A Graph-based Optimization Method for the Design of Compliant Mechanisms and Structures

Author(s):
Sauter, D.

Publication Date:
2008

Permanent Link:
https://doi.org/10.3929/ethz-a-005700688

Rights / License:
In Copyright - Non-Commercial Use Permitted
A Graph-based Optimization Method
for the Design of
Compliant Mechanisms and
Structures

A dissertation submitted to the
Swiss Federal Institute of Technology
Zurich

for the degree of
Dr. sc. ETH Zurich

presented by
Michael Sauter
Dipl. Masch. Ing. ETH
born June 21, 1977
citizen of Ermatingen (TG)

accepted on recommendation of
Prof. Dr. P. Ermanni, examiner
Prof. Dr. F. van Keulen, co-examiner

2008
Abstract

This thesis investigates several aspects of the design of compliant mechanisms and structures. It focuses the structural synthesis as well as the postprocessing of the data obtained by the optimization. Finally, the adaptive care seat concept demonstrates the capability of the synthesis procedure.

A method is introduced to solve compliant mechanism and structure problems that combines evolutionary optimization techniques with the ground structure approach whereas the use of graph theory and a complex-shaped, two-dimensional beam element can increase both numerical efficiency and solution space. The beam element is curved and of variable thickness so that a localized compliant region can be placed within one beam element. The ground structure topology is represented by a mathematical graph whose edges and vertices fall together with the beams and their nodes, respectively. All genetic operators, i.e., mutation and crossover, directly apply on the graph representation (vertices and edges). This implies a set of modified and new genetic operators, which are also developed within this work. The graph-based representation allows the optimization of topology, geometry, and sizing of the beam structures at the same time. Besides this, it can handle individuals of different sizes. As a further feature, the graph-based optimization is able to find solutions for highly-constrained and discrete optimization problems such as for multi-material optimization. The inverter and the gripper optimization sample problems serve to demonstrate the advantages and disadvantages of the method.

Compliant mechanisms and structures often undergo large displacement, in order to provide their functionality. Therefore, geometrical nonlinear analysis is required. The presented optimization provides both options; geometrical linear as well as nonlinear analysis. Sample large displacement grippers are illustrating the difference of the linear and nonlinear analysis.

The graph-based design environment helps setting up new optimizations, visualizing and analyzing the optimized results and allow a fast aftertreatment of the structures obtained by the optimization. The
aftertreatment or the so-called final design interpretation step is necessary, since the outcomes of the optimization are encoded according to the graph representation and not directly real-world models. The final design interpretation maps the encoded data into CAD (Computer Aided Design) models.

An adaptive car seat concept is presented as a new, possible application of large scale, shape morphing, compliant structure. The adaptive car seat consists of compliant rib-like structures, which are kept by two columns in position. The compliant rib-like structure are designed to adapt themselves to the driver by embracing the back, when he leans into the seat and applying an additional actuation force. The whole development procedure is presented starting from the design domain parametrization and the fitness formulation up to the postprocessing of the optimization result. Finally, the real-world prototype, which integrates a pneumatic actuation system, validates the structural behavior predicted by the graph-based optimization. The adaptive car seat concept serves as an example of simultaneously increasing the functionalities and decreasing the complexity of a construction (number of parts).
Zusammenfassung

Diese Arbeit beschäftigt sich mit verschiedenen Aspekten im Bereich nachgiebiger Mechanismen und Strukturen. Sie konzentriert sich sowohl auf die strukturelle Synthese, als auch auf die Nachbearbeitung der Daten aus der Optimierung. Schlussendlich zeigt das adaptive Autositz-Konzept die Möglichkeiten des Entwicklungprozesses auf.


Acknowledgements

The present thesis was accomplished at the Centre of Structure Technologies at ETH Zurich and was internal funded.

Several people assisted this work and it could not have been carried without their help. Especially I would like to thank:

- First of all, Christina, my wife, for her support and love throughout the years.
- Professor Paolo Ermanni for giving me the opportunity of working in this interesting field of research and his guidance during my doctoral thesis.
- Professor Fred van Keulen for being co-examiner of my thesis, for his persistence in reading the thesis and for his very helpful feedbacks.
- Dr. Gerald Kress and Dr. Mathias Giger for their valuable contributions to this work.
- All students contributing to this work, particularly Andri Bezziola for his great work concerning the adaptive car seat concept, Thomas Kern for his contribution in the field of the final design interpretation and Michael Winkler for his implementations related to the enhanced mapping concept.
- Hansruedi Nef and Hanspeter Eigenmann for their technical support and their enthusiasm concerning the hockey stick project.
- All the people at the Centre of Structure Technologies, particularly Alberto Belloli, René Roos, David Keller, Chiara Zaniboni, René Arbter and Simon Steiner for the inspiring discussions and for sharing a good time.
- Last but not least, my best room mate, Max Fickel for the very quiet hours in the morning and very funny discussions in the afternoon.
List of Symbols

\( \beta_1 \) Run-out angle 1, 29  
\( \beta_2 \) Run-out angle 2, 29  
\( \varepsilon \) Strain, 25  
\( \gamma \) Shear stress, 38  
\( \kappa \) Curve curvature, 33  
\( \nu \) Poisson’s rate, 25  
\( \varphi \) Rotation, 36  
\( \sigma \) Normalized coordinate, 29  
\( \chi \) Local curvature, 37  

\( b \) Width, 24  
\( k \) Shear correction factor, 37  
\( l \) Arc length of the beam, 32  
\( r \) Local radius parameter, 37  
\( s \) Arc length parameter, 32  
\( t \) Thickness, 24  
\( u \) Displacement in x direction, 36  
\( \Delta u \) Unite displacement, 25  
\( v \) Displacement in y direction, 36  
\( x \) Coordinate, 29  
\( y \) Coordinate, 29  
\( \hat{y} \) Local parameter, 37  

\( A \) Cross-sectional area, 24
$E$ Young’s modulus, 24
$F_x$ Force in x direction, 36
$F_y$ Force in y direction, 36
$G$ Shear modulus, 25
$I$ Cross-sectional moment of inertia, 24
$K$ Stiffness matrix, 42
$L$ Shape parameter, 29
$M$ Bending moment, 36
$N$ Normal force, 25, 37
$P_1$ Endpoint 1, 29
$P_2$ Endpoint 2, 29
$Q$ Transverse force, 37
$Q_B$ Transverse force, 25
$Q_S$ Shear force, 25
$R$ Local radius of the shape curvature, 37
$U$ Deformation energy, 36
$U_s$ Local deformation energy, 38
$W_B$ Stored energy due to bending, 25
$W_{in}$ Input energy, 27
$W_N$ Stored energy due to axial loading, 25
$W_{out}$ Output energy, 26
$W_S$ Stored energy due to shearing, 25
$W_{stored}$ Stored energy, 27
$Y$ Ratio of $R$ to $r$, 38
List of Acronyms

APDL  ANSYS Parametric Design Language, 178
ASCII American Standard Code for Information Interchange, 45
AWW  Nasa Active Aeroelastic Wing, 5
BGL  Boost Graph Library, 68
C++  Object-oriented programming language, 151
CAD  Computer Aided Design, 103
EA  Evolutionary Algorithm, 13
ECP  Element Connectivity Parameterization, 12
EO  Evolving Objects, 68
EP  Evolutionary Programming, 15
ES  Evolution Strategies, 15
FE  Finite Element, 68
FEM  Finite Element Model, 11
GA  Genetic Algorithm, 13
GP  Genetic Programming, 15
GUI  Graphical User Interface, 165
MEMS  Micro-Electro-Mechanical System, 113
SISO  Single-Input-Single-Output, 5
VB  Visual Basic, 105
# Contents

List of Symbols ix  
List of Acronyms xi  

## 1 Introduction 1  
1.1 Compliant mechanisms 3  
1.2 State-of-the-art 8  
1.3 Research needs and goals 20  
1.4 Thesis outline 21  

## 2 Complex-shaped beam element 23  
2.1 Complex-shaped beam element as a ground structure 24  
2.2 Parametrization of the complex-shaped beam element 29  
2.3 Influence of shape parameter $L$ 31  
2.4 Constant stretching concept 35  
2.5 Beam-theory approach 36  
2.6 Plane-elements approach 45  

## 3 Graph-based optimization 49  
3.1 The graph-based topology representation 51  
3.2 Graph representation based operators 57  
3.3 Fitness formulation 59  
3.4 Pre-evaluation routines 62  
3.5 Implementation 69  
3.6 Validation 71  
3.7 Nonlinear optimization 82  
3.8 Two-and-a-half-dimensional optimization 87
CONTENTS

3.9 Multi-material solutions ........................................ 90
3.10 Enhanced mapping concept ................................. 95

4 Postprocessing of the graph-based optimization 101
  4.1 Graph-based design environment .......................... 103
  4.2 Final design interpretation of a parallel gripper ....... 104

5 Application: Adaptive car seat concept 113
  5.1 Design objectives ........................................... 114
  5.2 Adaptive car seat concept .................................. 115
  5.3 Design domain parametrization ............................ 118
  5.4 Results ..................................................... 122
  5.5 Final design ............................................... 123
  5.6 Functional prototype ........................................ 125
  5.7 Discussion ................................................. 128

6 Conclusion and outlook 131
  6.1 Conclusion .................................................. 131
  6.2 Outlook ...................................................... 136

A Approximating equation for constant stretching concept 141
  A.1 Approximation of shape parameter λ ..................... 145

B EoGraph optimization 151
  B.1 Data sources for the geometry ............................. 151
  B.2 Data Source for the running parameters of an optimization 158

C EoGraph preprocessing tool 165
  C.1 Graphical layout ........................................... 167

D EoGraph postprocessing tool 175
  D.1 EoGraph postprocessing graphical user interface ....... 177
  D.2 EoGraph postprocessing-Catia interface .................. 179

List of Tables 183

List of Figures 185
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bibliography</td>
<td>191</td>
</tr>
<tr>
<td>Own publications</td>
<td>203</td>
</tr>
<tr>
<td>Curriculum Vitae</td>
<td>205</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Classical criteria for the design of lightweight structures are stiffness, stability and in some cases also strength. Generally, the structures are designed to not deflect more than a certain value under load using as less material as possible. Compliant mechanisms represent a newer design discipline of lightweight structures. The basic idea of compliant mechanisms is to design the compliancy of the structure, in order to integrate a certain functionality into the structural element. Often the target functionality is known in advance, but the structural layout, which fulfills the request, is unknown.

Figure 1.1: Da Vinci’s catapult and crossbow drawings [1]

This thesis deals with design and development of compliant mechanisms. Compliant devices were already known in the Middle Ages.
Leonardo da Vinci’s (1452-1519) drawings of a catapult and a long-bow serve as examples of compliant devices (Figure 1.1). The early catapults and long bows were constructed of wood that were deflected to store energy and release it to the projectile. Since the eighties of the last century compliant mechanisms and devices are again receiving growing attention in the scientific communities because of their potential for mass reduction and intrinsic multi-functionality. New methods were developed and studied for the design of compliant mechanisms. Each of them provides advantages, but also disadvantages. However, the design synthesis of compliant mechanism is still challenging.

This introductory chapter is organized as follows. First, in Section 1.1 an introduction to compliant mechanisms is given. Their advantages as well as their disadvantages are discussed. In the subsequent Section 1.2 the state-of-the-art design methods for compliant mechanisms and the basic modules of an Evolutionary optimization are presented. Section 1.3 points out the goals of this thesis emerging from the identified needs for further research. And finally, a detailed overview of the present thesis is given in Section 1.4.

Figure 1.2: Minimal invasive surgery gripper (source: EMPA)
1.1 Compliant mechanisms

Compliant mechanisms are single-piece devices that combine the features of both structures and mechanisms. While they are carrying loads like conventional structures, they are also designed to be intentionally flexible providing hingeless mechanisms [2]. Mechanisms transfer or transform motion, force or energy. Unlike rigid-link mechanisms, compliant mechanisms gain their mobility from the deflection of the flexible members. Examples of a rigid-link and a compliant gripper for the minimal invasive surgery are given in Figure 1.2.

Both invasive surgery grippers transfer the input force to the output. While the rigid-link device gains their functionality by a hinge mechanism, the compliant device undergoes elastic deformations.

Examples of Figure 1.3 shows that compliant devices are already everyday objects. The main advantages of the presented devices are the very inexpensive mass production and their reliability.

Figure 1.3: Common compliant devices; a binder clip, paper clips, backpack latch, lid eyelash curler, and nail clipper [3]
Introduction

Figure 1.4: Ice hockey stick

Figure 1.5: Long bow

Figure 1.6: Active Aeroelastic Wing (AWW) project [4]

Figure 1.7: Micro gripper

Figure 1.8: Macro gripper
1.1 Compliant mechanisms

Generally speaking today’s compliant devices show four specific functionalities:

1) Clamping or fixation mechanisms
2) Energy storage and release
3) Transferring or transforming mechanisms of motion, forces and energy
4) Shape adaptation

Examples are illustrating the last three functionalities. Hockey sticks and long bows serve as examples for energy storage and release devices (Figure 1.4 and 1.5). In both devices the strain energy is transformed to kinetic energy of the arrow or the puck, respectively. In case of the hockey stick, the player hits the ice in front of the puck, bends it like a bow and releases it. In doing so, he accelerates the puck.

Grippers as shown in Figures 1.7 and 1.8 are examples for devices that transfer or transform motion, forces and energy. Often, conventional hinge solutions can not be miniaturized. Micro compliant mechanisms are thereby a viable alternative. They require no assembly and less space, and finally, they are less complex. For instance, they can be fabricated in a planar lithography process.

Another fascinating and interesting functionality is shape adaption for shape control (Figure 1.6). Due its complexity and its structural interaction between the flexible members, the development is challenging. All aspects, such as structural design, material behavior, manufacturing technology and actuation concept, have to be considered concurrently. The Nasa Active Aeroelastic Wing (AWW) project is an example for a shape adaption application. Their flexible wing displayed in Figure 1.6 provides roll comparable to that achieved by a standard stiff-wing, while the experimental wing does it with less need for coordinated tail surface inputs [4].

In the following this thesis uses the terms definition as shown in Figure 1.9. It is distinguished between compliant mechanisms and compliant structures. The thesis uses compliant mechanisms if devices with a discrete number of inputs/outputs, e.g. single-input-single-output (SISO) devices, are considered. Compliant structures specify
distributed output devices such as for shape control. In combination with an actuation system it is termed *compliant system*.

So far, it was explained, what compliant mechanisms/structures are, and examples of them are given. But, what are the features such devices provide? Of course, advantages and disadvantages of the features strongly depend on the considered application.

Compliant mechanisms/structures offer the potential of the dramatic reduction in the total number of parts due to the integration of functionalities. The reduction in parts affects manufacturing and assembly time as well as costs. Some mechanisms may be constructed of a single piece and may be manufactured in an one-step process such as injection molding. And, since each hinge is a concentration of mass, compliant mechanisms open up the opportunity for mass reduction. In addition, they have not any friction or backlash and need no lubrication. This may result in a higher performance (increased precision, increased reliability, reduced wear, and reduced maintenance). Finally, compliant mechanisms/structures provide an additional feature if we regard aerodynamics. Due to the elastic deformation of the structure, they avoid discontinuities and gaps in the structures, which often cause turbulence.

Compliant mechanisms store stain energy in the structure when
providing its function. Thereby not the whole amount of input energy is transferred to the output, which limits the efficiency. But, this fact needs not to be a disadvantage in all cases. The energy may be regained after releasing, and the integration of additional springs becomes redundant. A further feature is the deformation induced stress in the structure, which has to be taken into account. Also fatigue and creep analysis have to be performed. Since compliant mechanisms are often used cyclonically, it is important to design the structure so that it has sufficient fatigue life to perform its function. In Table 1.1 the general features of compliant mechanisms/structures are summarized.

Perhaps the largest challenge is the synthesis of the compliant mechanism design. Knowledge and tool for the analysis as well as for the synthesis are required. Since the flexible members often undergo large deflection, linearized models are not longer valid. Nonlinear methods, that account for geometric nonlinearities and in some cases also for material nonlinearities, must be used. In the past, many compliant mechanisms were designed by trail and error. Appropriate and robust tools for the synthesis of real-world applications were missing.

<table>
<thead>
<tr>
<th>General features</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Single-piece devices</td>
</tr>
<tr>
<td>- Integrated functionality</td>
</tr>
<tr>
<td>- Stored strain energy</td>
</tr>
<tr>
<td>- Lightweight structures</td>
</tr>
<tr>
<td>- Free of backlash, friction and wear</td>
</tr>
<tr>
<td>- Ease of miniaturization</td>
</tr>
<tr>
<td>- No gap or break in the surface</td>
</tr>
<tr>
<td>- General tendency to fatigue and creeping behavior</td>
</tr>
</tbody>
</table>

Table 1.1: General features of compliant mechanisms/structures
1.2 State-of-the-art

The synthesis of compliant mechanisms is challenging. In contrast to conventional hinge solutions, a sequential dimensioning, whereas first the link is dimensioned and then the next hinge and so on, is not possible due to the interaction of the structural members. Thus, a simultaneous synthesis of the whole structure is necessary. Topology, shape, or sizing optimization techniques help finding best design solutions for compliant mechanism problems.

The synthesis methods can be categorized into three general areas; pseudo-rigid body methods, the ground structure methods and continuum-based methods. The pseudo-rigid body methods are kinematic type which start with a known rigid-link mechanism and convert it to a compliant mechanism [3]. There are three different types of the continuum-based methods; the homogenization methods [5, 6, 7], the element connectivity parameterization methods [8] and the level set methods [9].

1.2.1 Pseudo-rigid body methods

Pseudo-rigid models of the compliant mechanisms predict the deflection path for each flexible segment. The motion is modelled by rigid links attached at pin joints and the force/delection relationship by torsional springs, respectively. Figure 1.10 displays the pseudo-rigid deflection path approximation for a small-length flexural pivot.

1.2.2 Homogenization methods

The homogenization methods [10, 11] were originally developed for finding, for a given amount of material, structural topologies for maximum stiffness. They divide the geometric design space into a number of cells which can be either filled with material or void. Thus, finding the best solution implies an optimization problem with discrete parameters. Because of the astronomically high number of possible combinations it has proven to be very efficient to introduce a continuous density distribution for each cell. The density of each cell can vary between almost 0 and 1. The intermediate density values are mapped to the cell material properties transforming the originally discrete problem into a smooth
1.2 State-of-the-art

Figure 1.10: Pseudo-rigid deflection path approximation; a) Small-length flexural pivot, b) Pseudo-rigid model [3]
one and making it accessible to efficient mathematical programming solution techniques. A further feature of the homogenization method is, that the derivatives, which are needed for the mathematical programming, can be extremely easily computed according to the sensitivity formulation, since the derivative involves information at the element levels. Moreover, the minimum compliance problem has been proven to be convex. But, in contrast to the minimum compliance optimization, the goal of compliant mechanisms optimization is to maximize the displacement, the force or the work at the output.

A classical example for the compliant mechanism synthesis using the homogenization method is the modified 99 line topology optimization code by Sigmund [12, 13] for a force inverter as shown in Figure 1.11. Thereby, the goal is to maximize the work at the output of the inverter.

Figure 1.11: Force inverter according to Bendsøe

By specifying different values for the spring constant at the inputs and outputs, the displacement and the stiffness of the inverter mechanism can be controlled. Rahmatalla and Swan [14] introduced two different sets of springs. The first is an artificial spring set of relative large stiffness that prevents quasi-hinge solutions. The second is only attached to the output with smaller stiffness that represents the resistance of the workpiece as it is manipulated by the mechanism. Yin and
Ananthasuresh [15] applied two different methods for the design of distributed compliant mechanisms. The first method penalizes high-stress regions, and the second uses local, relative rotations and restrains it by using an objective function that makes the local deformation uniform throughout the structures. However, only the second method is successful in giving distributed compliant designs. Another primary issue in the synthesis of a distributed compliance mechanism is the efficient transfer of energy from input to output while providing the desired mechanical traits such as mechanical or geometric advantage [16]. The geometric advantage formulation maximizes the ratio of the output displacement over the input displacement and the mechanical advantage formulation optimizes the ratio of the output force over the input force, respectively.

The original homogenization method was developed for the topology optimization of isotropic, monolithic materials. Later, it has been extended to allow for anisotropic material models [11, 17], multi-material optimization [18], large-displacement [19, 20], and combined shape optimization strategies [21]. Hull and Canfield [22] introduced an approach, which subdivides each cell based on control points. The control points help smoothing the surface and creating a manufacturable design free of some numerical instabilities.

Although the homogenization method in combination with mathematical programming has proven to be very fast for continuous and convex problems, it cannot handle typical structural optimization problems that are of discrete and non-convex nature. It identifies only local minima of non-convex problems. Another drawback is the mesh dependency of the solutions. It was observed, that the solutions differ on each other depending on the fineness of the Finite Element (FE) mesh. In addition, side-effects such as checkerboard have to be prevented. The checkerboard effect is an artificial stiffness, which is caused by the finite element formulation and a sequence of filled and void cells.

