<tr>
<td>Number of the communication points</td>
<td>200</td>
</tr>
<tr>
<td>EoM Index</td>
<td>2</td>
</tr>
</tbody>
</table>

**TO data (only for flexible bodies)**

<table>
<thead>
<tr>
<th>Body 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum thickness</td>
</tr>
<tr>
<td>Maximum thickness</td>
</tr>
<tr>
<td>$\alpha$ in OC method</td>
</tr>
<tr>
<td>$q$ factor in SIMP</td>
</tr>
<tr>
<td>Filter radius</td>
</tr>
<tr>
<td>Fraction of the initial volume</td>
</tr>
<tr>
<td>Number of iterations</td>
</tr>
</tbody>
</table>

Cons1 and Cons2 are the functions of upper and lower trajectories, respectively. External forces are active when the wagons are in diverging part of the road.
**Table 3**

TOMBS data of the Squeezer MBS.

<table>
<thead>
<tr>
<th>Bodies</th>
<th>Body 1</th>
<th>Body 2</th>
<th>Body 3</th>
<th>Body 4</th>
<th>Body 5</th>
<th>Body 6</th>
<th>Body 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body is flexible?</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Modulus of elasticity, $E$ [GPa]</td>
<td>–</td>
<td>–</td>
<td>$2e10$</td>
<td>–</td>
<td>$2e10$</td>
<td>–</td>
<td>$2e10$</td>
</tr>
<tr>
<td>Poisson ratio, $v$</td>
<td>–</td>
<td>–</td>
<td>0.3</td>
<td>–</td>
<td>0.3</td>
<td>–</td>
<td>0.3</td>
</tr>
<tr>
<td>Initial thickness [m]</td>
<td>$5e-3$</td>
<td>$5e-3$</td>
<td>$5e-3$</td>
<td>$5e-3$</td>
<td>$5e-3$</td>
<td>$5e-3$</td>
<td>$5e-3$</td>
</tr>
<tr>
<td>Density [$kg/m^3$]</td>
<td>7800</td>
<td>7800</td>
<td>7800</td>
<td>7800</td>
<td>7800</td>
<td>7800</td>
<td>7800</td>
</tr>
<tr>
<td>$x$ size [m]</td>
<td>$7e-3$</td>
<td>$28e-3$</td>
<td>$20e-3$</td>
<td>$3e-3$</td>
<td>$40e-3$</td>
<td>$20e-3$</td>
<td>$4.49e-3$</td>
</tr>
<tr>
<td>$y$ size [m]</td>
<td>$3e-3$</td>
<td>$3e-3$</td>
<td>$35e-3$</td>
<td>$20e-3$</td>
<td>$9.16e-3$</td>
<td>$3e-3$</td>
<td>$40e-3$</td>
</tr>
<tr>
<td>Initial position of the origin of the Body coordinate system with respect to the global coordinate system</td>
<td>(0,0)</td>
<td>(-0.02096, 0.0013)</td>
<td>(-0.0077, 0.01701)</td>
<td>(-0.02946, 0.0074)</td>
<td>(-0.07148, 0.00177)</td>
<td>(-0.03163, -0.01561)</td>
<td>(-0.05198-0.013177)</td>
</tr>
<tr>
<td>Initial rigid body orientation [rad]</td>
<td>$-0.06171$</td>
<td>$-0.06171$</td>
<td>0.45528</td>
<td>0.71</td>
<td>0.48736</td>
<td>-0.22266</td>
<td>1.23</td>
</tr>
</tbody>
</table>

**Revolute Joints**

<table>
<thead>
<tr>
<th>Between bodies</th>
<th>Rev 1</th>
<th>Rev 2</th>
<th>Rev 3</th>
<th>Rev 4</th>
<th>Rev 5</th>
<th>Rev 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revolute Joint</td>
<td>1,2</td>
<td>2,3</td>
<td>2,4</td>
<td>2,6</td>
<td>4,5</td>
<td>2,6</td>
</tr>
<tr>
<td>Local position on the first body</td>
<td>($7e-3$,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(1.5e-3,10e-3)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>Local position on the second body</td>
<td>($28e-3$,0)</td>
<td>(0,17.5e-3)</td>
<td>(1.5e-3,10e-3)</td>
<td>($20e-3$,0)</td>
<td>(4.84e-3,4.84e-3)</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>