Wang et al. [23] overcame the limitation of non-convex problems and solved the original discrete black-and-white (01) problem using a bit-array representation method in combination with connectivity analysis and genetic algorithms. Akhtar et al. and others [24, 25, 26] combined the Bezier curves representation with the graph theory and genetic algorithms. The Bezier curves represent the underlying
topology of the structures. In a second step the curves are mapped into a continuum finite element model (FEM).

1.2.3 Element connectivity parameterization methods

In order to avoid numerical difficulties caused by low-density values and large deformation in nonlinear analysis, the element connectivity parameterization (ECP) methods [8, 27, 28] optimize instead of the element density the element connectivity. Artificial zero-length elastic links are introduced to determine the element connectivity. If the link stiffness becomes small or large, the adjacent elements are assumed to be disconnected or connected, respectively. Thereby, the adjacent elements have the same properties throughout the whole design process and only the elastic link properties are changing.

1.2.4 Level set methods

Luo et al. [9] presented a parameterization level set approach for shape and topology optimization using compactly supported radial basis function. With the level set method [18, 29], the original optimization problem is transformed into a size optimization problem in which the expansion coefficients of the interpolants serve as a limited number of design variable. The contours of the parameterized family of level-set functions are used to generate the boundaries of a structure, and the topology can change with changes in the level-set function. The level set methods are a relative new technique, which is receiving growing attentions by the research community. For more information about the level set methods refereed to Wang et al. [30].

1.2.5 Ground structure method

Another way for the representation of the design is the use of ground structures such as beam or truss elements [31] (Figure 1.12). Often the cross sections of a fixed number of beam elements are optimized. A cross section closed to zero means there is not any connection. The ground structure approach reduces the number of parameters compared to the homogenization method. Frecker et al. [33], Saxena and Ananthasuresh [34, 35] and others have adapted the frame element-based
1.2 State-of-the-art

The results of an optimization are heavily influenced by the optimization algorithm. The fast gradient-based solution methods generally only identify local minima of non-convex problems, a problem overcome by Evolutionary Algorithms (EA). Although these typically require a much higher number of function evaluations, multi-criteria problems are set without significant additional effort. Genetic algorithms (GA), which belong to the evolutionary algorithms, were first introduced by Holland and his colleagues at the University of Michigan [39], and, since then have found application in many areas of engineering optimization. These algorithms are searching techniques based on the natural forces of evolution. They are robust, computationally simple yet capable in their search abilities, and do not require derivatives.

In the past few years the compliant mechanism design by genetic algorithms and ground structure approach has been given increasing attention by the scientific community. Parsons and Canfield [32] explored genetic algorithms for the multi-criteria topology optimization of compliant structures. Saxena [40] presented a procedure to synthesize compliant mechanisms for a prescribed nonlinear output path.
and extended it to a geometrical nonlinear, multi-material optimization [41]. Rai et al. [42] integrated initially curved frame elements in the genetic optimization. Lu and Kota [43, 44, 45] utilized a load path representation method to efficiently exclude the invalid topologies (disconnected structures) from the genetic algorithm solution space, in order to avoid useless function evaluations. Zhou and Ting [46] introduced the spanning-tree theory for the topological synthesis of compliant structures. The spanning-tree theory is based on the graph theory where a graph consists of vertices and edges. A valid topology is regarded as a network connecting input, output, support, and intermediate nodes (vertices), which contains at least one spanning-tree among the introduced nodes.

1.2.6 A general Evolutionary Algorithm

There is a huge variety of Evolutionary Algorithms and approaches in the literature. Nevertheless, a brief introduction to Evolutionary Algorithms (EAs) and an overview of the most common algorithms are given in this section. In addition, a general EA is outlined, which serves to explain the basic mechanisms of an Evolutionary system.

The main motivation of EAs is to overcome the limits of standard deterministic methods in many optimization problems [47]; when the search space involves both discrete and continuous domains, when the objective function or constrains lack regularity, or when the objective function admits a huge number of local minima. EAs employ a randomness which makes them less sensitive to noise, discontinuities, and the danger get trapped in a local optimum. While evolutionary algorithms do not possess highly efficient convergence properties like mathematical programming, they have the important feature of being inherently parallel. Nevertheless, when it is possible, e.g. in case of convex and continuous optimization problems, mathematical programming should be applied, since they are typically a magnitude faster than EAs.
EAs are inspired by the theory of Universal Darwinism. According to De Jong [48] the Darwinian evolutionary systems embody the following core components:

- one or more populations of individuals competing for limited resources,
- the notion of dynamically changing populations due to the birth and death of individuals,
- a concept of fitness which reflects the ability of an individual to survive and reproduce, and
- a concept of variational inheritance; offsprings closely resemble their parents, but are not identical.

Although there is a huge bandwidth of diversity in implementations, all EAs are based on these principles. Commonly they are divided into four categories with respect to their historical background:

- **Evolution Strategies** (ES), created by Ingo Rechenberg [49] in the 1960s, focused real-valued parameter optimization, whereas the individuals are represented by vectors of real-valued parameters. Today it is strongly promoted by Thomas Bäck [50].

- **Evolutionary Programming** (EP), developed by Lawrence Fogel [51] in the 1960s, concentrated on models involving a fixed-size population of parents, each of which produced a single offspring. It was developed further by his son David Fogel [52].

- **Genetic Algorithms** (GAs) were developed by John Holland [53]. The goal of the early GAs was to develop robust, adaptive and more application-independent algorithms. The basic terminology of genetic search and its principal components are discussed by Goldberg [39]. An introduction to the application of Genetic Algorithms to structural optimization using traditional binary string coding is given by Hajela [54].

- **Genetic Programming** (GP) is the most recent development in the field by John Koza [55]. It is a machine learning technique used to optimize a population of computer programs according to a fitness landscape of the optimization problem.
In spite of the different categories of EAs, they work the same. They start with the initialization of the first individuals that make up an evolving population. They perform the reproduction of individuals, either directly cloning parents or by using recombination and mutation operators to allow inheritance with variation. All EAs also use some form of selection to determine which solutions will have the opportunity to reproduce, and which will not. The key thing to remember about selection is that it exerts selection pressure, or evolutionary pressure to guide the evolutionary process towards specific areas of the search space. To do this, certain individuals must be allocated a greater probability of having offsprings compared to other individuals. The termination of an EA loop is normally based on solution quantity or time criteria. An example of a basic Evolutionary optimization loop is shown in Figure 1.13.

Figure 1.13: Example of a basic Evolutionary optimization loop
1.2 State-of-the-art

**Initialization**

The initialization can be performed either randomly or based on existing individuals. But typically, the first population is generated with entirely random values. Evolution is then used to discover which of the randomly sampled areas of the search space contain better solutions, and then to converge upon that area. Although the concept of EAs predicts, that all optimizations converge to the global minimum, if we wait infinitely, the initial population may influence the result. The reason for this is that the initial population has a strong influence on the speed of convergence and generally, the human patience is not infinite. Therefore, the initial population shall be set up carefully, spread over the entire search space. In some cases the integration of the user’s knowledge can be beneficial. Then, the first population is constructed from mutants of the user-supplied individuals.

It is also common to check the initial individuals for feasibility. Unfeasible individuals and individuals, that do not satisfy the constraints, are removed from the initial population and new ones are generated until a sufficient number of feasible individuals are obtained.

**Evaluation**

The evaluation is needed to assess each individual. Thereby, the evaluation involves fitness functions, in order to assign fitness values to the solutions (individuals). Fitness functions consist of at least one objective and of an arbitrary number of constraints. It could be observed that the implementation of fitness function often exerts strong influence on the performance of EAs [56]. Mostly, the coded solutions (genotypes) have to be mapped onto actual solutions (phenotypes), before the fitness of each solution can be determined. Typically, it is distinguished between the search space and the solution space. The search space represents the space of the genotypes, and the solution space is the space of phenotypes, respectively.

Generally, a single run of an EA will involve thousands of evaluations, which means that almost all computation time is spent performing the evaluation process. In structural optimization, evaluation is often performed by dedicated analysis software which can take minutes or even hours to evaluate a single solution. Thus, a strong emphasis exists towards reducing the number of evaluations during evolution.
Selection
The most forms of EAs perform a certain kind of parent selection. The parent selection is responsible for the identification of parent individuals of the next generation, and thereby, it has an essential influence on the creation of the next offsprings. Choosing the fitter individuals to be parents is the most common and direct way of inducing a selective pressure towards the evolution of fitter solutions. Two basic categories of selection mechanisms are known;

- deterministic and
- stochastic selection methods.

Examples of deterministic methods are uniform and truncation selection. The uniform method selects all individuals independently of their fitness. The truncation method sorts the individuals by the objective fitness, and the top of the individuals are selected, the remaining individuals die. The stochastic selection methods such as fitness ranking, tournament selection or fitness proportionate selection, choose the individuals with an individual probability, which is related to their fitness. Fitness ranking sorts the population in order of the fitness values and bases the probability of a solution being selected for parenthood on its position in the ranking. Tournament selection bases the probability of a solution being selected on how many other randomly picked individuals it can beat. Fitness proportionate selection (or roulette wheel selection) bases the probability of a parent being selected on the relative fitness score of each individual, e.g. a solution ten times as fit as another is ten times more likely to be picked as parent (Goldberg [39]).

Reproduction
Reproduction is the fundamental mechanism of EA, since it generates new solutions, the so-called offsprings, from parent solutions. Or, in other words, as long as new individuals are created, which differ from the parent individuals, evolution will occur. It is crucial, that the reproduction combines characteristics from both, inheritance and variation between child and parent. This is achieved by the use of the genetic operators: recombination and mutation.

Recombination operators require two or more parent solutions. The solutions (or genotypes) are shuffled together to generate child solu-
1.2 State-of-the-art

tions. EAs normally use recombination to generate most or all offspring. Recombination is normally performed by *crossover operators* in EAs.

Mutation operators modify a single solution. Some EAs mutate a copy of a parent solution in order to generate the child, some mutate the solution during the application of the recombination operators, others use recombination to generate children and then mutate these children. In addition, the probability of mutation varies depending on the EA.

While mutation operators help to explore new regions of the search space, crossover operators exploit the regions already explored by combining parental genes. The challenge is to find an effective balance between these two mechanisms. Crossover operators generally increase the speed of convergence, but they also increase the likelihood of convergence to a local minimum. In contrast, when the emphasis on mutation is too strong, the evolution becomes no more than a random search algorithm.

**Replacement**

Once offsprings have been generated, the next population has to be constructed. EAs usually maintain populations of fixed sizes, hence existing individuals must die, if new individuals are added. Two models of replacement exist:

- *overlapping-* or

- *non-overlapping-geneation* model

With the non-overlapping-generation model, the parent population dies off each generation and the offspring only compete with each other for survival. With the overlapping-generation model, the parents as well as their offsprings compete with each other. The last model provides a stronger competition, which results in an increased evolutionary pressure over a non-overlapping version.

Replacement needs not to be fitness-based, it can be based on constraint satisfaction, the similarity of genotypes, the age of solutions, or any other criterion.
1.3 Research needs and goals

In the following we identify the needs for research and define the goals of this thesis.

1.3.1 Research needs in the field of compliant structures

There are several different methods for the design of compliant mechanisms in the literature. But there are less appropriate design methods available to handle large-scale, highly constrained real-world structures. Often, these design methods are not able to handle discrete and rough (non-smooth) objective functions within the given design-space environment.

Design methodologies discussed in the literature do rarely take care of an efficient transfer of the structural result into a manufacturable design. We are convinced that a robust and adaptive optimization routine in combination with a fast and efficient final design routine is necessary in order to maximize the potential of compliant structures in real-world applications.

1.3.2 Goal of the thesis

Goal of the present research is to develop methods for the design of large-scale, compliant structures. Evolutionary Algorithms in combination with the ground structure concept is chosen for the synthesis of compliant structures. The idea of the synthesis is that the procedure distributes the ground structure in the design domain so that the structure morphs under load to a predefined shape as illustrated in Figure 1.14, and withstands the additional external loads. Since both small and large displacements are expected, the optimization shall provide geometrically linear or nonlinear modelling. We identified the following milestones:

**M1:** Development and implementation of a design methodology for load carrying, compliant structure based on Evolutionary Algorithms and ground structure concept

**M2:** Implementation of efficient and fast final design interpretation routines
1.4 Thesis outline

M3: Development of an adaptive car seat concept based on compliant structures including the realization of a functional prototype

1.4 Thesis outline

The thesis is organized as follows: Chapter 2 is concerned with the complex-shaped beam element as a ground structure. It explains the parametrization concept as well as the evaluation of the element stiffness matrix. In addition, an alternatively meshing concept is introduced. Chapter 3 presents the graph-based optimization. The graph topology representation and the necessary control routines are also explained. Numerical examples demonstrate the capability of the developed method. Furthermore, some extensions of the original method such as multi-layer and multi-material optimization as well as the enhanced mapping concept are discussed. Chapter 4 outlines the graph-based design environment that allows a fast interpretation of the design. Two different final design interpretation methods are discussed and their efficiency are compared with each other. In Chapter 5 the whole design methodology is validated by the adaptive car seat concept. The vertebra design as well as the compliant rib-like structure in combination with an integrated actuation system is presented. Finally, the thesis is concluded with Chapter 6 summarizing the findings and giving an outlook for further research in the fast growing field of compliant mechanisms and structures.
Chapter 2

Complex-shaped beam element

Compliant mechanisms replace discrete joints by identifying regions with increased compliance which can likewise be regarded as regions with low bending stiffness. Many compliant mechanism design problems require finding a best topology as well as an optimized shape. Often the two aspects are separated. A first method is used for finding the topology, and a second-step method refines the result by obtaining the best shape solution for the fixed topology. The complex-shaped beam element, used in this work, in combination with the ground structure approach opens up the possibility to combine the topology and thickness distribution optimization within one computational process. This appears attractive and motivates the development of a complex-shaped beam element.

The ground structure approach foresees a number of points which are multiply connected and it is obvious, from a mechanical point of view, to use beam elements for the connectors. The complex-shaped beam element is curved and of variable thickness so that a localized compliant region can be placed within one beam element (Section 2.1). For the representation the Hermite curve in combination with cubical thickness functions are chosen (Section 2.2), whereas the shape parameter $L$ influences the shape of the curve (Section 2.3). In order to prevent some redundancies and irregular shapes, the constant stretching concept is introduced (Section 2.4).
We implemented two ways to evaluate the complex-shaped beam’s structural properties. The first one uses the Castigliano’s theorem [57] (Section 2.5) for connecting the set of beam parameters with the beam’s structural properties because it provides a good compromise between accuracy and numerical costs for linear elasticity problems [58, 59]. But, the evaluation of stress and strain is not provided by this formulation. The second one overcomes this limitation by mapping the beam geometry into Finite Elements (FE). In addition, this meshing concept (Section 2.6) enables to integrate commercial FE tools providing linear as well as nonlinear analysis.

2.1 Complex-shaped beam element as a ground structure

The structural element beam is most useful for compliant mechanism design with ground structures [60], since it allows to efficiently adapt the stiffness and displacement of the beam element in the three directions $x$, $y$ and $z$. This feature is an import aspect for the design of compliant, but also load carrying structure.

Figure 2.1: Parameters of the beam; a) uniform, b) variable thickness beam
The geometrical parameters of a uniform beam element are displayed in Figure 2.1. The stiffness is a function of both material properties and geometry. In bending, the flexural stiffness is defined by $EI$, where $E$ is Young’s modulus, and $I$ is the cross-sectional moment of inertia, which depends on the thickness $t$ and the width $b$, respectively. For an axial load, the axial stiffness is given by $EA$, where $A = t \cdot b$ is the cross-sectional area.

In the following, the fact that bending is the most efficient mechanism to deform a beam structure, is pointed out. Of course, shearing and bending are always combined. Nevertheless, we calculate the forces that are needed to displace the end node by a unit length $\Delta u$ in case of axial loading, pure shearing and pure bending. Since the displacement $\Delta u$ is the same in all cases, the forces are proportional to the introduced energies. In the axial case, the axial force $N$ can be determined by

$$N = A \varepsilon E = \frac{tb \Delta u E}{l},$$

(2.1)

where $\varepsilon$ is the axial strain and $l$ the length of the beam. In the bending case, the transverse force $Q_B$ is

$$Q_B = \frac{3 \Delta u EI}{l^3} = \frac{\Delta u E l^3 b}{4l^3},$$

(2.2)

and in case of shearing the shear force $Q_S$ is

$$Q_S = \frac{\Delta u GA}{l} = \frac{\Delta u E t b}{l(1 + \nu)},$$

(2.3)

where $G$ is the shear modulus of a isotropic material and $\nu$ is its Poisson’s rate. We calculate the following ratios of energy in function of length $l$ to thickness $t$:

$$\frac{W_N}{W_B} = \frac{\Delta u N/2}{\Delta u Q_B/2} = \frac{4l^2}{l^2}$$

$$\frac{W_S}{W_B} = \frac{\Delta u Q_S/2}{\Delta u Q_B/2} = \frac{4l^2}{(1 + \nu)tl^2},$$

(2.4)

where $W_N$ is the stored energy in the axial case, $W_B$ in case of bending and $W_S$ in case of shearing. We choose length $l$ to thickness $t$ ratio of
10 and the Poisson’s rate \( \nu = 0.42 \) and get:

\[
\frac{W_N}{W_B} = 400.0
\]

\[
\frac{W_S}{W_B} = 281.7
\]

(2.5)

The stored energy for pure shearing and axial displacement is two orders of magnitude higher than for bending. It is much more efficient to displace the end node of a beam element transverse than along the beam element. Figure 2.2 displays the ratio of energy in function of the ratio \( l \) to \( t \).

![Figure 2.2: Comparison of the deformation mechanisms of a beam element depending on the length to thickness ratio](image)
Considering a compliant mechanism optimization, we generally optimize or maximize the output energy $W_{out}$ or work of a mechanism for a given input force or energy:

$$W_{out} = W_{in} - W_{stored},$$

(2.6)

where $W_{in}$ is the input energy and $W_{stored}$ the stored energy. Since the input energy $W_{in}$ is limited, the optimization tends to use that deformation mechanism that minimizes the stored energy $W_{stored}$ while providing the target functionality. For this reason bending is the preferred mechanism in compliant systems. We are arriving at the following conclusions summarizing the above considerations:

- The beam element is a most appropriate ground structure for the connection of the points.
- The beam element provides the required combination of stiff and compliant deformation modes.
- The beam element has to be placed transverse to the desired deformation direction providing the necessary flexibility along and the resistance transverse to the deformation direction.
- Bending is the preferred deformation mechanism in compliant beam structures.

If the length of the beam $l$ is given by some geometric relations, two parameters $t$ and $b$ remain, in order to adapt the stiffness in the directions $x$, $y$, and $z$. But $t$ and $b$ have a small influence on the geometrically nonlinear displacement path of the end nodes since the deformation path is highly depending on $l$. Thereby, the variation of the thickness along the centerline opens up the opportunity to manipulate the displacement path as shown in Figure 2.3. Three examples of different displacement paths are illustrated; the displacement path a) of a uniform beam, b) of a variable thickness beam, and c) of a curved, variable thickness beam. Of course, the complex-shaped beam element offers many other displacement paths, which are not shown in Figure 2.3.
Figure 2.3: Geometrically nonlinear displacement path; a) uniform beam, b) variable thickness beam, c) curved, variable thickness beam

Having beams with high shape complexity, such as varying thickness and curved centerline, not only allows the modification of the displacement path of the end nodes of each structural member, but also adds the flexibility necessary for placing compliant regions within the beam. Figure 2.4 illustrates the shape complexity and its parameters.

Figure 2.4: Curved, variable-thickness beam parameters
2.2 Parametrization of the complex-shaped beam element

For the curved centerline the Hermite curve representation

\begin{align}
   x(\sigma) &= a_0 + a_1 \sigma + a_2 \sigma^2 + a_3 \sigma^3 \\
   y(\sigma) &= b_0 + b_1 \sigma + b_2 \sigma^2 + b_3 \sigma^3
\end{align}

is chosen [58], where \( \sigma \) is a local normalized line coordinate ranging from \(-1\) to \(1\).

The coefficients of these two equations depend on the coordinates of the nodes with which the element is connected, the chosen angle at the nodes, and the factor \( L \). **Important note**, the shape parameter \( L \) is unequal to the arc length \( l \) of the beam. The curve must satisfy given endpoint conditions in terms of the positions of, and slopes at the endpoints \( P_1(x_1, y_1) \) and \( P_2(x_2, y_2) \).

\[
\frac{dy}{dx}|_{\sigma=-1} = \tan(\beta_1) \quad \frac{dy}{dx}|_{\sigma=1} = \tan(\beta_2)
\]
which gives the centerline in terms of the normalized parameter $\sigma$:

$$
\begin{align*}
x(\sigma) &= \frac{1}{4}
\left[
\begin{array}{l}
\left( 2 - 3\sigma + \sigma^3 \right) x_1 \\
+ \frac{L}{2} \left( 1 - \sigma - \sigma^2 + \sigma^3 \right) \cos(\beta_1) \\
+ \left( 2 + 3\sigma - \sigma^3 \right) x_2 \\
+ \frac{L}{2} \left( -1 - \sigma + \sigma^2 + \sigma^3 \right) \cos(\beta_2)
\end{array}\right], \\
y(\sigma) &= \frac{1}{4}
\left[
\begin{array}{l}
\left( 2 - 3\sigma + \sigma^3 \right) y_1 \\
+ \frac{L}{2} \left( 1 - \sigma - \sigma^2 + \sigma^3 \right) \sin(\beta_1) \\
+ \left( 2 + 3\sigma - \sigma^3 \right) y_2 \\
+ \frac{L}{2} \left( -1 - \sigma + \sigma^2 + \sigma^3 \right) \sin(\beta_2)
\end{array}\right]
\end{align*}
$$

(2.9) (2.10)

where $L$ influences the line length of the curved centerlines (See Figure 2.5)

$$
\frac{dx}{d\sigma}|_{\sigma=-1} = a_1 - 2a_2 + 3a_3 = \frac{L}{2} \cos(\beta_1),
$$

$$
\frac{dy}{d\sigma}|_{\sigma=-1} = b_1 - 2b_2 + 3b_3 = \frac{L}{2} \sin(\beta_1),
$$

$$
\frac{dy}{dx}|_{\sigma=-1} = \frac{dy(\sigma = -1)/d\sigma}{dx(\sigma = -1)/d\sigma} = \tan(\beta_1).
$$

(2.11)

The coefficients of the cubic thickness distribution

$$
t(\sigma) = t_0 + t_1 \sigma + t_2 \sigma^2 + t_3 \sigma^3
$$

(2.12)

are defined by thickness values at locations along the beam centerline as Figure 2.4 illustrates.

The parameter set for describing one complex-shaped beam element includes the values of

- 4 endpoint coordinates
- 2 centerline run-out angles at the endpoints,
- 1 shape parameter $L$, and
- 4 thicknesses at four points along the centerline.

The choice of the material, i.e. for multi-material optimization, adds the twelfth parameter to the set. The shape complexity requires six more parameters than would be necessary if simple beam elements were used.
2.3 Influence of shape parameter $L$

The parameter $L$ appearing in Eq. 2.11 influences the shape of the curve. We have the alternative of either use $L$ as a free shape parameter or fix it by making it dependent on the other parameters.

We intuitively believe that the first would render an optimization procedure less efficient for two reasons. First, there is some redundancy between $L$ and the other parameters in affecting the shape which will slow down the optimization process speed. Secondly, very large values of an unconstrained parameter $L$ may lead to curves with unpleasing shapes containing undesired features such as loops, cusps, or folds. Such features of unpleasing shapes are useless for the desired compliant mechanism designs and will, if admitted, also stand in the way of obtaining design solutions quickly. Producing Hermite curves with a pleasing shape is a research objective [61, 62] as mentioned by Yong and Cheng [63].

![Figure 2.6: Influence of shape parameter $L$; a) $L=0$, b) $L=2$, c) constant stretching and d) constant curvature concept](image-url)
All the curves plotted in Figure 2.6 satisfy the endpoint conditions of the shape problem studied by Horn [64], You and Wan [65] and mentioned by Moreton and Squin [66],

\[ x_1 = 0, \quad y_1 = 0, \quad \beta_1 = 90^\circ \]
\[ x_2 = 2, \quad y_2 = 0, \quad \beta_2 = -90^\circ \]

but differ from each other by the influence of the shape parameter \( L \) which affects, of course, the shape and length \( l \) of the curves. \( l \) is the total arc length of the curve

\[ l = \int_{-1}^{1} s_{,\sigma} \, d\sigma \]

where we use the normalized independent parameter \( \sigma \in [-1,1] \) and at any point \( \sigma \) we find the variation of the infinitesimal ratio \( s_{,\sigma} \):

\[ s_{,\sigma} = \sqrt{x_{,\sigma}^2 + y_{,\sigma}^2} \]

(2.15)

The values of \( L \) and \( l \) are listed in Table 2.1 and will also be discussed in the following. Setting \( L = 0 \) produces the straight line between the endpoints. Its actual length is therefore \( l = 2 \). The circle markers show that evenly spaced points on \( \sigma \) are mapped onto points on \( s \) whose distance varies. More specifically, the point density increases toward the endpoints.

Choosing \( L \) equal to the distance between the endpoints, \( L = 2 \), produces the curve with arc length \( l = 2.442552 \) whose markers indicate a more evenly spaced point distribution.

This observation inspires to adjust \( L \) so that distortion of points on \( s \) becomes a minimum, giving \( L = 3.237188 \). This value of \( L \) happens to be close to the length \( \pi = 3.1416 \) of a semi-circle with radius \( r = 1 \) but

<table>
<thead>
<tr>
<th>( L )</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>2.000</td>
<td>3.237</td>
<td>4.083</td>
<td></td>
</tr>
<tr>
<td>2.000</td>
<td>2.443</td>
<td>2.877</td>
<td>3.205</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Shape parameter \( L \) and curve length \( l \) of the curves plotted in Figure 2.6
2.3 Influence of shape parameter $L$

the actual curve length is only $l = 2.877$ and the curve does not match a semi-circle very well. We call the concept of minimum distortion of points on $s$ the constant stretching concept and it is stated in Eq. 2.17. A circle is a curve with constant curvature and this inspires to adjust $L$ to minimize the standard deviation of the curvature distribution along the cubic curve. We call it the constant curvature concept. This gives $L = 4.083$ with an actual curve length $l = 3.205$ which is a good approximation of the semi-circle.

The considered curve problem illustrates the here summarized parameterization concepts for determining the shape parameter $L$:

(a) set it equal to zero
(b) set it equal to the Euclidian distance between the endpoints
(c) minimize with it the deviation from constant point spacing
(d) minimize with it the deviation from constant curve curvature

Concept (a) is trivial because it annihilates the possibility of presenting curved lines and is therefore neither desired nor further considered. Concept (b) falls together with the standard cubic Hermite curve presentation and is most convenient to implement because

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.16)$$

is calculated with no numerical burden. An interest in achieving a point distribution as close as possible to constancy would favor concept (c), where the shape parameter $L$ follows from minimizing

$$F = \int_{-1}^{1} (s_{s, \sigma} - \bar{s}_{s, \sigma})^2 d\sigma = \min, \quad (2.17)$$

where $\bar{s}_{s, \sigma}$ is the average value of $s_{s, \sigma}(\sigma)$. Circle sections are approximated best with concept (d) which is attractive because a circle is the geometrically simplest of all curves. Then the shape parameter $L$ minimizes

$$G = \int_{-1}^{1} (\kappa - \bar{\kappa})^2 d\sigma = \min, \quad (2.18)$$

where $\bar{\kappa}$ is the average value of $\kappa(\sigma)$. 

Both, the conditions Eq. 2.17 and Eq. 2.18 can be satisfied by using a numerical optimization scheme. The iterations can be terminated if the optimality criterion

\[ F_{iL} = 0 \quad \text{or} \quad G_{iL} = 0 \]  

(2.19)

is satisfied within some chosen tolerance level. It is therefore convenient to cast objective functions in the form

\[ O_F = (F_{iL})^2 \quad \text{or} \quad O_G = (G_{iL})^2 , \]  

(2.20)

because the minimum point values of these are equal to zero.

Although d) is a viable alternative, we decide to employ c), since it ensures a constant projection of the thickness distribution \( t(\sigma) \) along the centerline independent of the curvature as illustrated in Figure 2.7.

![Figure 2.7: Shape dependence on parameter \( L \) using the same thickness function \( t(\sigma) \) and the same \( \beta_1 \) and \( \beta_2 \); a) \( L=0 \), b) \( L=2.7 \), c) constant stretching concept](image-url)
2.4 Constant stretching concept

The points on $\sigma$ are distributed over a normalized length of 2. Those on $s$ are stretched over the total arc length $l$. It is therefore that the average spacing $s_{\sigma}$ equals one-half of the total arc length $l$.

$$F = \int_{-1}^{1} \left( s_{\sigma} - \frac{l}{2} \right)^2 d\sigma = \text{min.} \quad (2.21)$$

The expanded form of Eq. 2.21 we split into three individual terms

$$F = \int_{-1}^{1} s_{\sigma}^2 d\sigma - \int_{-1}^{1} s_{\sigma} d\sigma l + \frac{l^2}{4} \int_{-1}^{1} d\sigma = \text{min} \quad (2.22)$$

where the second integral gives the actual line length $l$. Thus we obtain the simple form

$$F = \int_{-1}^{1} s_{\sigma}^2 d\sigma - \frac{l^2}{2} = \text{min} \quad (2.23)$$

 Extreme values of $F$ are found where its derivative with respect to $L$ vanishes and this condition obtains

$$F_{,L} = \left( \int_{-1}^{1} s_{\sigma}^2 d\sigma \right)_{,L} - ll_{,L} = 0 \quad (2.24)$$

Because of

$$s_{\sigma} = \sqrt{x_{\sigma}^2 + y_{\sigma}^2} \quad (2.25)$$

we have

$$F_{,L} = \left( \int_{-1}^{1} (x_{\sigma}^2 + y_{\sigma}^2) d\sigma \right)_{,L} - ll_{,L} = 0 \quad (2.26)$$

The first integral appearing in the condition Eq. 2.26 can be solved analytically. The practical benefit of this fact is, however, limited since the total arc length $l$, requiring solution of an integral Eq. 2.14, and its derivative with respect to the parameter $L$ must be left to numerical evaluation.

But the equation of the constant stretching Eq. 2.17 concept could not be explicitly resolved for $L$. Although the numerical procedures give reliable and meaningful results, it would be a serious drawback having to implement them into the pre-processing step of a finite-element program using curved beam elements for whom the parameterization concepts are intended. Therefore we derived closed-form
equations, for approximating the numerical results for the shape parameter $L$, depending on trigonometric functions of the endpoint conditions $\beta_1$ and $\beta_2$. Interested readers may refer to Appendix A.

### 2.5 Beam-theory approach

We use the Castigliano’s theorem [57], in order to connect the complex-shaped beam element with its structural properties [58].

Timoshenko [67] calculates deflections at the free end of clamped curved thin bars, which have the shape of a quarter circle, by the use of the Castigliano’s theorem. He assumes that only bending occurs and consequently lets only the local curvature $\kappa$ contribute to the deformation energy $U$:

$$U = \int_0^{\frac{\pi}{2}} \frac{M^2}{2EI} ds$$ (2.27)

The internal bending moment $M$ is derived from global equilibrium with the forces acting upon the free end and the displacements at it are calculated by

$$u = \frac{\partial U}{\partial F_{Bx}}, \quad v = \frac{\partial U}{\partial F_{By}}, \quad \varphi = \frac{\partial U}{\partial M_B}$$ (2.28)

As have others [68, 69, 70, 71, 72], we use this as an inspiration and consider a more generally curved beam indicated by Figure 2.8. Points along the centerline of the beams are given in parametric form $x(s)$ and $y(s)$. The internal forces depend on the externally applied loads

![Figure 2.8: Coordinates for defining the centerline shape through endpoints](image-url)
and the curved shape
\[ F_x(s) = F_{Bx} \] (2.29)
\[ F_y(s) = F_{By} \] (2.30)
\[ M(s) = M_B - (y_B - y(s)) F_{Bx} + (x_B - x(s)) F_{By} \] (2.31)
and must be decomposed into components tangential and normal to the centerline direction \( \beta(s) \):
\[
\begin{bmatrix}
N(s) \\
Q(s) \\
M(s)
\end{bmatrix}
= 
\begin{bmatrix}
\cos\beta(s) & \sin\beta(s) & 0 \\
-\sin\beta(s) & \cos\beta(s) & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
F_x(s) \\
F_y(s) \\
M(s)
\end{bmatrix}.
\] (2.32)

We assume cross sections of a rectangular shape with variable thickness \( t(s) \) and unit width so that the values of the area \( A \)
\[ A = t(s) \] (2.33)
and the second moment of inertia \( I \)
\[ I = \frac{t(s)^3}{12}. \] (2.34)
are also functions of \( s \). The shear deformation \( \gamma \) we capture by first-order shear theory and for it we need the corrected shear area \( A^* = kA \) with the shear correction factor \( k = 1.2 \). However, we intend to develop a beam model as general as possible and realize that curved shapes cause strains to be no longer linearly distributed over the thickness. Furthermore, the apparent stiffness values of cross-sections of strongly curved beams increase. Timoshenko bases a model to account for the curvature effect on the observation that the neutral fiber is shifted towards the center of curvature [67]. Other authors model the same effect by using the coupling-stiffness-matrix concept of laminated plate theory [73, 74]. The local radius of the shape curvature \( R(s) \) is the reciprocal \( R = 1/\chi \) of the local curvature \( \chi \) and this we calculate, because of \( x = x(s) \) and \( y = y(s) \), as
\[ \chi = \sqrt{x''^2 + y''^2}. \] (2.35)
We introduce the local thickness coordinate \( \ddot{y} \) as shown in Figure 2.9 and a variable radius \( r \) which depend on each other by
\[ r = \begin{cases} 
R - \ddot{y}, & \chi > 0 \\
R + \ddot{y}, & \chi < 0 
\end{cases}. \] (2.36)
The line element displayed in Figure 2.9 is parallel to the beam center line and change their lengths linearly through the thickness and recalling the definition

\[ \varepsilon_s = \frac{\partial u}{\partial s} \]  \hspace{1cm} (2.37)

helps to understand that

\[ \varepsilon_s(\bar{y}) = \frac{s(0)}{s(\bar{y})} \varepsilon_s^0 = \frac{R}{r} \varepsilon_s^0 = Y \varepsilon_s^0 \] \hspace{1cm} (2.38)

where \( \varepsilon_s^0 \) is the direct strain in the beam centerline. More generally the direct strain distributions \( \varepsilon_s \) depend on the strain \( \varepsilon_s^0 \) of the beam reference fiber and the curvature \( \kappa \)

\[ \varepsilon_s = Y \varepsilon_s^0 + \bar{y} Y \kappa_s. \] \hspace{1cm} (2.39)

For the shear stress distribution it holds that

\[ \gamma_s = Y \gamma_s^0. \] \hspace{1cm} (2.40)

Figure 2.10 illustrates the relationship between the coordinates \( \bar{y} \) and \( r \). The deformation energy of the curved beam element is obtained from integrating through the thickness and through the arc length

\[ U = \frac{1}{2} E \int_0^l \int_{-\frac{1}{2}}^{\frac{1}{2}} \left( \varepsilon_s^2 + \kappa \gamma_s^2 \right) \frac{1}{Y} \, d\bar{y} \, ds = \int_0^l U_s \, ds. \] \hspace{1cm} (2.41)
2.5 Beam-theory approach

The internal loads $N(s)$, $Q(s)$, and $M(s)$ are obtained from derivative of the local internal energy $U_s$.

$$
N(s) = \frac{\delta U_s}{\delta \varepsilon_s} \quad M(s) = \frac{\delta U_s}{\delta \kappa_s} \quad Q(s) = \frac{\delta U_s}{\delta \gamma_s}.
$$

To evaluate the integrals $\ddot{y}Y(\ddot{y})$, and $\ddot{y}^2Y(\ddot{y})$

$$
U_s = \int_{\frac{-R}{2}}^{\frac{R}{2}} \frac{1}{2} E(Y \varepsilon_s^0)^2 + 2\dot{y}Y \varepsilon_s^0 \kappa_s + \ddot{y}^2 Y \kappa_s^2 + Y k \gamma_s^0)^2 d\ddot{y}.
$$

of the function $\ddot{y}$, we find it convenient to start with the power series expansion of $Y(\ddot{y})$ by writing it in the form

$$
Y(\ddot{y}) = 1 + \left( \frac{\ddot{y}}{R} \right) + \left( \frac{\ddot{y}}{R} \right)^2 + \left( \frac{\ddot{y}}{R} \right)^3 ... = 1 + \left( \frac{\ddot{y}}{R} \right) + \left( \frac{\ddot{y}}{R} \right)^2 + O^3 \left( \frac{\ddot{y}}{R} \right)
$$

obtained from Bronstein and Semendjajew [75] where the sum of the terms of order three and higher is abbreviated by using the symbol $O^3 \left( \frac{\ddot{y}}{R} \right)$. The convenience is that from the series Eq. 2.44 the series $\ddot{y}Y(\ddot{y})$ and $\ddot{y}^2Y(\ddot{y})$

$$
\ddot{y}Y(\ddot{y}) = \ddot{y} + \frac{1}{R} \ddot{y}^2 + RO^3 \left( \frac{\ddot{y}}{R} \right)
$$

$$
\ddot{y}^2Y(\ddot{y}) = \ddot{y}^2 + R^2 O^3 \left( \frac{\ddot{y}}{R} \right)
$$

Figure 2.10: Definition of Function $Y(y)$ (after [74])
are immediately derived and render the evaluation of the integral in Eq. 2.43:

\[
U_s = \frac{1}{2} E \left( \varepsilon_s^0 \left( t + \frac{t^3}{12R^2} \right) + \varepsilon_s^0 \kappa_s + \kappa_s^2 + \kappa_s^2 \left( t + \frac{t^3}{12R^2} \right) \right) + E \int_{\frac{\gamma}{R}}^{\frac{\gamma}{R}} O^3 \left( \frac{\gamma}{R} \right) d\gamma.
\]

(2.46)

\[
A^* = t + \frac{t^3}{12R^2} + T = A + \frac{1}{R^2} D + T
\]

(2.47)

\[
B^* = \frac{t^3}{12R} + T = \frac{1}{R} D + R T
\]

(2.48)

\[
D^* = \frac{t^3}{12} + T = D + R^2 T
\]

(2.49)

Here, \( T \) is the integral with respect to the sum of the higher-order terms \( O^3 \left( \frac{\gamma}{R} \right) \):

\[
T = \int_{\frac{\gamma}{R}}^{\frac{\gamma}{R}} O^3 \left( \frac{\gamma}{R} \right) d\gamma.
\]

(2.50)

As a result, the following relations between line loads and deformations hold:

\[
\begin{pmatrix}
N \\
M \\
Q
\end{pmatrix}
= E \begin{bmatrix}
A^* & B^* & 0 \\
B^* & D^* & 0 \\
0 & 0 & kA^*
\end{bmatrix}
\begin{pmatrix}
\varepsilon_s^0 \\
\kappa_s \\
\gamma_{s\gamma}
\end{pmatrix}
\]

(2.51)

It is analogous to the ABD matrix of the so-called classical lamination theory (CLT) \cite{76} with an extension to account for shear stiffness. It appears that centerline curvature couples extension and bending even if the beam is made from a homogeneous material. We introduce the compliance matrix \( \text{abd}^* \) and obtain it by inverting the stiffness matrix \( \text{ABD}^* \). With it we write the deformation energy of a beam of a more general shape including shear and curvature effects:

\[
U = \frac{1}{2} \int_A^B \left( a^* N(s)^2 + 2b^* N(s)M(s) + d^* M(s)^2 + a_s^* Q(s)^2 \right) ds
\]

(2.52)
The displacements at point B depicted in Figure 2.8 follow from applying the Castigliano’s theorem. The derivation of the deformation energy with respect to the global forces must obey the rule of chains:

\[
\frac{\partial U}{\partial F_i} = \int_A^B \left[ (a^* + b^*) N \frac{\partial N}{\partial F_i} + (b^* + d^*) M \frac{\partial M}{\partial F_i} + a^*_s Q \frac{\partial Q}{\partial F_i} \right] ds \quad (2.53)
\]

and the symbol \( F_i \) stands for \( F_{Bx}, F_{By} \), and the externally applied moment \( M_B \). The derivations of the internal forces with respect to the externally applied forces are:

\[
\frac{\partial N}{\partial F_{Bx}} = \cos \beta(s) \quad (2.54)
\]

\[
\frac{\partial N}{\partial F_{By}} = \sin \beta(s) \quad (2.55)
\]

\[
\frac{\partial Q}{\partial F_{Bx}} = -\sin \beta(s) \quad (2.56)
\]

\[
\frac{\partial Q}{\partial F_{By}} = \cos \beta(s) \quad (2.57)
\]

\[
\frac{\partial M}{\partial F_{Bx}} = -(y_B - y(s)) \quad (2.58)
\]

\[
\frac{\partial M}{\partial F_{By}} = x_B - x(s) \quad (2.59)
\]

\[
\frac{\partial M}{\partial M_B} = 1 \quad (2.60)
\]

Using this obtains the formulae for calculating the displacements:

\[
u_B = \int_A^B \left[ (a^* + b^*) N \cos \beta - (b^* + d^*) M (y_B - y) - a^*_s Q \sin \beta \right] ds \quad (2.61)
\]

\[
v_B = \int_A^B \left[ (a^* + b^*) N \sin \beta + (b^* + d^*) M (x_B - x) + a^*_s Q \cos \beta \right] ds \quad (2.62)
\]
\[ \varphi_B = \int_A^B (b^* + d^*) M ds \]  

(2.63)

In the case of complex-shaped beams, closed-form evaluation of the integrals would be tedious, if possible, and we content ourselves with a numerical integration scheme where the number of supporting points can be chosen high enough to achieve desired accuracy levels. On the other hand, if choosing \( s = x, x_A = 0, x_B = l \) and clamping at \( A \), one verifies easily that Eq. 2.61 through Eq. 2.63 reproduce the well-known formulae for cantilever beams:

\[ u(l) = \frac{F_x(l)}{EA} \]  

(2.64)

\[ v(l) = \frac{F_y(l)l^3}{3EI} + \frac{M(l)l^2}{2EI} + \frac{F_y(l)}{EA^*} \]  

(2.65)

\[ \varphi(l) = \frac{M(l)l}{EI} \]  

(2.66)

These are the constitutive equations from which simpler beam finite elements are developed.