**Trajectory constraints**

<table>
<thead>
<tr>
<th>TC</th>
<th>TC 1</th>
<th>TC 2</th>
<th>TC 3</th>
<th>TC 4</th>
<th>TC 5</th>
<th>TC 6</th>
<th>TC 7</th>
<th>TC 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of the body</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Local position on the body</td>
<td>(0,0)</td>
<td>(0,0)</td>
<td>(0,17.5e-3)</td>
<td>(0,17.5e-3)</td>
<td>(0,-4.85e-3)</td>
<td>(0,-4.85e-3)</td>
<td>4.49e-3,4.49e-3</td>
<td>(0,0)</td>
</tr>
<tr>
<td>Residual functions that define the trajectory</td>
<td>$x$</td>
<td>$y$</td>
<td>$x + 0.0363$</td>
<td>$y - 0.0327$</td>
<td>$x + 0.06934$</td>
<td>$y + 0.00227$</td>
<td>$x + 0.06934$</td>
<td>$y + 0.00227$</td>
</tr>
</tbody>
</table>

**Forces**

<table>
<thead>
<tr>
<th>Number of the body</th>
<th>External force 1</th>
<th>Spring-Damper-Actuator element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local position on the body</td>
<td>(3.5e-3,0)</td>
<td>Between 3 and ground</td>
</tr>
<tr>
<td>Force data</td>
<td>200 $N$ always in body $y'$ direction</td>
<td>$K = 4530 \frac{N}{m^3} l_o = 0.07785 m, c = 0$, $F_a = 0$</td>
</tr>
</tbody>
</table>

**Reference Conditions (only for flexible bodies)**

- **Body 3**: Local position $(0,17.5e-3)$ pinned ($x,y$ DoFs are fixed).
- **Body 5**: Local position $(0,-17.5e-3)$ free to slide only in local $y$ direction (corresponding elastic coordinates in local $x$ direction is fixed).
- **Body 5**: Local position $(40e-3, -4.58e-3)$ free to slide only in local $x$ direction (corresponding elastic coordinates in local $y$ direction is fixed).
<table>
<thead>
<tr>
<th>Body 7</th>
<th>Local position ((4.49e - 3, 3.20e - 3)) pinned ((x, y)) DoFs are fixed</th>
<th>Local position ((4.49e - 3, -20e - 3)) free to slide only in local (y) direction (corresponding elastic coordinates in local (x) direction is fixed)</th>
</tr>
</thead>
</table>

**Finite Element Model data (only for flexible bodies)**

<table>
<thead>
<tr>
<th>Element type</th>
<th>Body 3</th>
<th>Body 5</th>
<th>Body 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element type</td>
<td>Four-node rectangular element</td>
<td>Four-node rectangular element</td>
<td>Four-node rectangular element</td>
</tr>
<tr>
<td>Number of elements in (x) direction</td>
<td>80</td>
<td>210</td>
<td>40</td>
</tr>
<tr>
<td>Number of elements in (y) direction</td>
<td>140</td>
<td>50</td>
<td>280</td>
</tr>
</tbody>
</table>

**Modal (Coordinate) reduction data (only for flexible bodies)**

<table>
<thead>
<tr>
<th>Number of the first modes used</th>
<th>Body 3</th>
<th>Body 5</th>
<th>Body 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of the first modes used</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

**IDA Solver and EoM data**

| RelTol | \(1.0e - 6\) |
| AbsTol | \(1.0e - 6\) |
| suppressAlgVars | on |
| MaxNumSteps | 20000 |
| Stopping time | 0.0022 \(s\) |
| Number of the communication points | 200 |
| EoM Index | 2 |

**TO data (only for flexible bodies)**

<table>
<thead>
<tr>
<th>Body 2</th>
<th>Minimum thickness</th>
<th>(1e - 5 \text{ m})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum thickness</td>
<td>(5e - 3 \text{ m m})</td>
<td></td>
</tr>
<tr>
<td>(\alpha) in OC method</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(q) factor in SIMP</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Filter radius</td>
<td>3.5</td>
<td></td>
</tr>
<tr>
<td>Fraction of the initial volume</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>Number of iterations</td>
<td>198</td>
<td></td>
</tr>
</tbody>
</table>