### 2.5.1 Beam Finite Element stiffness matrix

Finite-element theory transforms a given mechanical boundary-value problem into a system of numerical equations

\[ K \ddot{u} = r \]  

(2.67)

where the coefficient matrix \( K \) is called stiffness matrix, \( \ddot{u} \) presents the degrees-of-freedom of the unknown displacement solution, and the right-hand side \( r \) reflects discrete nodal forces stemming from the boundary conditions and other loads.

In order to determine the stiffness matrix \( K \), we calculate the displacements at the free end of a cantilever beam due to applied force. Applying the unit forces \( F_{Ax} = 1, F_{Ay} = 1, \) and \( M_A = 1 \) individually to
the beam clamped at point $B$, as the sketch at the top of Figure 2.11 illustrates, obtains the corresponding displacement vectors $\{\tilde{u}_A, \tilde{v}_A, \tilde{\phi}_A\}$ which are assembled in a compliance matrix $S_A$ with three rows and columns:

$$
\begin{bmatrix}
    s_{A11} & s_{A12} & s_{A13} \\
    s_{A12} & s_{A22} & s_{A23} \\
    s_{A13} & s_{A23} & s_{A33}
\end{bmatrix}
\begin{bmatrix}
    F_{Ax} \\
    F_{Ay} \\
    M_A
\end{bmatrix}
= 
\begin{bmatrix}
    \tilde{u}_A \\
    \tilde{v}_A \\
    \tilde{\phi}_A
\end{bmatrix}.
$$

(2.68)

Clamping point $A$, as the bottom sketch in the figure shows, and applying analog unit forces at $B$ gives the other compliance matrix $S_B$:

$$
\begin{bmatrix}
    s_{B11} & s_{B12} & s_{B13} \\
    s_{B12} & s_{B22} & s_{B23} \\
    s_{B13} & s_{B23} & s_{B33}
\end{bmatrix}
\begin{bmatrix}
    F_{Bx} \\
    F_{By} \\
    M_B
\end{bmatrix}
= 
\begin{bmatrix}
    \tilde{u}_B \\
    \tilde{v}_B \\
    \tilde{\phi}_B
\end{bmatrix}.
$$

(2.69)

Figure 2.11: Coordinates for defining the centerline shape through endpoints

The desired beam finite element stiffness matrix $K$ for six degrees of freedom is derived in two steps. First the compliance matrix $S_A$ is inverted and the result gives a sub-matrix of $K$ for $1 \leq i \leq 3$ and $1 \leq j \leq 3$. The inverse of $S_B$ gives the other sub-matrix where $4 \leq i \leq 6$.
and $4 \leq j \leq 6$. The structure of the incomplete stiffness matrix $K^*$ is

$$K^* = \begin{bmatrix} S_A^{-1} & 0 \\ 0 & S_B^{-1} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \\ k_{44} & k_{45} & k_{46} \\ k_{54} & k_{55} & k_{56} \\ k_{64} & k_{65} & k_{66} \end{bmatrix}. \quad (2.70)$$

The void entries of it are filled by using global equilibrium to obtain reaction forces at the respective clamped ends. The procedure begins with setting $u_A$ equal to unity and all other displacements to zero. This displacement pattern raises forces where the incomplete stiffness matrix contains in its first column the values $F_{Ax}, F_{Ay}$, and $M_A$. The lacking entries of the first column of $K$ are the reaction forces $F_{Bx}, F_{By}$, and $M_B$ at the clamped endpoint $B$. Global equilibrium dictates that

$$\begin{align*}
F_{Ax} + F_{Bx} &= 0 \\
F_{Ay} + F_{By} &= 0 \\
M_A + M_B - F_{xB}(y_B - y_A) + F_{yB}(x_B - x_A) &= 0 \\
M_A + M_B + F_{xA}(y_B - y_A) - F_{yA}(x_B - x_A) &= 0 
\end{align*} \quad (2.71)$$

which obtains that

$$\begin{align*}
k_{4j} &= -k_{1j} \\
k_{5j} &= -k_{2j}, \quad 1 \leq j \leq 3. \\
k_{6j} &= -k_{3j} - (y_B - y_A)k_{1j} + (x_B - x_A)k_{2j} 
\end{align*} \quad (2.72)$$

The other void is filled by

$$\begin{align*}
k_{1j} &= -k_{4j} \\
k_{2j} &= -k_{5j}, \quad 4 \leq j \leq 6. \\
k_{3j} &= -k_{6j} + (y_B - y_A)k_{4j} - (x_B - x_A)k_{5j} 
\end{align*} \quad (2.73)$$

This completes the beam finite element stiffness matrix $K$. 

*Complex-shaped beam element*
2.6 Plane-elements approach

Instead of evaluating the stiffness matrix by the use of the Castigliano’s theorem, the alternative concept maps the complex-shaped beam geometry into a FE mesh of plane-stress or plane-strain elements. While the beam theory approach provides a very fast determination of the structural properties for linear elasticity problems, the meshing of the beam geometry enables to efficiently evaluate the stress as well as stain and it opens up the opportunity to employ commercial FEM tools providing linear and nonlinear analysis. It is obvious that the number of elements drastically increases by mapping the beam geometry into a FE mesh and thus, the size of global stiffness matrix increases by several orders of magnitude. Some additional numerical costs are compensated by the fact that the numerical integration over the beam geometries is not necessary anymore.

The parametrization of the complex-shaped beam geometry remains the same, only, the determination of the structural behavior is differently performed. Figure 2.12 displays the meshed sample geometries; a) coarse mesh, b) medium mesh, c) fine mesh. The meshing of the complex-shaped beam geometry is provided by the freely available mesh generator Triangle [77]. Triangle is a commonly used, two-dimensional mesh generator tool, which was developed by Jonathan Shewchuk. It reads and writes ASCII files.

In order to avoid the bending blocking effects of simple tree-nodal triangular elements, the tree-nodal elements are transferred into six-nodal plane elements by adding nodes in the middle of each edge of the elements. Unlike nodes, that are connected by beam elements, nodes of two-dimensional plane elements have only two degrees of freedom, displacement $u$ in x-direction and $v$ in y-direction, and do not transfer rotations or moments. Artificial beams are added in order to transfer moments and/or rotations. They connect the main node, which are given by the structural beam representation and link the beam geometries, and nodes at the end of the beam representation as shown in Figure 2.13. Without doing so, the structure would perform like link mechanisms. A very high Young’s modulus and an infinitesimal small density is assigned to the artificial beams by reason that the purpose of the artificial beams only are to transfer the rotation and/or moments and not to influence the mass balance of the FE computation.
Figure 2.12: Meshed beam geometry; a) coarse mesh, b) medium mesh and c) fine mesh

Figure 2.13: Connection to the main node
Table 2.2 displays the structural behavior of differently meshed beam geometries in comparison to Castigliano, when applying a unit force $F = 1$ in $y$-direction. The fine-meshed geometry approximates the result of Castigliano very precise. But the coarse meshed geometry is 40.1\% stiffer than Castigliano. This is a dramatic deviation and can not be neglected. For straight beam geometries the differences are negligibly small. This effect always appears when the contour line is highly curved. The stiffening is caused by a bad approximation of the outer contour of the compliant zone within the beam geometry. This fact is illustrated in Figure 2.14.

Since the bending stiffness depends on the thickness to the power of three, it has a significant influence on the structural behavior. By shifting the intermediate nodes towards the outer contour line (Figure 2.15),

<table>
<thead>
<tr>
<th>displacement</th>
<th>x [mm]</th>
<th>y [mm]</th>
<th>$\varphi$ [rad]</th>
<th>deviation [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Castigliano</td>
<td>-0.07</td>
<td>1.36</td>
<td>0.021</td>
<td></td>
</tr>
<tr>
<td>Fine mesh</td>
<td>-0.07</td>
<td>1.37</td>
<td>0.021</td>
<td>0.4%</td>
</tr>
<tr>
<td>Medium mesh</td>
<td>-0.07</td>
<td>1.29</td>
<td>0.020</td>
<td>5.2%</td>
</tr>
<tr>
<td>Coarse mesh</td>
<td>-0.04</td>
<td>0.82</td>
<td>0.012</td>
<td>40.1%</td>
</tr>
<tr>
<td>Mod. medium mesh</td>
<td>-0.07</td>
<td>1.34</td>
<td>0.020</td>
<td>1.4%</td>
</tr>
<tr>
<td>Mod. coarse mesh</td>
<td>-0.07</td>
<td>1.24</td>
<td>0.019</td>
<td>9.0%</td>
</tr>
</tbody>
</table>
Figure 2.15: Mesh modification; a) original, b) modified mesh

the approximation can be drastically improved. The deviation of the modified coarse-meshed beam geometry is about 9.0\%, and the deviation of the modified medium-meshed beam geometry is less than 1.5\%, which is tolerable.

An example of a compliant gripper model consisting of meshed beam geometries is given in Figure 2.16. The half symmetric model is composed of 167 plane and beam elements compared to the seven complex-shaped beam elements of the beam-theory approach.

Figure 2.16: Alternatively meshed gripper
Chapter 3

Graph-based optimization

Topography, shape, or sizing optimization techniques help identifying design solutions of complex compliant mechanism problems. This thesis introduces an approach to solve compliant-mechanism problems that applies evolutionary optimization techniques where the use of graph theory and a complex-shaped, two-dimensional beam element can increase both numerical efficiency and solution space.

The ground structure topology is represented by a mathematical graph whose edges and vertices correspond to the beams and their nodes, respectively. All genetic operators, i.e. mutation and crossover, directly apply on the graph representation (vertices and edges).

The graph-based representation allows the simultaneous optimization of topology, geometry, sizing of the beam structures as well as that of the support/loading and the material (See Figure 3.1). The graph-based data structure thereby provides a very flexible and adaptive representation of the structures as well as of the support and loading, which is the basic requirement for the optimization of all the above-mentioned disciplines.

The original graph-theory method was introduced by Giger [78] for link elements and minimum compliance problems. In the present thesis the developed complex-shaped beam element was integrated into the program code and the method was extended in order to handle complex, compliant mechanism optimization problems. This implies an extended data structure, a set of modified and new genetic operators as well as new pre-evaluation routines and objective functions.
This chapter is organized as follows: Section 3.1 is concerned with the graph-based topology representation and Section 3.2 deals with the graph representation based operators. The mapping of the objective and the constraints to a single fitness value is discussed in Section 3.3. Section 3.4 presents the pre-evaluation routines that are needed to guarantee feasible designs and to avoid useless function evaluations. In Section 3.6, discrete and multi-criteria optimization sample problems serve to validate the methodology. The subsequent Section 3.7 deals with the nonlinear optimization of compliant mechanisms and structures. And finally, Sections 3.8, 3.9 and 3.10 contain extensions of the method such as two-and-a-half-dimensional, multi-material optimization as well as an enhanced mapping concept.
3.1 The graph-based topology representation

A graph is an abstract mathematical model; it is an ordered pair \((V, E)\), where \(V\) is a finite set called vertex set and \(E\) is a binary relation on \(V\) called edge set. Elements of \(V\) are called vertices, elements of \(E\) edges.

Some notations that are relevant for the study presented here are briefly explained in the following [79].

**Graph:**

- An edge of an undirected graph is an unordered pair \((u, v)\) with \(u, v \in V\). This means that \(e(u, v)\) and \(e(v, u)\) represent the same edge.

- A graph is said to be labelled, if its vertices are distinguished from one another by labels like \(v_1, v_2, .. v_n\).

- The order of the graph \(G(V, E)\) denotes the number of vertices.

- The size of the graph \(G(V, E)\) is determined by the number of edges.

**Vertex:**

- An isolated vertex is not connected to any other vertex.

- Two vertices are adjacent to each other, if there is an edge with both vertices as endpoints.

**Edge:**

- Two edges are parallel if they have the same endpoints.

- An edge is called a loop if the two endpoints are the same.

**Connected component:**

- A path is a sequence of vertices where each vertex is connected by an edge to the subsequent vertex in the path.

- A vertex is reachable from another vertex, if a path exists from one to the other vertex.
A connected component is a group of vertices in an undirected graph that are all reachable from one another.

In this work, an undirected, labelled graph topology representation with a predefined order and a variable size is applied. Loops and parallel edges are not allowed.

3.1.1 Graph topology representation

The topology is represented by a graph, wherein each vertex corresponds to a node and each beam is described by an edge (Fig. 3.2). In other words, instead of holding the information of the beam structure in a conventional one-dimensional genotype, where the genotype is represented by a series of parameters, the graph itself is considered as the genotype and special operators are directly applied on it.

![Graph representation](image)

Figure 3.2: Beam representation (left) and the corresponding graph representation (right)

Publications in the field of compliant mechanism optimization often rely on the beam ground structure approach with a fixed number of nodes and beams [33, 36, 80]. To each beam a weight factor is assigned describing its cross-sectional dimension. A zero weight factor of an element means that there is no structural connection between the respective element nodes. The present work is assuming a fixed number of vertices, which can either be connected or not, and in contrast to other publications, a variable number of edges. The graph representation is thereby very useful, since it overcomes the ordinary restriction of
having a predefined number of elements. The length of the graph-based genotype is varying depending on that number of elements.

We find several advantages of graph-based representation. The variable length simplifies the modification of the topology by adding or removing edges and thus, it increases the design flexibility of the optimization. Furthermore, the geometry, which is mainly defined by the coordinates of the moveable nodes, can be easily modified. The nodes themselves are represented by labelled vertices and their related coordinates. The edges refer to the labels of the vertices (see Table 3.1 and Table 3.2). We only have to change the properties of the vertices, if we want to modify the geometry, and thereby, the edge parameters remain unchanged.

In addition, the graph theory provides a few very useful and fast routines such as the connected-component algorithm that is based on the depth-first search algorithm. It identifies all vertices, which are members of a certain component. A component is a group of vertices that are linked by edges. This algorithm is very efficient for determining whether all clamped or loaded vertices are connected to each other. For further details about the connected-component algorithm, the reader is referred to Siek [81].
According to the Sections 2.2 and 2.4 seven parameters and two nodes are needed to define one curved, variable thickness beam. The parameters are attached to edges as properties and the coordinates of the nodes to the vertices, respectively.

Table 3.1, Table 3.2 and their values correspond to the example in Figure 3.2; Table 3.1 contains the edges and their attached properties, namely two labels of the vertices, two angles, four thicknesses, and the material. And Table 3.2 lists the vertices and their properties like label of vertices and $x$- and $y$-coordinates. In addition to these properties, two other parameters are introduced:

- A boolean *moveable* defines whether node coordinates may change or not.

- an unsigned number attached to the edge elements describes the *status* of the edge/beam.

Three settings of the status are possible:

0 **Endpoints and shape are fixed**

1 **Only endpoints are fixed**

2 **Endpoints and shape can be modified**

Consequently, edges with status 0 or 1 must not be removed or added from/to the graph and from/to the structure during the optimization.

### 3.1.2 The universal gene concept

In Section 3.1 the graph representation with element properties was introduced. In this section the graph representation is extended by combining the graph representation with the universal gene concept.

The basic idea of the universal gene concept [56] is to locally hold not only the information, like thickness or coordinate, but also its bounds and mutation parameters of the optimization in each gene. The universal gene concept provides a variety of gene types such as *boolean-gene* and *string-list-gene* that can be combined to a heterogeneous list of properties. The gene types differs in the type of parameters they
3.1 The graph-based topology representation

<table>
<thead>
<tr>
<th></th>
<th>current value</th>
<th>upper limit</th>
<th>cyclic properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>int_gene</td>
<td>true</td>
<td>true</td>
<td>llower true lupper ε σ false</td>
</tr>
<tr>
<td>double_gene</td>
<td>true</td>
<td>true</td>
<td>llower true lupper ε σ false</td>
</tr>
<tr>
<td></td>
<td>lower limit</td>
<td>mutation parameter</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.3: Example for int_gene and double_gene

represent. For example a double-gene stands for a thickness or a coordinate, a int-gene for a material number and a bool-gene for an applied boundary, respectively. In the frame of this thesis int-genes and double-genes are integrated into the program code. Figure 3.3 displays these two types of genes and their parameters. The first boolean denotes whether the gene is active with current value \( v_{current} \). The boolean of the following two pairs specifies whether the lower \( l_{lower} \) or upper limits \( l_{upper} \), respectively, are activated. Furthermore, the mutation parameters \( \varepsilon \) and \( \sigma \) are specified. \( \varepsilon \) denotes the range for uniform mutation, if the lower and upper limit are not declared (unbounded gene), and \( \sigma \) defines the standard deviation to be used for Gaussian mutation. The Gaussian probability distribution is defined as

\[
    f(v) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{v - \mu}{\sigma}\right)^2\right),
\]

(3.1)

whereas \( f(v) \) measures the probability, that mutations to \( v \) occurs, if \( \mu \) is equal to the current value \( v_{current} \). The last boolean indicates whether the respective gene has or does not have cyclic properties. Figure 3.4 visualizes the influence of the gene parameters.

This implementation scheme allows to individually adapt the gene and its bounds to the optimization problem. In addition, the procedure to check whether the parameters are out of range is very easy to handle due to the localized information.

In Figure 3.5 a sample genotype containing two lists of genes is given. The first list represents the vertices and their properties and the second list the edges and their properties, respectively.
Figure 3.4: Influence of the mutation parameters $\varepsilon$, which denotes the range of the uniform mutation of a unbounded gene, the standard deviation $\sigma$ of the Gaussian mutation $f(v)$, the bounds $l_{\text{lower}}$ and $l_{\text{upper}}$ as well as the cyclic properties option.

Figure 3.5: Sample variable length graph genotype
3.2 Graph representation based operators

All information necessary to the evolutionary operators are contained in the genes. The operators are applied to the individuals within a population with a predefined probability, whereas an individual denotes one possible solution of an optimization problem and the population itself represents a set of solutions of one generation.

Two classes of operators are distinguished: Mutation operators change the values of the parameters with a defined probability independently of other individuals of the generation. In contrast to this, crossover operators recombine the properties of two selected parent individuals in the hope of generating even better offsprings.

Mutation and crossover operators can be further classified in three types of mutation or crossover operators;

- geometric,
- topology,
- or sizing operators.

The geometry usually depends on the position of the connected nodes. Geometric operators affect the coordinates of these nodes. The geometrical representation of this thesis constitutes a special case, since the curvature of the variable thickness beam also affects the geometry of the structure. The curvature of a beam element is defined by the position of the nodes as well as by the out angles of the centerline.

While topology operators connect nodes with beam elements or remove the beam connection, sizing operators only modifies the properties of the beam elements such as the thickness parameters.

3.2.1 Complex-shaped beam element operators

In the following, only operators especially introduced for the curved, variable-thickness beam are described. For more details on the general operators, we refer to Giger [78].

To increase the efficiency of the topology optimization, operators, who split one curved, variable thickness beam in two beams and who merge two beams, were implemented. The splitting operator looks for a non-connected node and places it within the beam element. Then, it
Graph-based optimization

Figure 3.6: Split and merge operators; a) split, b) original and c) merged beam structures

replaces the original beam element by two others. In order not to end up in many unfeasible beams, a minimal beam length is prescribed. This restriction prevents too small beam elements and a too drastic increase of the number of elements during optimization. Furthermore, topology operators are integrated who search for connected nodes and remove or add complex-shaped beam elements according to a predefined probability.

Sizing operators for conventional beams only modify one property, i.e. one cross-sectional area. But the curved, variable thickness beam includes four thickness parameters. We implement two types of sizing operators. The first ones modify a single thickness value and the second ones change all of them within one computational process. Most of the optimization problems include either a minimization of the mass or a mass restriction within the fitness function. If the operator increases both the functionality and the mass of the structure, both effects may work against each other influencing the fitness of the solution. The
3.3 Fitness formulation

All objective and constraint values computed during optimization are mapped to an addend $D_i$ of the fitness function. The idea [56] of the mapping is to apply functions, which scale the objective and the constraints to the interval $[0,1]$, to avoid that one of these fitness terms becomes much larger than the other ones and therefore dominant. Accordingly, the final fitness function is defined as a weighted sum of the mapped values:

$$S = \sum_i w_i D_i$$  \hspace{1cm} (3.2)

where $D_i$ represents the rating of an objective value or a specified constraint and $w_i$ is the corresponding weight.

implemented sizing operators modify the shape of the beam, thus the functionality of the structure, under the constraint that the mass of the beam, respectively the mass influenced addend of the fitness function, remains constant. In other words, these operators improve the fitness by increasing functionality and not by the mass balance.
We distinguish between objective and constraint mapping functions. The first one is a measure of the objective of the design optimization problem and the second one penalizes violated constraints.

The design objective mapping function is defined as:

\[ D_i(v) = (a v + b)^\alpha \]  

where \( v \) is the objective value, the choice of the exponential factor \( \alpha = 5 \) is based on experience, and \( a \) and \( b \) are scaling factors defined by conditions:

\[
\begin{align*}
D_i(v = v_{\text{init}}) &= 1 \\
D_i(v = v_{\text{estim}}) &= 0.1
\end{align*}
\]

where \( v_{\text{init}} \) represents an estimated objective value of the initial designs and \( v_{\text{estim}} \) is the estimated goal value of the optimization.

Figure 3.8 shows the mapping of the objective value to fitness values \( D_i \). The constraint mapping function is defined as:

\[ D_i(v_c) = \frac{1}{1 + \exp(-\lambda (v_c - v_{c\text{-limit}} - \Delta))} \]

where \( v_c \) is the value, which has to fulfill our constraint \( v_{c\text{-limit}} \), \( \lambda \) and \( \Delta \) are scaling factors defined by conditions:

\[
\begin{align*}
D_i(v_c = v_{c\text{-limit}}) &= 0.01 \\
D_i(v_c = v_{c\text{-limit}} + v_{c\text{-tol}}) &= 0.5
\end{align*}
\]

where \( v_{c\text{-tol}} \) is a tolerance value which defines the steepness of the constraining function as shown in Figure 3.9.
3.3 Fitness formulation

Figure 3.8: Mapping of the objective value

Figure 3.9: Mapping of the constraint value
3.4 Pre-evaluation routines

The pre-evaluation routines guarantee that only legal design solutions are evaluated. Under illegal design solutions we understand that solutions can not be correctly evaluated by the FE analysis or are not complying with our restrictions. Therefore, we distinguish two types of pre-evaluation routines.

The first one regards the FE analysis and checks whether all parts of the domain boundary, where non-zero forces and prescribed displacements are described, are connected so that the mechanism is realized and/or the stiffness matrix does not remain singular. In addition, it checks whether all beam elements are sufficiently long with respect to their thickness to comply with the beam theory assumptions. If an illegal design solution is detected, the solution is set unfeasible. Since the FE analysis evaluates only feasible design solutions, a predefined, large fitness value is assigned to unfeasible solutions, in order to minimize their chances to survive during the evolving process.

The second type of pre-evaluation routines checks restrictions given by the user and does not affect the FE analysis. For example, we do not want that beams are crossing each other or that the restriction of the maximum or minimum number of elements is violated. The check for crossing of beams includes either:

a) search routine for crossing centerlines or
b) search routine for overlapping zones

Routine a) only checks the crossing of the centerline of each beam with the centerline of other beams (Fig. 3.10). Since overlapping almost always occurs if two or more beams are connected by the same node, the routine b) divides the beams into restricted and non-restricted overlapping zones as shown in Figure 3.11. While overlapping of restricted zones is punished by the optimization, the overlapping of non-restricted zones, which are located close to the nodes, is tolerated.

Figure 3.10 visualizes the differences between the routines using the same example. In contrast to the crossing centerlines routine the overlapping zones routine punishes this configuration.

Avoiding overlapping zones contributes to finding manufacturable designs. Nevertheless, it narrows the solution space and makes it more
3.4 Pre-evaluation routines

Figure 3.10: Non-crossing centerlines (left) and overlapping of restricted zones (right)

Figure 3.11: Beams divided into restricted and non-restricted overlapping zones

difficult to find a suitable topology for the mechanism which decreases the convergence. By experience it seems most efficient to use the crossing centerline routine for finding the best topology and to use the overlapping routine for refinements.
Contrary to checks regarding FE analysis, there are two possible consequences of detecting crossing or overlapping:

- The first one sets the solution unfeasible according to the FE analysis checks.
- And the second one counts the number of crossing or overlapping of the structures. The number is then mapped to a fitness addend of the solution. Thus, a large number of crossing or overlapping is penalized stronger than a low number.

The second approach has been proven to be more efficient in terms of convergence and diversity than simply setting the solution unfeasible, since it evaluated the number of violations and thus, it increases the chance that offsprings of unfeasible solutions turn into feasible solutions.

Nevertheless, checks for crossing or overlapping are exhausting in terms of computing effort, because no fast analytical evaluation is possible. For each combination of two elements the contours of both elements are numerically sampled in parallel. If we consider the layout displayed in Figure 3.12, which consists of thirteen elements, a total of $12 + 11 + 10 + 9... = 78$ checks has to be performed independently,

![Figure 3.12: Sample layout consisting of thirteen elements](image)
3.4 Pre-evaluation routines

whether the elements are located in the neighborhood or far away from each other. For this reason, we introduce bounding boxes. Bounding boxes cover a larger area than their assigned elements and their analytical formulation helps efficiently excluding combinations of elements, that are not located in their neighborhood. Two elements are not in their neighborhood, if no overlapping of their bounding boxes are detected.

![Bounding boxes](image)

Figure 3.13: a) circular bonding box, b) bonding box consisting of two circle segments

Figure 3.13 displays two versions of bounding boxes; a) circular bonding box and b) bonding box consisting of two circle segments.

3.4.1 Circular bounding box

The circular bounding box routine calculates the coordinates $x_c, y_c$ of the center point and the radius $r$ of the i-th circular bounding box in function of the coordinates $x_1, y_1, x_2, y_2$ of the start- and endnodes, and the maximum thickness $t_{max}$ of the i-th beam element.

The i-th and j-th beam elements are in the neighborhood, if the sum of the radius of two bounding boxes are larger than the distance between their center points:

$$r_i + r_j > \sqrt{(x_{ci} - x_{cj})^2 + (y_{ci} - y_{cj})^2} \quad (3.7)$$
In order to illustrate the benefit of the circular bounding boxes, the example in Figure 3.12 is considered. As mentioned before, the displayed structure is composed of 13 elements, which leads to 78 combinations of two elements. The circular bounding box routine detects 38 combinations of elements that are in the neighborhood of each other (Figure 3.14). In this sample case the circular bounding box reduces the number of expensive checks by 51.3%.

3.4.2 Circle segments bounding box

The circle segments bounding box consists of two circle segments, where \( x_s, y_s \) are the coordinates of the center point, \( r \) the radius and \( \alpha_{\text{start}}, \alpha_{\text{end}} \) are start- and end-angles of the circle segment. The circle segments bounding box incorporates additional information such as the angles, \( \beta_1 \) and \( \beta_2 \), of the centerline at start- and endnodes. These angles define the curvature of the centerline as shown in Equation 2.11. The incorporation of this information results in an enhanced approximation of the beam element.

As shown in Figure 3.15 two elements are in their neighborhood, if one or more intersection points \( P(x_p, y_p) \) of the circle segments are
3.4 Pre-evaluation routines

detected. At least one intersection point of two circle segments exists, if

$$r_1 + r_2 > \sqrt{(x_{s1} - x_{s2})^2 + (y_{s1} - y_{s1})^2}$$  \hspace{1cm} (3.8)

and

$$\alpha_{\text{start}-1} < \alpha_{p1} < \alpha_{\text{end}-1}$$
$$\alpha_{\text{start}-2} < \alpha_{p2} < \alpha_{\text{end}-2}$$  \hspace{1cm} (3.9)

is fulfilled, defining $\alpha_{\text{start}-i}$ to be smaller than $\alpha_{\text{end}-i}$, e.g. $\alpha_{\text{start}-i} = 371^\circ$ is set to $-9^\circ$.

The coordinates $x_p, y_p$ are determined by

$$x_p = \frac{a}{d}(x_{2s} - x_{1s}) \mp \frac{h}{d}(y_{2s} - y_{1s})$$
$$y_p = \frac{a}{d}(y_{2s} - y_{1s}) \pm \frac{h}{d}(x_{2s} - x_{1s})$$  \hspace{1cm} (3.10)

whereas $a, b, d$ and $h$ are equal to

$$a = \frac{r_1^2 - r_2^2 + d^2}{2d}$$
$$b = \frac{r_2^2 - r_1^2 + d^2}{2d}$$
$$d = \sqrt{(x_{2s} - x_{1s})^2 + (y_{2s} - y_{1s})^2}$$
$$h = \sqrt{(r_1)^2 + (a^2)}$$  \hspace{1cm} (3.11)
This extended bounding box formulation generally yields a large reduction of number of checks. Regarding the same example as in Figure 3.12, only twelve checks are still necessary, which is a reduction by 84.6% (Figure 3.16).

Table 3.3 summarizes the resulting numbers of checks and their related reductions when applying the respective bounding box formulation.

Table 3.3: Number of checks for overlapping

<table>
<thead>
<tr>
<th>bounding box</th>
<th>( n^\circ ) of combinations</th>
<th>( n^\circ ) of checks</th>
<th>reduction [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>78</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>circular</td>
<td>78</td>
<td>38</td>
<td>51.3%</td>
</tr>
<tr>
<td>circle segments</td>
<td>78</td>
<td>12</td>
<td>84.6%</td>
</tr>
</tbody>
</table>
3.5 Implementation

The code is written in C++ and includes four powerful libraries: the Evolving Objects (EO)\(^1\), Boost Graph Library (BGL)\(^2\), eoUniGene library [56], and the FELyX\(^3\) FE Code. Figure 3.17 illustrates the optimization loop from the initialization to the solution.

![Diagram of optimization loop](image)

Figure 3.17: The graph-based optimization incorporating three libraries (BGL, EO and eoUnigene) as well as the FELyX FE codes

\(^{1}\)http://eodev.sourceforge.net

\(^{2}\)http://boost.org

\(^{3}\)http://felyx.sourceforge.net
The EO library is providing the basic functionality for the evolutionary optimization and for the controlling of the actual program run. This includes the initialization of the different objects, ensuring the whole data flow between different populations, calling evaluation modules, and managing storage of any kind of data. It also provides the implementations for selection and replacement strategy. The selection identifies parent individuals for the next generation. The replacement chooses the individuals, which will be part of the next generation. Both mechanisms are based on the fitness criteria. The EO functionalities are independent of the optimization problem and the used data structures.

Problem specific data, such as the design domain, the boundary conditions, the fitness formulation, the predefined elements (vertices, edges, or artificial springs) and the edge prototype are summarized in text files. The edge prototype defines the bounds of each variable parameter. The initialization of the first generation is performed either randomly or based on existing designs. The required probabilities for each operator are listed in an additional control file. These operators are need for reproduction that generates new individuals of the next generation.

All data about the structure are stored in the graph-based representation using the BGL and the eoUnigene library. The BGL provides the graph-based functionalities and the eoUnigene the different types of genes, such as \textit{int\_gene} or \textit{double\_gene} (Section 3.1.2). After generating individuals, the pre-evaluation routines perform a \textit{feasibility check}. Having ensured the feasibility, the graph-based representations of the individuals are mapped to FEMs. The FE computation, which is done by our in-house developed tool FELyX, allowing for fast \textit{mapping and evaluation}, or by the commercial software ANSYS [82], gives the structural information like the deformation or the mass back. Depending on these data, the fitness of the individuals is evaluated. Any complex fitness function composed of weighted objective, constraints, and/or target value optimization can be utilized for determination of the fitness of each individual.

For more information about the basic mechanisms of an optimization loop, the reader is referred to Section 1.2.6.
3.6 Validation

In the following we present two academic optimization problems, namely the classical inverter and the parallel gripper problem. The results of an optimization using complex-shaped beam elements are compared with an optimization using conventional straight beam elements. Criteria for the comparison are the output energies of the resulting design. First, the classical inverter problem setup is described. Later, the parallel gripper optimization problem is added.

3.6.1 Setup of the inverter optimization

For the inverter problem the following setup of the evolutionary optimization shown in Figure 3.18 is considered.

![Inverter optimization setup](image)

Figure 3.18: Inverter optimization setup

**Domain:**
The optimization domain is represented by a square area. The problem is geometrically symmetric, which reduces the design domain $\Omega$ by a half.
Boundary:
The displacements and the rotation of the node in the upper left corner are blocked. Nodes on the symmetric line can only move in the $x$-direction and their rotations are suppressed, too.

Springs:
At the input and output port a predefined set of springs is added.

According to Rahmatalla and Swan [14] the structural input/output stiffness tends to mimic the stiffness of the artificial spring $k_0/k_1$. This can be partially explained by considering the energy/work equation:

$$ W_{in} = W^{spring}_{in} + W^{trans}_{in}, \quad (3.12) $$

where $W^{spring}_{in}$ is the energy stored in the input spring and $W^{trans}_{in}$ the energy transferred to the mechanism. $W^{trans}_{in}$ can be divided into:

$$ W^{trans}_{in} = W_{out} + W_{stored}^{mechanism}, \quad (3.13) $$

where $W_{out}$ is equal the energy stored $W^{spring}_{out}$ in the output spring. $W_{stored}^{mechanism}$ is the energy stored in the compliant mechanism.

If we want to maximize the output work, we have to maximize first the energy $W^{trans}_{in}$ transmitted to the mechanism and then minimize the energy stored $W_{stored}^{mechanism}$ in the mechanism.

Rahmatalla and Swan [14] explain that the device stiffness at the input tends to $k_0$ using an analogy of two springs in parallel, whereas one spring represents the artificial spring and the other the structural stiffness at the input. Assuming that the added spring has a spring constant $k_s$ and the structure stiffness is represented by the spring constant $\alpha k_s$, the work $W^{trans}_{in}$ of the spring in parallel is maximized if $\alpha = 1$:

$$ W^{trans}_{in} = \frac{u_{in}^2 \alpha k_s}{2} = \frac{\alpha}{2k_s} \left( \frac{F}{1+\alpha} \right)^2, \quad (3.14) $$

where $F$ is the constant applied force and $u_{in}$ is the input displacement induced by $F$:

$$ u_{in} = \frac{F}{k_{tot}} = \frac{1}{\frac{F}{k_s} + \frac{F}{\alpha k_s}}, \quad (3.15) $$
3.6 Validation

where $k_{tot}$ is the total input stiffness of two springs in parallel.

**Actuation:**
A constant force $F$ is applied.

**Objective:**
The inverter problem has been solved by Bendsøe [12] using the homogenization method before. The homogenization method uses an indirect way for maximizing the output work. It applies the mutual energy approach, which is a combination of the results of a dummy load case and the real load case, which allows generating derivatives for each cell. As our algorithm is independent of any gradient, we directly optimize the output work of the inverter.

Our objective is to maximize the work in the negative $x$-direction, i.e. to minimize the value of $w_{out}$ given by:

$$w_{out} = \text{sign}(u) \frac{uk_1u}{2},$$

where $u$ is the displacement and $k_1$ the spring constant at the output.

Moreover the structure must fulfil the constraint of maximum volume fraction of 30%. Greater volume fractions are penalized. The volume fraction $v$ is defined as the ratio of the area covered by the beams $A_i$ over the area of the domain $A_\Omega$.

$$v = \frac{\sum_{i=1}^{n} A_i}{A_\Omega}$$

**Evaluation**
Since displacements are small, linear analysis is carried out.

3.6.2 Three levels of complexity

Three levels of beam complexity are considered and each of them is combined with two levels of the optimization problem complexity, namely topology problems with fixed and moving nodes.

We receive a total of six different optimization setups as shown in Table 3.4. Each of them is optimized 30 times and out of those the fifteen best are taken and the averages of their results are computed, to eliminate outliers. Initial populations were set by random and, as
Table 3.4: Three levels of beam complexity and two levels of optimization problem complexity

<table>
<thead>
<tr>
<th>beam complexity</th>
<th>fixed nodes</th>
<th>moving nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>straight beam with constant thickness</td>
<td>1 a)</td>
<td>2 a)</td>
</tr>
<tr>
<td>straight beam with variable thickness</td>
<td>1 b)</td>
<td>2 b)</td>
</tr>
<tr>
<td>curved beam with variable thickness</td>
<td>1 c)</td>
<td>2 c)</td>
</tr>
</tbody>
</table>

Stop criteria the number of generations ($n_{noimpr} = 5000$) with no improvement and the maximal number of generations ($n_{max} = 200000$) are defined.

### 3.6.3 Results of the inverter optimization

The optimization is not as simple as it appears. After initialization, the optimization usually tends to minimize the compliance of the first solutions (see Eq. 3.16). During the initialization process only feasible solutions survive. Solutions are feasible, if they connect all nodes with boundary conditions or loads. In order to fulfill this request, the first solutions often consist of many elements, which block any mechanisms. Generally, in a later step, the compliant mechanism is established.

As mentioned in the beginning, we intend to simultaneously increase the functionality and minimize the number of elements. First, the best solutions employing three elements are considered. Three is the minimal number of beams that can describe a semi-symmetric inverter.

Figure 3.19 displays the best results for the fixed and moving nodes configuration. Considering the fixed nodes configuration, the performance improves by 22.0% using straight, variable thickness beams, by 35.8% using curved, variable thickness beams over that of achieved by straight conventional beams (see Table 3.5). Surprisingly, the variable thickness distribution exerts an even greater influence than the moving nodes ($0.0050\ Nmm$ compared to $0.0047\ Nmm$). But of course, this is depending on the predefined positions of the nodes. Furthermore, the curved centerlines improve the results only if fixed nodes are specified. It is assumed that they partially compensate the geometrical limitation.
Figure 3.19: Best solution employing three beams for fixed and moving nodes; a) conventional, b) variable-thickness, c) curved, variable-thickness

Table 3.5: Inverter optimization; best individuals consisting of three beams

<table>
<thead>
<tr>
<th>fixed nodes</th>
<th>best work [Nmm]</th>
<th>[%]</th>
<th>vol. fraction [ ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 a)</td>
<td>0.0041 Nmm</td>
<td>100.0%</td>
<td>0.131</td>
</tr>
<tr>
<td>1 b)</td>
<td>0.0050 Nmm</td>
<td>122.0%</td>
<td>0.299</td>
</tr>
<tr>
<td>1 c)</td>
<td>0.0056 Nmm</td>
<td>135.8%</td>
<td>0.299</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>moving nodes</th>
<th>best work [Nmm]</th>
<th>[%]</th>
<th>vol. fraction [ ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 a)</td>
<td>0.0047 Nmm</td>
<td>100.0%</td>
<td>0.123</td>
</tr>
<tr>
<td>2 b)</td>
<td>0.0058 Nmm</td>
<td>124.6%</td>
<td>0.299</td>
</tr>
<tr>
<td>2 c)</td>
<td>0.0058 Nmm</td>
<td>124.6%</td>
<td>0.299</td>
</tr>
</tbody>
</table>
caused by the fixation of the nodes (see Figure 3.19c).

Generally, the volume fraction of the conventional beam solution is even much lower than the allowed maximum amount of material. The two effects, namely that less material increases the flexibility and more material decreases the loss of elastic energy, work here stronger against each other. By contrast, all solutions using a variable thickness distribution tend to exploit the maximum non-penalized amount of material.

The results employing more than three beams confirm the improved efficiency of the variable thickness elements (Table 3.6 and 3.7), although the improvements are smaller compared to the previous configuration. The average work increases by 16.0% and 27.1% using a complex-shaped beam and fixed nodes, respectively. Employing moving nodes, the increases are 16.6% and 17.0% over that of the straight conventional beam.

In spite of the above-mentioned increased efficiency the cubic thickness function approach is not optimal as shown in Figure 3.20c. The
3.6 Validation

Table 3.6: Inverter optimization; best and average of the fifteen best individuals using fixed nodes

<table>
<thead>
<tr>
<th>fixed nodes</th>
<th>best work [Nmm]</th>
<th>[%]</th>
<th>vol. fraction [ ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 a)</td>
<td>0.0044 Nmm</td>
<td>100.0%</td>
<td>0.232</td>
</tr>
<tr>
<td>1 b)</td>
<td>0.0051 Nmm</td>
<td>116.2%</td>
<td>0.299</td>
</tr>
<tr>
<td>1 c)</td>
<td>0.0056 Nmm</td>
<td>128.4%</td>
<td>0.299</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fixed nodes</th>
<th>average work [Nmm]</th>
<th>[%]</th>
<th>stddev</th>
<th>vol. fraction [ ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 a)</td>
<td>0.0043 Nmm</td>
<td>100.0%</td>
<td>1.9%</td>
<td>0.248</td>
</tr>
<tr>
<td>1 b)</td>
<td>0.0050 Nmm</td>
<td>116.0%</td>
<td>0.6%</td>
<td>0.299</td>
</tr>
<tr>
<td>1 c)</td>
<td>0.0055 Nmm</td>
<td>127.1%</td>
<td>1.8%</td>
<td>0.299</td>
</tr>
</tbody>
</table>

Table 3.7: Inverter optimization; best and average of the fifteen best individuals using moving nodes

<table>
<thead>
<tr>
<th>moving nodes</th>
<th>best work [Nmm]</th>
<th>[%]</th>
<th>vol. fraction [ ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 a)</td>
<td>0.0051 Nmm</td>
<td>100.0%</td>
<td>0.298</td>
</tr>
<tr>
<td>2 b)</td>
<td>0.0059 Nmm</td>
<td>116.0%</td>
<td>0.299</td>
</tr>
<tr>
<td>2 c)</td>
<td>0.0059 Nmm</td>
<td>116.4%</td>
<td>0.299</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>moving nodes</th>
<th>average work [Nmm]</th>
<th>[%]</th>
<th>stddev</th>
<th>vol. fraction [ ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 a)</td>
<td>0.0050 Nmm</td>
<td>100.0%</td>
<td>1.8%</td>
<td>0.297</td>
</tr>
<tr>
<td>2 b)</td>
<td>0.0058 Nmm</td>
<td>116.6%</td>
<td>0.5%</td>
<td>0.299</td>
</tr>
<tr>
<td>2 c)</td>
<td>0.0059 Nmm</td>
<td>117.0%</td>
<td>0.8%</td>
<td>0.299</td>
</tr>
</tbody>
</table>

evolutionary algorithms have generated quasi "bubble" beams, which seem to be the best compromise between an elastic hinge at one end and a very stiff section at the other end using as little material as possible. The observed funny shapes are due to the limited solution space of the cubic polynomial thickness parametrization.
3.6.4 Influence of the springs on the results

Springs are used to influence the resistance of the structure. The geometry as well as the sizing depend on the springs. Generally, stiffer springs lead to stiffer mechanisms. Fig 3.21 illustrates the dependence.

![Figure 3.21: Solution dependent on spring factor, a) $f = 0.01$, b) $f = 1$, c) $f = 10$, and on spring ratio, d) $k_{in} : k_{out} = 1 : 8$, e) $1 : 1$, f) $8 : 1$](image)

The initial set of spring constants is multiplied by the spring factor $f$, $k_{new,i} = f k_i$, to achieve a new set of spring constants.

Not only the stiffness of the optimal solution is influenced by the springs, but also the geometry of the inverter depends on the spring constants. The ratio of the magnitude of the spring constants defines the optimal $x$-coordinate of the node connected by three elements. If the ratio is one, the optimal $x$-coordinate is approximately in the middle of the input and output. If one spring is stiffer than the other, the node moves towards the weaker spring (see Figure 3.21).
3.6.5 Result of parallel gripper optimization

A second compliant mechanism optimization problem, the parallel gripper, is studied. Three different configurations using moving nodes are optimized:

a) Using conventional, straight beams
b) Using curved beams with a quadratic thickness function
c) Using curved beams with a cubic thickness function

As the optimization setup is very similar to the inverter problem, the detailed description of the optimization setup becomes redundant.

The major differences is the prescription of the straight beam at the output (see Figure 3.22) and the fitness formulation. The goal of the optimization is maximum parallel displacements of $u_1$ and $u_2$. For this reason we optimize the double of the lower energy of either the first or the second output spring:

$$w_{out} = \text{sign}(u)uk_1u,$$

where $k_1$ the spring constant at the output and $u = \min(u_1, u_2)$. 

![Gripper optimization setup](image-url)
Because the algorithm always maximizes the lower one of output energies and seeks to exploit the whole amount of transmitted energy, the optimization leads to a result where the output energies at both springs are the same and where parallel displacements are achieved. The results of the optimized grippers displayed in the Figures 3.23, 3.24 and 3.25 confirm this fitness formulation. Their displacements differ by less than 0.002% at the outputs.

We find that the average improvement of using a complex-shaped beam for the parallel gripper is less than for the inverter problem; 4.1% for the curved, quadratic thickness function beam, and 8.1% for the curved, cubic thickness function beam approach in comparison to the straight beam optimization (Table 3.8). The lower efficiency increase compared to the inverter problems may be explained by the fact that the demand for compliant and stiff zones within one beam element is less strong. Nevertheless, the geometries found by solving the curved, variable thickness beam gripper problem are very close to manufacturable designs, which can be regarded as an additional efficiency increase.

<table>
<thead>
<tr>
<th>moving nodes</th>
<th>best</th>
<th>average</th>
</tr>
</thead>
<tbody>
<tr>
<td>beam type</td>
<td>work [Nmm]</td>
<td>[%]</td>
</tr>
<tr>
<td>a)</td>
<td>0.0159 Nmm</td>
<td>100.0%</td>
</tr>
<tr>
<td>b)</td>
<td>0.0171 Nmm</td>
<td>107.8%</td>
</tr>
<tr>
<td>c)</td>
<td>0.0173 Nmm</td>
<td>108.9%</td>
</tr>
</tbody>
</table>
Figure 3.23: Gripper optimization using conventional beams

Figure 3.24: Gripper optimization using curved beams with a quadratic thickness function,
Figure 3.25: Gripper optimization using curved beams with a cubic thickness function

3.7 Nonlinear optimization

So far, we considered compliant mechanism devices, which undergo only small displacements. The parallel grippers illustrated in Figure 3.23, 3.24 and 3.25 represent small deflection devices. More or less, they transfer the input force to the output, whereas the output displacements are less than 0.1\text{mm}. Because the displacements are small compared to the length $l = 140\text{mm}$ of the gripper, linear modelling was legitimated.

Nevertheless, this thesis additionally focuses structures, that undergo large displacement providing the desired functionality. Several authors [19, 83, 84] have presented comparisons of structures and compliant mechanisms that are generated by linear and nonlinear elastic analysis in topology optimization, which showed significantly different design solutions. Two types of nonlinearity are distinguished:

- Geometrical nonlinearity
- Material nonlinearity
The second type of nonlinearity, the material nonlinearity, is not addressed within this thesis. Nonlinear stress-strain relationships of nonlinear material like plastic, multi-linear elastic, or hyperelastic materials can cause a change of structure’s stiffness during the analysis. As long as strains remain small, it can often be neglected.

### 3.7.1 Geometric nonlinearity

When we consider geometrically nonlinear behavior in this thesis, the strains are assumed to be small, but displacements and rotations are large. An example is a long, slender fishing rod as shown in Figure 3.26. When it bends due to the fish, each segment may strain slightly, but the total deformation may be large.

![Figure 3.26: Geometric nonlinearity](image)

The geometrical nonlinear behavior of a structure occurs as stiffness and loads become a function of displacement or deformation [85]. It is not possible to solve \( u \) immediately, since the nonlinear relation between load \( f \) and displacement \( u \) is not known in advance. Therefore, an iterative process is needed to obtain \( u^n \) so that internal load \( f_{int}^n(u^n) \) corresponding to the element stresses is equal to applied load \( f^n \).
3.7.2 Geometrical nonlinear optimization

Because our in-house tool FELyX does not provide an iterative, nonlinear solver, we implemented an interface between the graph-based optimization and the commercial software ANSYS [82]. The interface (Fig. 3.27) writes and reads text files, and is able to start ANSYS in the batch mode. The input text files contain all necessary information such as nodes, elements, material parameters, loads and boundaries. The FE mesh, including the elements and nodes, was previously created by a self-written routine in FELyX or optionally by the meshing tool Triangle. This increases the controllability of the meshing and reduces unfeasible, meshed structures. The results are written in output files, which are again read by the interface. The interface assigns the displacement to the nodes of the FELyX model. That following, the evaluation of the finite element results is performed.

ANSYS employs the commonly used Newton approach to solve nonlinear problems. Although it has been proven to be efficient for many nonlinear problems, the Newton method alone is not capable to handle nonlinear buckling analysis in which the structure either collapses completely or "snaps through" to another stable configuration. For such situations, an alternative iteration scheme such as the arc-length method [82] has to be activated.

Figure 3.27: Linear and nonlinear analysis
The Newton method subdivides the load into a series of load increments as shown in Figure 3.28. The load increments $\Delta f = f^n - f^{n-1}$ can be applied over several steps. Before each solution step, the Newton method evaluates out-of-balance load vector $r^n_i$, which is the difference between the restoring forces (the loads corresponding to the element stresses) and the applied loads $f_a$. Then, the program calculates the tangential matrix $K^n_{Ti}$ and performs a linearized incremental solution, using the out-of-balance loads, and checks for convergence. If convergence criteria are not satisfied, the out-of-balance load vector is reevaluated, the tangential matrix $K^n_{Ti+1}$ is updated, and a new step solution is obtained. This iterative procedure continues until the problem converges and all load increments are applied.

Figure 3.29 displays two large deflection grippers. Both are optimized for a different size of workpieces using geometrically nonlinear modelling. Corresponding to these gripper, Table 3.9 underlines the fact, that geometrical nonlinear effects must not be neglected. The deviation of the linear solution is about 17% for nodes A and B, and about 26% for C and D, respectively.
Figure 3.29: Large deflection grippers
3.8 Two-and-a-half-dimensional optimization

Two-and-a-half-dimensional optimization was motivated by the fact that crossing beam elements are not realizable in a planar structure, because they would interpenetrate. The key idea is that the elements are arranged in different layers above each other as shown in Figure 3.30. The layers are mirrored, in order to ensure planar mechanisms. This extension of the original optimization increases the solution space, which may yield an improvement in performance of the structures depending on the optimization problem.

An additional parameter, the *layer number*, is added to the graph representation. The layer parameter is represented by an integer number with three possible states:

- Beams with layer number 0 or 1 are placed in the first layer and
- Beams with layer number 2 in the second, respectively.

### Table 3.9: Gripper optimization; geometrical linear and nonlinear analysis

<table>
<thead>
<tr>
<th>displacement</th>
<th>x [mm]</th>
<th>y [mm]</th>
<th>$\varphi$ [rad]</th>
<th>deviation [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>node A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nonlinear</td>
<td>-3.14</td>
<td>-4.41</td>
<td>0.0009</td>
<td></td>
</tr>
<tr>
<td>linear</td>
<td>-2.46</td>
<td>-3.86</td>
<td>0.0030</td>
<td>16.2%</td>
</tr>
<tr>
<td>node B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nonlinear</td>
<td>-3.13</td>
<td>-4.40</td>
<td>-0.0067</td>
<td></td>
</tr>
<tr>
<td>linear</td>
<td>-2.45</td>
<td>-3.73</td>
<td>-0.0028</td>
<td>17.7%</td>
</tr>
<tr>
<td>node C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nonlinear</td>
<td>-23.49</td>
<td>-22.71</td>
<td>-0.4839</td>
<td></td>
</tr>
<tr>
<td>linear</td>
<td>-15.51</td>
<td>-25.81</td>
<td>-0.4770</td>
<td>26.2%</td>
</tr>
<tr>
<td>node D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nonlinear</td>
<td>-18.78</td>
<td>-42.82</td>
<td>-0.4786</td>
<td></td>
</tr>
<tr>
<td>linear</td>
<td>-6.32</td>
<td>-44.21</td>
<td>-0.4700</td>
<td>26.8%</td>
</tr>
</tbody>
</table>
In contrast to elements within the same layer, which are not allowed to cross each other, beams with different layer numbers are admitted to cross each other. The only exception are elements with the layer number 0. These elements are manually set elements, which generally define a outer contour of a structure. We do not tolerate any crossing of these elements.

We can either predefine the layer number or allow the optimization to randomly mutate the layer parameters. Examples of a single- and a multi-layer inverse gripper are given in Figure 3.31 and Figure 3.32. The multi-layer gripper consists of a total of 21 beam elements, whereas 16 are placed in the first layer and 5 in the second. This structural representation yields an improvement in efficiency by 12.3% over the single-layer solution; the output work of the single-layer gripper is \( w_{\text{out}} = 0.0120 \text{Nmm} \) and the output work of the multi-layer gripper \( w_{\text{out}} = 0.0135 \text{Nmm} \), respectively. Hence, the multi-layer gripper stores less elastic energy within the structure in order to provide its functionality.

Of course, the multi-layer optimization uses a two-dimensional FE model, which does not take account of three-dimensional effects. The identified topology has to be checked in an additional postprocessing step. Thereby, postprocessing tools help analyzing three-dimensional behavior and possibility modifying the multi-layer structure. Figure 3.33 shows the three-dimensional CAD model corresponding to the multi-layer gripper.
3.8 Two-and-a-half-dimensional optimization

Figure 3.31: Single-layer inverse gripper

Figure 3.32: Multi-layer inverse gripper
Table 3.10: Inverse gripper optimization; single layer versus multiple layers

<table>
<thead>
<tr>
<th>layer type</th>
<th>work [Nmm]</th>
<th>[%]</th>
<th>vol. fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 3.31</td>
<td>0.0120 Nmm</td>
<td>100.0%</td>
<td>0.199</td>
</tr>
<tr>
<td>Figure 3.32</td>
<td>0.0135 Nmm</td>
<td>112.3%</td>
<td>0.199</td>
</tr>
</tbody>
</table>

### 3.9 Multi-material solutions

An efficient compliant structure generally consists of compliant and stiff sections. The compliant sections provide the functionality or the flexibility and the stiff sections the necessary resistance. Since two reverse characteristics are requested, multi-material solutions suggest themselves. The combination of different materials with different properties such as Young’s modulus or maximum tolerable strain may improve the structural result.
3.9 Multi-material solutions

We identify three kinds of material combinations as shown in Figure 3.34:

a) Use of a particle reinforced and homogeneous material

b) Combination of two homogeneous materials

c) Reinforcing the homogeneous material by affixing of fiber straps.

Although the reinforcement with fiber straps seems to be a viable alternative, since, from the point of view of the manufacturing, the application of the strap reinforcement is very easy to accomplish, we concentrate on combination a) and b) in the following.

<table>
<thead>
<tr>
<th>material</th>
<th>Young’s modulus $E [\text{MPa}]$</th>
<th>Poisson ratio $\nu []$</th>
<th>Density $\rho [\text{kg/dm}^3]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>steel</td>
<td>210000</td>
<td>0.30</td>
<td>7.80</td>
</tr>
<tr>
<td>aluminum</td>
<td>69000</td>
<td>0.33</td>
<td>2.70</td>
</tr>
<tr>
<td>polyamide 12</td>
<td>1700</td>
<td>0.42</td>
<td>0.91</td>
</tr>
<tr>
<td>polyamide 12 (particle)</td>
<td>3200</td>
<td>0.42</td>
<td>1.25</td>
</tr>
</tbody>
</table>

Table 3.11: Material properties
An example of an inverse, particle reinforced gripper is given in Figure 3.35. The colored areas represent pure Polyamid 12 (PA12), which offers a greater tolerable strain than particle reinforced PA12 (white). In contrast, the reinforced PA12 provides a higher Young’s modulus.

In order to answer the question, whether all kind of material combinations contributes to an improved solution, a cantilever beam is considered (Figure 3.36). The length $L$ of the beam as well as the displacement $u_0$ at the free end is predefined. We let now vary the thickness $t$ from 1 to 10 and compute the bending stiffness $D(t)$, the mass $m(t) = \rho Ltb$, and maximum strain $\varepsilon_{\text{max}}(t)$ for different materials such as steel, aluminum and PA 12 (Table 3.11).
The bending stiffness $D$ is defined as

$$D = EI = E\frac{bl^3}{12}, \quad (3.19)$$

where $E$ is the Young’s modulus, $I$ the cross-sectional inertia, $b$ the width, which is set to 1. The maximum strain $\epsilon_{max}$ is equal to

$$\epsilon_{max} = \frac{M}{EI} t = \frac{3u_0 t}{2L^2}, \quad (3.20)$$

where $M$ is the moment, that is induced by the displacement $u_0$.

Figure 3.37 illustrates the dependance of the mass on the bending stiffness. The weight of a structure is at least a constraint of a structural optimization, since it helps to eliminate material without functional purpose within design domain. This leads to the conclusion that the weight generally plays an essential role during structural optimization. If we only consider the Figure 3.37, it turns out, that for bending stiffness less than $D_{alu}(t = 10)$, aluminum is the only and best choice, since it is much lighter than steel. The combination of steel and aluminum only makes sense, if bending stiffness/compliance less and more than $D_{alu}(t = 10)$ is simultaneously required. But, if we additionally take account of the maximum strain (Figure 3.38), we realize that steel solution offers a lower maximum strain than aluminum for a given displacement and bending stiffness. It is not anymore clear, if aluminum is really the best choice. The best choice depends on the the required bending stiffness or compliance, the weight and the maximum tolerable strain. The optimal choice of materials regarding several aspects is a typical task for an optimization routine.

Another interesting observation is that aluminum offers a lower weight as well as a lower maximum strain than PA12 for a given bending stiffness, although the density of aluminum is three times higher than PA12. But, PA12 also provides a 40 times lower Young’s modulus. In order to compensate the lower Young’s modulus, much more material is needed to attain the same bending stiffness. This leads to a higher weight than for aluminum. The mass for a certain bending stiffness can be calculated by

$$m = Lb\rho \frac{\sqrt{12D}}{bE}, \quad (3.21)$$
Graph-based optimization

Figure 3.37: Mass dependance on the bending stiffness

Figure 3.38: Strain dependance on the bending stiffness for a given displacement $u_0$
Manufacturing and assembly technologies of multi-material solutions, for instance multi-material molding [86], are not addressed in this work. But, we are convinced that the constraints of these technologies have to be integrated into future optimizations, which will significantly influence the results. Figure 3.39 illustrates one possible solution. The components of the multi-material gripper are plugged on each other.

Figure 3.39: CAD model

\section*{3.10 Enhanced mapping concept}

The enhanced mapping concept is the last extension of the original method within this thesis. It is motivated by eliminating the overlapping regions and by improving the mapping of the structural properties. Since it yields a better mapping of the real-world behavior of a structure, it minimizes the loss of transferring the structural result into a real-world structure and thus, it increases its final performance. Moreover, the postprocessing steps are minimized or redundant.

The enhanced mapping concept subdivides the beam geometries into two halves which are represented by a set of polygons. In contrast to the plane-elements approach (Section 2.6), whereas the geometry is meshed edge-wise, the enhanced mapping routine picks the vertices of
the graph-based representation one after the other. It determines how many edges are connected by the vertex. If a single edge is connected with the vertex, half the beam geometry next to the considered vertex is meshed according to the plane-elements approach. If more than one edge is connected by the same vertex, the enhanced mapping is applied (Figure 3.40). The enhanced mapping routine is divided into five steps:

**Step 1: Determination of circle intersection points.** Circles are introduced in order to determine appropriate intersection points. Starting with an initial value, the radius grows as long as the lines between the corresponding intersection points do not cross with lines of other corresponding intersection points anymore and the minimal distance between the intersection points is more than $d$. The predefined minimal distance $d$ reduces notch effects and improves the mesh quality by preventing too many small triangular elements.

**Step 2: Deleting of the region close to the node/vertex.** The polygon points and lines are deleted within the circle from the list of polygons.

**Step 3: Connecting intersection points.** The previously determined intersection points are now connected according to the position on the introduced circle.

**Step 4: Meshing of the contour.** The subsequent meshing of the polygon contour is provided by the mesh generator Triangle [77]. Thereby, the triangular elements are transformed into six nodal elements, in order to prevent bending blocking effects.

**Step 5: Unifying of the structure.** After meshing the halves of the beam geometries, the last step unifies the corresponding halves by merging the adjoining nodes.
3.10 Enhanced mapping concept

Figure 3.40: Enhanced mapping procedure
An example of an enhanced mapped structure is given in Figure 3.41. The colors illustrate the vertex-wise meshing. The FE analysis results in 5% less displacements at the outputs compared to plane-elements approach. The increased stiffness is caused by the reduced free length of the beam geometries.

During the optimization the robustness of the mapping routine is crucial. The mapping procedure shall be able to map all possible and impossible variations of the structure without generating FE models, which can not be evaluated. Thereby, both the beam-theory and plane-elements approach have been proven to be very efficient. In order to ensure the robustness of the enhanced mapping method, a switch was implemented. If no appropriate intersection points are found, e.g. in case of very large overlapping, the plane-elements method is applied instead of the enhanced mapping method. Then, the routine counts the number of regions that are meshed according to the plane-elements approach and the structure is punished by the fitness addend depending on that number. The punishment is needed in order to force the optimization towards solutions, which are entirely mapped according to the enhanced mapping concept. Figure 3.42 displays a sample structure. Three of thirteen regions are failed to mesh according to the enhanced mapping concept.
Finally, we add an example of an optimization using the enhanced mapping concept. Figure 3.43 visualizes the result of a force inverter optimization. We export the result into a CAD model, mesh again the structure in a Finite Element tool and calculate the displacements. We find that the deviation between the optimization result and the final CAD model is less than 0.5% (Table 3.12), which is tolerable. By comparison, similar solutions of using the conventional meshing method (Section 2.6) showed a deviation greater than 2.0%.

Certainly, the enhanced mapping increases the accuracy of the optimization, but it also increases both the complexity and the efforts, and slows down an optimization run. During the first stages of an optimization the accuracy is less important than finding an appropriate topology. For this reason, it is most efficient to use the enhanced mapping for refinement of topologies obtained by conventional optimization runs, in order to increase the final accuracy of the results.
Figure 3.43: Force inverter optimization; a) complex-shaped beam representation, b) FEM of the optimization, c) Finely meshed CAD model

Table 3.12: Enhanced mapping; force inverter

<table>
<thead>
<tr>
<th>version</th>
<th>$F_{out}[N]$</th>
<th>deviation [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimization</td>
<td>1.1695 N</td>
<td></td>
</tr>
<tr>
<td>roughly meshed CAD model</td>
<td>1.1748 N</td>
<td>0.46%</td>
</tr>
<tr>
<td>medium-size meshed CAD model</td>
<td>1.1691 N</td>
<td>0.03%</td>
</tr>
<tr>
<td>finely meshed CAD model</td>
<td>1.1686 N</td>
<td>0.07%</td>
</tr>
</tbody>
</table>
Chapter 4

Postprocessing of the graph-based optimization

The whole synthesis procedure of compliant mechanisms/structures can be typically divided into four steps [87]:

1) *Problem definition* which specifies goals of the development including the objective and all constraints.

2) *Design domain parametrization* which contains all necessary information for the optimization. It specifies the parametrization of the design domain, of the coordinates of the nodes as well as of the properties of the beam elements. The positions of some nodes or some beam elements and their properties may be predefined or proposed.

3) *Structural optimization* which identifies compliant mechanisms by evolutionary techniques in combination with the ground structure approach and the complex-shaped beam element. Since thousands of iterations have to be performed until good results are obtained, the optimization is the most time consuming step of the development procedure.

4) *Postprocessing* of the data obtained by the optimization.

This section addresses the last step of the development procedure, the postprocessing of the FE models obtained by the optimization.
Our postprocessing includes:

- the visualization of the results obtained by the optimization including their structural behaviors

- the check of secondary constraints which were neglected in order not to slow down the optimization

- the analysis of the structural behavior in terms of variation of the loads

- the transfer of the optimized structures into a manufacturable design

A detailed visualization of the results is very helpful, especially in the beginning of an optimization run. The definition of the fitness formulation for a new optimization is very challenging, since the whole structural behavior of the solutions and their properties are mapped to a single fitness value. It is not uncommon, that the solutions exploit undesired effects in order to maximize the objective or minimize the fitness value, respectively. The visualization helps to understand the structural behavior and thus, to avoid these undesired effects by adapting the design problem or the fitness formulation. An example is the structural performance displayed in Figure 4.1. Although the displace-

![Figure 4.1: Visualization of the structural behavior](image)

ments of the straight beam endnodes are parallel, a undesirably strong
bending deformation of the straight beam occurs. An additional incorporation of the difference between the rotations at the beam endnodes into the fitness formulation may prevent such undesired solutions.

In Section 4.1 a short introduction into the graph-based design platform is given and in Section 4.2 the final design interpretation of a parallel gripper is addressed.

4.1 Graph-based design environment

The purpose of the graph-based design environment displayed in Figure 4.2 is to enable an efficient pre- and postprocessing of the optimization. Thereby, the graph-based design environment provides a high flexibility and compatibility between the EoGraph optimization and the different tools such as FELyX [88], ANSYS [82], CATIA [89] as well as the EoGraph pre- and postprocessing tools.

Figure 4.2: Graph-based design environment
These tools help setting up new optimizations (Appendix C), visualizing and analyzing the optimized results and they allow a fast final design interpretation of the encoded data obtained by the optimization. The so-called final design interpretation step is necessary, since the outcomes of the optimization are encoded according to the graph representation, and not directly CAD (Computer Aided Design) models. The final design interpretation provides the mapping of the encoded data into CAD models.

Two types of result files of the optimization are distinguished:

- The \textit{generationsX.sav} file represents the final population, which includes all genotypes of the \textit{Xth} generation.

- The \textit{modelX.matlab} file contains the best phenotype of the \textit{Xth}-generation including all nodes, elements, loads, boundary conditions, artificial springs and material properties.

While the \textit{generationsX.sav} file serves to restart the optimization from an already existing population, the result file \textit{modelX.matlab} can be read by the EoGraph pre- and postprocessing tool as well as by FELyX for further development. FELyX is responsible for the mapping of the graph-based representation into a FE model and provides linear analysis. Furthermore, it enables us to test new implemented routines before integrating them into the optimization program code. Both FELyX and the EoGraph postprocessing can access ANSYS providing linear and nonlinear analysis. In addition, in the EoGraph postprocessing tool an interface is implemented, that helps transferring the encoded data into a CAD model. Later on, the CAD model again can be imported into ANSYS for further analysis or provides the basis for the manufacturing of a compliant system.

4.2 Final design interpretation of a parallel gripper

In this section we address the final design interpretation of compliant grippers. The major challenge is to transfer the structural information into a manufacturable design without changing the optimized properties. The use of complex-shaped beam finite elements enhances the
4.2 Final design interpretation of a parallel gripper

Evolutionary optimization of a two-dimensional compliant-mechanism design, but leaves the problem that overlapping of geometries may occur in regions close to nodes [90]. This problem can be only overcome by the enhanced mapping concept (Section 3.10). In the following, solutions with complex-shaped beam elements are considered and the effect of overlapping of the beam elements at the nodes is studied. Two CAD based versions for the final design interpretation are presented and their accuracies of the resulting final designs are compared with the solution of the optimization. The first one is user-assisted and the second one establishes the CAD model automatically.

In the course of this work, the interpretation of the parallel gripper shown in Figure 4.3 as an example is considered. Compliant grippers are common design problems and variations of them have been broadly studied in the literature [32, 37, 87, 91, 92, 93, 94, 95].

Figure 4.3: Optimized gripper
4.2.1 User-assisted CAD based final design interpretation

The CAD based procedure can be described as follows; a MATLAB routine generates a data file including the structural information of the optimized structure, which can be directly loaded into the CAD program CATIA V16 by a Visual Basic (VB) script. This script generates a set of nodes which describes the contour of the 2-dimensional structure. The designer can now manually connect these nodes using splines. Because of the same nature of the splines and of the parametrization of complex-shaped beam structures the CAD models match very good with the optimized structures even if less contour nodes are employed. The user-assisted routine allows to modify the structures by shifting

![Figure 4.4: Contour nodes connected by splines](image)

or skipping nodes as shown in Figure 4.4. The incorporation of the designer’s knowledge seems to be advantageous due to multiple interpretation possibilities of the transition zone and due to the limited solution space of the cubic polynomial thickness parametrization. Figure 4.5 displays the manually generated CAD model of the parallel gripper.

4.2.2 Automated CAD based final design interpretation

In contrast to the user-assisted final design interpretation, the second method generates the CAD model fully automatically. This version
4.2 Final design interpretation of a parallel gripper

Figure 4.5: Manually generated CAD model of the parallel gripper

Figure 4.6: Automated CAD model of the parallel gripper
Postprocessing of the graph-based optimization

uses the same approach as the manual version and connects the contour nodes of each beam using splines. The greatest challenge is to overcome the gap between the beams as shown in Figure 4.7. The routine adds chamfers at the ends of the beams which overlap with each other and close the undesired gap. And, an additional logical operation unites the beams. The added chamfer can be manipulated by the tension factor $T$; $T=0.01$ for an approximate straight line and $T=2.0$ for an approximate semi-cycle. By experience the use of the tension factor $T=1.0$ has often given good results. Figure 4.6 presents the automated CAD Model.

### 4.2.3 Discussions

As mentioned before, the main challenge is to transfer the optimized structure into a manufacturable design without loosing the structural properties. As properties the optimized energy at the output and the difference of the displacements, which describes the parallelism of the gripper output, are considered. For comparison, both values are normalized; the output energy by the output energy of the optimized structure and the difference of displacements by the average value of the displacements. Table 4.1 summarizes these results of the two versions using geometrically linear and nonlinear modelling, respectively. The

![Figure 4.7: Gap between two beams and added chamfer](image.png)
difference between linear and nonlinear modelling is very small and therefore, optimization using linear computation is legitimated.

The losses of output energies due to the final design interpretation are 3.3% and 3.1%, respectively, and the differences of the displacements are less than 1.2%. These losses can be explained by the increased stiffness next to the nodes and the reduced free length of the continuous structure compared to the beam representation.

In the following, we are trying to minimize the overlapping zones and thus, to increase the accuracy of the later final design interpretation. We use the routine for the detecting of overlapping zones according to Section 3.4 and add an additional constraint, which takes account of the number of overlaps and minimizes it. But each additional constraint limits the solution space. The extended optimization results in 1.8% lower output energy over the output energy of the previous optimization. While the accuracy of the automated CAD based final design interpretation is slightly better (97.2% see Table 4.2), the final output energy of the design is lower (97.2% * 98.2% = 95.5%) than without using the routine for overlapping zones. An improved design in terms of the output energy could not be achieved.

We arrive at the conclusion that the final design interpretation process reduces the objective values obtained by the beam model. An output energy loss of about 3% is incurred considering the present

Table 4.1: Comparison of the versions; using geometrically linear and nonlinear modelling

<table>
<thead>
<tr>
<th>version</th>
<th>modelling</th>
<th>$w_{out}$ [mm]</th>
<th>$u_1$ [mm]</th>
<th>$u_2$ [mm]</th>
<th>diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimization</td>
<td>linear</td>
<td>100.0%</td>
<td>-0.08195</td>
<td>-0.08195</td>
<td>0.0%</td>
</tr>
<tr>
<td>user-assisted CAD</td>
<td>linear</td>
<td>96.7%</td>
<td>-0.08125</td>
<td>-0.08061</td>
<td>0.8%</td>
</tr>
<tr>
<td>user-assisted CAD</td>
<td>nonlinear</td>
<td>96.8%</td>
<td>-0.08118</td>
<td>-0.08065</td>
<td>0.7%</td>
</tr>
<tr>
<td>automated CAD</td>
<td>linear</td>
<td>96.9%</td>
<td>-0.08068</td>
<td>-0.08158</td>
<td>1.1%</td>
</tr>
<tr>
<td>automated CAD</td>
<td>nonlinear</td>
<td>96.8%</td>
<td>-0.08062</td>
<td>-0.08162</td>
<td>1.2%</td>
</tr>
</tbody>
</table>
Postprocessing of the graph-based optimization

compliant gripper. While the automated CAD based version allows a very fast final interpretation, the manual CAD based version offers a greater flexibility and incorporates the user’s knowledge. Avoiding overlapping zones contributes to finding manufacturable designs. But, it narrows the solution space and makes it more difficult to find a suitable topology for the mechanism.

In Figure 4.9 the final prototype of the compliant gripper is displayed.

Table 4.2: Parallel gripper with limited overlap

<table>
<thead>
<tr>
<th>version</th>
<th>modelling</th>
<th>$w_{out}$</th>
<th>$u_1$</th>
<th>$u_2$</th>
<th>diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimization</td>
<td>linear</td>
<td>100.0%</td>
<td>-0.08122</td>
<td>-0.08139</td>
<td>0.2%</td>
</tr>
<tr>
<td>automated CAD</td>
<td>linear</td>
<td>97.2%</td>
<td>-0.08009</td>
<td>-0.08101</td>
<td>1.1%</td>
</tr>
</tbody>
</table>

Figure 4.8: Parallel gripper with limited overlaps
Figure 4.9: Rapid prototyped parallel gripper
Chapter 5

Application: Adaptive car seat concept

Mainly micro- and nanotechnologies as well as precision engineering have discovered the benefits that compliant mechanisms offer over rigid link mechanisms [3, 96, 97, 98]. The miniaturization of a conventional hinge mechanism is very difficult or not possible to realize. In contrast, compliant mechanisms can be manufactured as single-piece systems severely facilitating the fabrication in micro scale. Furthermore, they involve no wear, backlash, noise and lubrication, which increase the precision of the devices [99, 100, 101, 102]. For these reasons compliant mechanism solutions become very attractive, especially in the field of MEMS (Micro-Electro-Mechanical System). MEMS are submillimeter mechanical systems coupled with electronic circuits. Most of the MEMS applications of compliant mechanisms involve mechanically rather simple single-input/single-output (SISO) mechanisms such as force inverters, motion amplifiers, force amplifiers or gripping mechanisms [16, 20, 103, 104, 105, 106, 107, 108, 109].

Middle- and large scale applications of compliant mechanisms are not as numerous as the micro applications. Kota et al. [110] highlight some of the benefits of engineered elasticity in the design of surgical instruments for laparoscopic and robot-assisted surgery. Trease [111] developed a concept for a biomimetic compliant wing that mimics the motion of the lateral fin of a fish. And the possibilities of shape morphing compliant structures on the basis of an antenna reflector was
demonstrated by Lu and Kota [112].

Shape control of adaptive wing systems is currently an important and active research area in the field of compliant structures. Several smart wing concepts incorporating compliancy have been proposed and developed: Saggere and Kota [113] addressed the leading edge of an adaptive wing and in a later work Kota et al. [114] present a demonstrator wing model with adaptive leading and trailing edges that has been tested in a wind tunnel. The DARPA/AFRL/NASA Smart Wing Program also investigates a smart wing for military aircraft seeking to increase the flight performance [115]. Campanile [116, 117, 118, 119] developed a belt-rib concept for variable-camber airfoils. The belt-rib structural frame consists in a closed shell (belt) reinforced by in-plane stiffeners (spokes) allowing a predefined deformation mode. Maute and Reich [120] presented an optimization approach for the design of adaptive wings that merges topology optimization for mechanism layouts with design optimization of coupled fluidstructure problems. And finally, Diaconua et al. [121] introduced a bi-stable morphing airfoil concept.

While the concept of a smart wing is promising, it is also, however, showing some serious limitations such as reliability or scalability of laboratory-scale prototype models into real-scale models. Nevertheless, the smart wing remains an attractive and exciting research topic.

In the following we address an adaptive car seat as a new example of a potential, large scale application of compliant systems.

5.1 Design objectives

The car seat is the largest contact surface between the driver and the car, making it one of the most important factors in comfort feeling of the passenger. Seat comfort is heavily influencing how relaxed, and therefore secure, the driver arrives at his target location. But the car seat and especially the backrest have to fit a wide variety of people, and thus, it needs some sort of adaptability which is mostly provided by padding materials. Padding materials are not mechanically stiff enough to support the driver in situations with high lateral acceleration, and an adaptation of the material properties to different driving conditions is very difficult or not possible to obtain. Some vehicles use air cushions
5.2 Adaptive car seat concept

Figure 5.1: The backrest clasping around the driver’s back (top view) in order to partly compensate the lateral acceleration during cornering.

The presented car seat concept shall open up the possibility to simultaneously increase the functionalities and decrease the complexity (number of elements) of a car seat construction. In particular, it is motivated by the following targets:

- Adaptability to the driver (see Figure 5.1)
- Adaptability to driving conditions
- Minimal number of parts

5.2 Adaptive car seat concept

To achieve our targets we propose a car seat concept based on compliant ribs that are stacked together to a full backrest as shown in Figure 5.2. Columns, which are fixed to the bottom seat structures, keep the compliant ribs in position, and adapters, which connect the rib-like structures with the columns, serve to modify the S-Curve along the columns imitating the human spine. The compliant ribs are designed to adapt to the driver by embracing the back and to provide a lateral support, e.g. during cornering.

However, the design criteria, adaptation and support of the human body, are an apparent contradiction. The compliancy, which is a precondition for the adaptation, and the stiffness request for the
support work against each other. In order to solve this problem, two types of rib-like structure are distinguished; active and passive rib-like structures. The active structures are combined with an actuation system, which provides an additional energy source. The more energy is available to deform a structure, the stiffer is the structural result of the optimization. In addition, the variation of the loads caused by the inertia of human body affects the structure less:

$$W_{def} = W_{act} + W_{m} + \Delta W_{m}$$

where $W_{def}$ is the elastic energy stored in the structure neglecting any plastic deformation, $W_{act}$ is the energy introduced by the actuation system, $W_{m}$ is the input energy as a result of the human weight and $\Delta W_{m}$ its variation depending on the different driving situations such as lateral acceleration. The increased energy input $W_{in}$, which is equal to $W_{def}$, leads to stiffer structures, since the ratio of $\Delta W_{m}$ to $W_{in}$ becomes smaller using an additional input energy, and $\Delta W_{m}$ affects the structures much less.

Moreover, the combination of an actuation system with sensors and a control unit opens up the possibility not only to minimize the lateral deflection, but also to actively increase the support by guiding the
driver against the lateral acceleration. Such a system would have to permanently adapt itself to the driving situations.

Although the whole concept is motivated by imitating the human spine, two columns which keep the rib-like structures in position seem to be more beneficial than a single column. In contrast to the center single column configuration the torque moment caused by an asymmetric load case $F_{a,m}$ can be absorbed by a pair of forces $F_{re1}$ and $F_{re2}$ as shown in Figure 5.3. This fact implies two major advantages. The bending moment, which affects the compliant structure in case of an asymmetric load case, is less and the integration of a linear actuation system simplifies the design of the columns and the choice of the actuation system.

The passive rib-like structures are stacked together according to the demands along the spine. They can differ in the structural behavior such as stiffness or twist properties. For example, an increased twist capacity of the top ribs improves the overview and therefore the security when backing up.

In the following, this work concentrates on the development of the active rib-like structures for the symmetric load case, when the driver is leaning back and manually applying the actuation system.

Figure 5.3: Center single column versus two columns (top view)
The following sections are organized according to the four steps of the development procedure:

- Step 1: Problem definition (this Section)
- Step 2: Design domain parametrization (Section 5.3)
- Step 3: Structural optimization (Section 5.4)
- Step 4: Final design interpretation (Section 5.5)

5.3 Design domain parametrization

We aim for a rib-like structure, that adapts itself to the body contour of the driver by morphing the initial contour into a target contour. Hence, we define an initial, undeformed curve representing a maximum contour and a target, deformed curve, that is given by an average European male as shown in Figure 5.4. Sampling points are used to discretize the initial contour along the curve, whereas these points are connected by beam elements, that complete the contour.

As shown in Figure 5.5 the optimization domain is represented by a square area. The problem is geometrically symmetric which reduces the
Figure 5.5: The axis-symmetric setup; initial shape (outlined), target shape (dashed) the driver’s body (gray), the forbidden area (horizontally lined), external loads $F_a$, $F_m$ and $F_r$, and boundary conditions.

design domain by a half. Within the forbidden area (horizontally lined) we do not allow any beam elements which would cross the body section of the driver. Nodes on the symmetric line ($x = 0$) are only allowed to move in $y$-direction and their rotation is blocked. The displacements and the rotations of the nodes on the line $y=0$ are suppressed too. All other nodes can move in the whole design domain and are not necessarily connected by beam elements. The loads are introduced by $F_m$, which is caused by the body force of the driver, $F_a$, which represents the actuation force and $F_r$. $F_r$ is introduced in order to increase the lateral resistance/stiffness of the structure. The displaced coordinates of the sampling points ($1 - 6$), which are a result of the introduced loads, serves to quantify the deviation of the deformed backrest to the target shape.

The fitness formulation is composed of an average displacement error in x-direction and a modified standard deviation approach in y-direction. The only use of the average displacement error in both directions as shown in Equation 5.2 and 5.3 would lead to a very demanding objective formulation, which narrows the solution space:
Δ_{x_{err}} = \frac{1}{n} \sum_{i=1}^{n} |x_{TAR,i} - x_i| \quad (5.2)

Δ_{y_{err}} = \frac{1}{n} \sum_{i=1}^{n} |y_{TAR,i} - y_i| \quad (5.3)

where \( n \) is the number of sampling points, \( x_{TAR,i}, y_{TAR,i} \) are the target coordinates and \( x_i, y_i \) are the displaced coordinates of the sampling points.

This formulation would require a close shape matching and that the two curves, target and deformed curve, lie at the same position. But in fact, a larger or smaller average deviation in y-direction is not relevant to the functionality of the car seat as long as the deformed curve is parallel to the target shape. Thus, the displacement error in y-direction is replaced by a modified standard deviation formulation (STD):

\[
\Delta_{y_{mstd}} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_{TAR,i} - (y_i - \Delta\bar{y}))^2} \quad (5.4)
\]

where \( \Delta\bar{y} \) is the average difference of the target and displaced y-coordinates of the sampling points:

\[
\Delta\bar{y} = \frac{1}{n} \sum_{i=1}^{n} (y_{TAR,i} - y_i) \quad (5.5)
\]

The standard deviation of a set of numbers is a value indicating how far the values are spread around their mean. In our case, a small \( \Delta_{y_{mstd}} \) indicates that they are clustered closely around the target shape corrected by the average difference \( \Delta\bar{y} \).

Figure 5.6 and Table 5.1 illustrate the difference between the optimization employing the fitness formulation \( \Delta_{y_{err}} \) and \( \Delta_{y_{mstd}} \). The dotted line is very close to the target shape (outlined) and thus to a low \( \Delta_{y_{err}} \).

But this optimization leads to a deformed shape which is not very parallel to the target shape anymore. Instead, \( \Delta_{y_{mstd}} \)-optimization leads to a more parallel deformed shape (dashed) resulting in a lower \( \Delta_{y_{mstd}} \) but a higher \( \Delta_{y_{err}} \) value.
5.3 Design domain parametrization

Figure 5.6: The target curve (outlined) and the difference of $\Delta y_{err}$ optimization (dotted) and $\Delta y_{mstd}$ formulation (dashed).

<table>
<thead>
<tr>
<th>Objective</th>
<th>$\Delta y_{err}$</th>
<th>$\Delta y_{mstd}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta y_{err}$-optimized (dotted)</td>
<td>0.115</td>
<td>0.138</td>
</tr>
<tr>
<td>$\Delta y_{mstd}$-optimized (dashed)</td>
<td>0.326</td>
<td>0.034</td>
</tr>
</tbody>
</table>

Table 5.1: Objective values corresponding to Figure 5.6.

Hence the objective $O(\vec{p})$ for the shape morphing problem of the compliant rib is defined as follows:

$$O(\vec{p}) = \min_{\vec{p}} (w_1 * \Delta y_{mstd} + w_2 * \Delta y_{err})$$ (5.6)

where $\vec{p} = \{p_1, p_2, \ldots, p_m\}$ are the design variables of the problem and $w_1, w_2$ are scalar weighting factors.

Additionally, the maximum average difference of the target and displaced y-coordinates (Eq. 5.5) and the maximum amount of material are limited by the incorporation of constraint functions according Section 3.3.
5.4 Results

The graph-based optimization yields a structure as illustrated in Figure 5.7. The structural layout consists of a total of ten beam elements. Stiff and compliant zones are distributed within the structures at different locations. The structures can be divided in two sections; section A tendentiously displaces in the negative and section B in the positive y-direction under load.

The objective values of the obtained structure are given in Table 5.2. The modified standard deviation $\Delta_{y_{\text{mstd}}}$ is equal to 1.157 using linear modelling, the least square error $\Delta_{x_{\text{err}}}$ = 0.962, respectively. Except sampling point 4 all sampling points reached the target displacement. The differences between target and obtained displacement of sampling points 1-3 is less than 0.7mm, of 5-6 less than 1.7mm. As we expected, the objectives increase for geometrical nonlinear modelling. In order to again improve the nonlinear performance, the subsequent postprocess step, which is CAD based, is applied.
### 5.5 Final design

The final design interpretation step transfers the encoded data into a CAD model and adapts the model to its geometrical nonlinear behavior.

<table>
<thead>
<tr>
<th></th>
<th>linear</th>
<th>nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta y_{mstd}$</td>
<td>1.157</td>
<td>2.176</td>
</tr>
<tr>
<td>$\Delta x_{err}$</td>
<td>0.962</td>
<td>1.391</td>
</tr>
</tbody>
</table>

Table 5.2: Objective function values using geometrical linear and nonlinear modelling.

#### 5.5 Final design

The following modifications have been carried out according to Figure 5.8:

1. A pocket was introduced in order to reduce the pressure on the spine.
2. Manual adjustments were applied to counteract the tendentiously stiffer CAD model.
3. Modifications are applied to reduce stress concentrations.
4. Areas without high deformations are modified for design purposes.

Figure 5.8: Manual modification of the optimized topology
(5) The curvature of the beam in the lower lateral corner was increased to enlarge the space for the actuator.

(6) A connection to the back rest support and an actuation pad are added.

In addition, we slightly move the actuation point and, since the CAD model generally tends to result in stiffer structures, we have to adapt the actuation force $F_a$. These modifications yield the structure with objective values of $\Delta_{ymstd} = 1.085$ and $\Delta_{xerr} = 1.747$ using nonlinear modelling (Table 5.3). The nonlinear performances of the final rib is visualized in Figure 5.9. The stress concentration at the transition zone (red circle) still exists, but it is not critical anymore.

![Figure 5.9: Deformation of the final rib; target shape (dashed), displaced structure (colored)](image)

<table>
<thead>
<tr>
<th>objective</th>
<th>nonlinear</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta_{ymstd}$</td>
<td>1.085</td>
</tr>
<tr>
<td>$\Delta_{xerr}$</td>
<td>1.747</td>
</tr>
</tbody>
</table>

Table 5.3: Objective function values of the modified structure using nonlinear modelling.
5.6 Functional prototype

After transferring the optimization result into a manufacturable design, we used rapid-prototype techniques for the manufacturing of a functional prototype. The prototype consists of two columns and three compliant ribs. Figure 5.10 displays the functional prototype; a) initial state, b) actuated state.

![Functional prototype](image)

Figure 5.10: Rib-like compliant mechanisms; a) initial state, b) actuated

For the optimization we have introduced the actuator force $F_a$. But we did not specify the actuator system so far. There are many different actuators like electromechanical, pneumatic or hydraulic actuators commercially available. An inelastic, tubelike air cushion shown in Figure 5.11 fulfills best our criteria, such as simplicity, low weight as well

![Inelastic, tubelike air cushion](image)

Figure 5.11: Inelastic, tubelike air cushion
as compatibility with the adaptive car seat concept. Air cushions are very light, consume minimal volume and open up the opportunity to act on several rib-like structures at the same time. Neither tools nor fastener are required for the assembly which minimizes the assembly as well as the maintenance costs. Furthermore, the compressibility of the air efficiently smooths pressure peaks out on the driver’s back. As shown in Figure 5.12 they are placed within the columns, in order to minimize the required space and facilitate the modularization of the whole system.

![Figure 5.12: Actuation; a) initial state, b) actuated](image)

Figure 5.12: Actuation; a) initial state, b) actuated

![Figure 5.13: Air cushion model](image)

Figure 5.13: Air cushion model
The pneumatic control system consists of a small compressor and two valves which regulate the pressure in the air cushions. The following, simplified model serves to quickly estimate the needed pressure $p$ in the air cushions.

Assuming that the tube of an inelastic air cushion can be represented by two semi-circle and two parallel contact lines $a(p, F_a)$, the circumference $C$ can be calculated as follows:

$$C = \pi d + 2a$$  \hspace{1cm} (5.7)

where $d(p, F_a)$ is the distance between the two parallel contact lines. The line force $F_a$ is the product of the pressure and the length of the contact line $a$ or, in other words, the contact line $a$ is equal to the division of the line force $F_a$ by the pressure $p(d, F_a)$:

$$a(p, F_a) = \frac{F_a}{p(d, F_a)}$$  \hspace{1cm} (5.8)

Since we know the circumference $C$ of the nearly inelastic membrane, the distance $d$ and the needed actuator line force $F_a$, we can

![Figure 5.14: Actuation line force $F_a$ and distance $d$ versus pressure $p$ plot](image-url)
now estimate the pressure $p(d, F_a)$ by the following equation:

$$p(d, F_a) = \frac{2F_a}{C - \pi d}$$ \hspace{1cm} (5.9)

Figure 5.14 displays the relation between actuation line force $f_a$, distance $d$ and the pressure $p$. The maximal distance $d_{\text{max}}$ between the contact lines is 28.01mm that is twice the radius of an air cushion tube.

### 5.7 Discussion

The resulting rib-like structure combines a feature of both structure and mechanism. On the one hand the rib-like structure provides the designed flexibility, and on the other hand it carries the loads and keeps the driver in position. We find several points of advantage.

- The rib-like structures are very light, since they are single-piece devices and provide their functionality without any hinges. Each hinge is a concentration of mass.

Figure 5.15: Adaptive car seat concept
5.7 Discussion

- Low manufacturing and maintenance costs are expected due to the small number of parts. The simple stacking together of the single-piece devices minimizes the assembly time.

- An enhanced, dynamic functionality can be realized. In combination with appropriate sensors, actuators and an automated control system the structure is able to adapt itself to the driving situation, e.g. enhanced side support in curvy roads (Fig. 5.16).

Of course, the adaptive car seat is still a concept and needs further development. Many real-world load cases, like crash, are not considered until now. Moreover the adaptability of structure was optimized regarding a single body size. Figure 5.17 displays the results of contact simulations regarding two other, a small and a big, torso.

Figure 5.16: Adaptation to driving situation; turning left

Figure 5.17: Nonlinear contact simulations; a) small b) big male/female
While the contact simulation a) yields a quite good result, the adaptability to the torso b) is limited (red circle). Thin padding materials could partly compensate this lack. But we speculate that an optimization regarding two torso sizes and an optimized initial shape can improve these results. Furthermore a direct nonlinear optimization improves the mapping of the structural behavior and its approximation of the structural, real-world performance. It makes the nonlinear post-processing step redundant or minimizes it.

However, the present design example of the car seat concept based on compliant rib-like structures shows that compliant mechanisms and structures may be a viable alternative in everyday’s objects. Due to the integrated functionality they can help to reduce weight, assembly time and production costs.
Chapter 6

Conclusion and outlook

6.1 Conclusion

This dissertation has investigated several aspects of the structural design synthesis in the field of compliant mechanisms and structures. A complex-shaped beam element has been introduced, and its parametrization as well as the influence of the parameter $L$ on the shape of Hermite curves have been explained. The constant stretching concept and its approximation equation is used as constraint to determine the shape parameter $L$, in order to remove redundancies and to prevent some unpleasing shape solutions. The structural behaviors are calculated according to the Castigliano’s theorem, or alternatively, the complex-shaped beam geometry is mapped into a FE mesh, which opens up the opportunity to employ commercial FE tools providing linear and nonlinear analysis.

The complex-shaped beam element has been integrated into the graph-based optimization for the design of compliant mechanisms/structures. The presented method is based on evolutionary algorithms and is combining the ground structure approach and the graph theory-based parameter representation. Pre-evaluation routines have been introduced in order to guarantee legal design solutions and thereby bounding boxes reduce the number of function evaluations. We have found several points of advantage of the method. The optimization has been proven to be robust, computationally simple yet capable in their search abilities, and can deal with discrete problems such as
multi-material or multi-layer design. The complex-shaped beam element with its thickness variation and curved centerline increases the design freedom over conventionally used, straight beams. And finally, its graph theory-based topology representation greatly enhances the flexibility because of its inherent simplicity and its ability to handle variable-length genotypes. The optimization becomes independent of a predefined number of elements. The whole methodology has been applied on classical inverter and gripper problems, and their efficiencies have been discussed too.

Furthermore, some extensions of the original method such as the two-and-a-half-dimensional and the multi-material optimization as well as an enhanced mapping concept, have been explained. The enhanced mapping concept efficiently overcomes the problem of overlapping of the geometries at the nodes. While each extension increases the quality of the design solutions as illustrated in Figure 6.1, it also increases the efforts of the optimization and slows down the convergence. By experience it has been proven to be most efficient to increase the complexity stepwise when running a new optimization.

The graph-based design environment provides a high flexibility and compatibility between the employed tools and the EoGraph optimization. The EoGraph preprocessing facilitates setting up new optimization problems and the EoGraph postprocessing helps visualizing and analyzing the results obtained by the optimization. Furthermore, it allows a fast final design interpretation of the structures by providing an interface to a commercial CAD tool. Two final design interpretation procedures have been discussed and the effect of the overlapping of the geometries at the nodes has been studied.

Finally, the development of the rib-like compliant structures for an adaptive car seat has demonstrated the capability of the above-mentioned synthesis procedure. The whole design methodology including the setting up of the optimization, the final design interpretation of the result obtained by the optimization as well as the actuation concept, has been discussed. The outcome, the adaptive car seat concept, serves as a potential application of compliant systems. The car seat adaptability is provided by the designed compliance and the modularization along the columns. In addition to the enhanced adaptability, the concept may open up the opportunity to facilitate the assembly of
6.1 Conclusion

Complex-shaped beam elements

Meshed beam elements

Enhanced mapping

Decreased complexity — Decreased efforts

linear analysis
no crossing centerlines

single layer
single material

nonlinear analysis
no overlapping elements

multi layers
multi materials

Increased complexity — Increased efforts

Figure 6.1: Increasing complexity and efforts
Conclusion and outlook

a car seat by reducing the number of parts as shown in Figure 6.2. This may yield a lighter and cost-efficient structure. A further advantage is that the compliant rib-like structure in combination with an actuation and control system make an adaptation to different driving situations possible. However, the near future will prove how big the potential of the adaptive car seat concept really is.

In spite of the above-mentioned advantages, there are also some serious limitations of the design methodology. The ground structure approach efficiently reduces the number of parameters, but it also limits the solution space. For this reason the optimization and its solutions are strongly influenced by the complex shape beam geometry and its parametrization. Though Evolutionary Algorithms are robust, they are also time-consuming and it is never guaranteed to find the global optimum. In addition, the setting up of a new optimization requires a lot of experience, since many parameters may influence or limit the optimization run.

The presented method does not compete with the continuum-based methods such as the homogenization or element connectivity parameterization methods for convex, non-discrete optimization problems. These methods combine mathematical programming solution techniques, which are much more efficient in terms of both convergence and solution space. The strength of the method developed within this thesis is the capability to find a solution for these optimization problems that can not be or can only be inadequately solved by the mathematical programming methods. The most of the real-world problems are highly constrained. For example, manufacturing or performance constraints are influencing the optimization run. As a result these kinds of optimization problems are often of discrete and non-convex nature. Methods using mathematical programming solution techniques generally identify only local minima of non-convex problems and are therefore limited. This problem can be overcome by the presented method, even though it is not guaranteed to find the global minimum. Comparing with other current ground structure methods, which often incorporate Evolutionary searching techniques to solve the optimization problem, the presented method offers an increased flexibility and solution space over them. In addition, the graph representation increases the flexibility, because we do not have to define the number of elements
and to limited the nodal position in the design space. Furthermore, the complex shape beam increases the solutions space due to the ability of having a curved centerline and a variable thickness distribution. For this reason, we are convinced that the graph-based method is a very promising approach, when considering large-scale, highly constrained compliant systems, and it opens up the opportunity for further innovations in the field of compliant systems.

Figure 6.2: Adaptive car seat concept and a conventional system
6.2 Outlook

This thesis has mainly focused the synthesis of compliant systems and examples have served to validate the methodology. But if we consider industrial applications of compliant systems some other aspects, which were not or slightly addressed within this work, have to be taken into account as Figure 6.3 illustrates. Efficient manufacturing technologies and exact predictions of the longtime structural behavior including nonlinear material properties, creep and fatigue, receive increasing significance and may influence the future optimization by constraints. In addition appropriate combinations of actuators, sensors, control units and compliant structures, that may additionally enhance the performance and competitiveness of compliant systems, should be studied too. We arrive at the conclusion that there is still an essential need for further research.

This thesis is concluded by summarizing ideas for further research in the field of compliant systems.
• **Adaption of the Evolutionary procedure.** With an increasing complexity the optimization tends towards genetic drift. Genetic drift describes the effect of a loss of population diversity that occurs due to the stochastic nature of selection in a finite population. Adaptations of operators and the fitness formulation should be studied, in order to prevent this effect and to increase the efficiency. Less and more general operators may perform better than a lot, specialized operators in terms of diversity. And the incorporation of similarity of the other solutions into the fitness formulation may guide the optimization towards an increased diversity.

• **Matching of the optimization with manufacturing technologies.** Each manufacturing technology such as drilling, extrusion or injection-molding, includes some restrictions. A matching of the optimization with manufacturing technology may enhance the real-world behavior of compliant systems. For example, a minimal radius or a maximum difference in thicknesses could be predefined and added to the optimization as constraints.

• **Optimization concerning fiber reinforced plastics.** So far, isotropic materials are considered. Nevertheless, fiber reinforced plastics provide excellent properties such as high strength and Young’s modulus in fiber direction as well as low density. It is expected that fiber reinforced plastics tend less towards creeping due to the integrated fibers. In addition, they allow a fabrication of very thin structures, which decreases the strain due to bending in the deformed state. The original thickness distribution should be replaced by the number of fiber reinforced plastic layers and failure criteria should be implemented.

• **Contact-aided compliant mechanisms** [122, 123]. Contact-aided compliant mechanisms are a new kind of compliant mechanisms, that combine elastic deformations with intermittent contacts to transmit force or motion. Compliant mechanisms
deform in the direction depending on the external loads. Intermittent contacts help limiting or guiding the deformation in certain directions. The implementation of contact-aided compliant mechanisms optimization requires an automatic formulation of contact conditions within FE evaluation.

- **Optimization that includes distributed actors** [124]. Distributed and embedded actuators in compliant systems is another, very attractive topic. It is often very difficult to define where to place the actuators within the design domain by intuition. The ability of the optimization to shift and place actuator systems within the domain may open up the possibility for further enhancement.

- **Final design interpretation optimization.** The structural results are heavily influenced and limited by the chosen thickness formulation. In a second step optimization, the results could be modified by slightly shifting the nodes on the outer contour line in order to additionally increase the performance and to minimize stress concentration.

- **Three-dimensional optimization.** This thesis has concentrated on cross-sectional problems which include two-dimensional optimizations. A most obvious, further approach is the extension to tree-dimensional problems. The representation of the structures may remain the same, but the mapping of the structures will be differently performed.

- **Fluid/structure coupling.** The coupling of structural and fluid dynamic analysis is certainly possible for aerodynamic purpose. But, it is also very challenging, since the time for a function evaluation increases and it thereby influences the speed of convergence. Thus, an efficient optimization methodology is a precondition for a fluid/structure coupling.
• **Design of crash boxes.** Finally, other topics such as the design of crash boxes could be tackled by using the same methodology. Nonlinear material behavior, failure criteria and contact conditions may be integrated into the program code, in order to exactly predict the behavior of crash boxes.

Figure 6.4: Hingeless medical gripper for minimally invasive surgery
Appendix A

Approximating equation for constant stretching concept

Closed-form equations for approximating the numerical results for the shape parameter $L$ is here derived, depending on trigonometric functions of the endpoint conditions $\beta_1$ and $\beta_2$ and according to the constant stretching concept. It appears convenient to replace, as illustrated in Figure A.1, these angles which are measured against the $x$ and $y$ directions of the cartesian reference system by their differences $\gamma_1$ and $\gamma_2$.

Figure A.1: Definition of relative angles $\gamma_1$ and $\gamma_2$
\[ \gamma_1 = \beta_1 - \bar{\beta} \]
\[ \gamma_2 = \beta_2 - \bar{\beta} \]  

(A.1)

to the direction \(\bar{\beta}\) of the straight line of length \(d\) connecting the two endpoints

\[ \bar{\beta} = \arctan \frac{\Delta y}{\Delta x}. \]  

(A.2)

We also normalize the shape parameter \(L\) with respect to the distance between the two endpoints, namely

\[ \lambda = \frac{L}{d} \]  

(A.3)

The terms appearing in the closed-form equations are constructed to deliver approximations of the \(\lambda\) values, calculated by the numerical procedures, for given values of \(\gamma_1\) and \(\gamma_2\) within the range \(-90^\circ \leq \gamma_1, \gamma_2 \leq 90^\circ\). The tabulated data obtained by the constant stretching concepts are discussed in Subsection A.1, to identify approaches for obtaining the approximating equations.

Figure A.2: Shape parameter \(\lambda\) data constant stretching concept

Figure A.2 provides values of the normalized shape parameter \(\lambda\) defined in Eq. A.3 for the range of angles \(-90^\circ \leq \gamma_1, \gamma_2 \leq 90^\circ\) in steps of 30° as illustrated in Figure A.3. The presented square data arrays possess point symmetry with respect to the center point \(\gamma_2 = \gamma_1 = 0^\circ\). It appears that the distributions along the straight lines, drawn across the data arrays given in Figure A.2, have identical shape in the two
Figure A.3: Range of $\gamma_1$ and $\gamma_2$ underlying the data in Figure A.2

directions where either $\gamma_1$ or $\gamma_2$ remain at $0^\circ$ but that other shapes appear along $\gamma_2 - \gamma_1 = 0^\circ$ and $\gamma_2 + \gamma_1 = 0^\circ$, so that the latter two directions seem to be axes of mirror symmetry.

We detect a periodicity of the data with respect to the circumferential direction indicated by the ellipses drawn on the data arrays. To generate a sequence of data points on this circle we introduce with Figure A.4 cylindrical coordinates $\alpha$ and $\rho$ which can be determined from

$$\rho = \sqrt{\gamma_1^2 + \gamma_2^2}$$

$$\alpha = \arctan \frac{\gamma_1}{\gamma_2}$$

(A.4)

but are presently used, with a given radius $\rho = 90^\circ$, to generate

$$\gamma_1 = 90^\circ \sin \alpha$$

(A.5)

$$\gamma_2 = 90^\circ \cos \alpha$$

in steps of $\Delta \alpha = 5^\circ$. We assume that the approximating functions $f^a$ be separable in the circumferential and radial directions $\alpha$ and $\rho$ so that $f^a = c(\alpha)r(\rho)$ and

$$\lambda^a = w_1c_1r_1 + w_2c_2r_2 + w_3c_3r_3 + w_4c_4r_4$$

(A.6)
We find the circumferential periodical functions $c(\alpha)$ by a trial-and-error procedure guided by intuition. The functions $c_3$ and $c_4$ are active in certain quadrants only and switched off in others as Figure A.5 illustrates. Candidate functions $c(\alpha)$ are normalized so that their maximum values are unity and fitted to a sequence of points on the circumference $\alpha$ where $\rho = 90^\circ$. The fitted functions $c(\alpha)$ are plotted in Figure A.6.

The radial functions $r(\rho)$ are also found by intuition and fitted to the data points in the radial directions. Please note that the distances along the directions $\alpha = \pm 45^\circ$ are longer than in the $\alpha = 0^\circ$ or $\alpha = 90^\circ$ directions. This is taken into account by introducing the stretching of the radius range as it depends on the angle $\alpha$. The moving along the edges of a square within the data array is conveniently described by using the variable radius $\rho^\square(\alpha)$:

$$
\rho^\square = \frac{\rho}{|\cos \alpha| + |\sin \alpha|}
$$

(A.7)

The data points shown in Figure A.7 are also spaced in steps of $\rho^\square = 5^\circ$. At each data point $i$ we expect a deviation $e_i$ of the approximation $\lambda^a_i$ from the numerically evaluated value $\lambda_i$. For a selected set of $N$ data points we estimate the quality of agreement by the mean square error

$$
\bar{e}^2 = \frac{1}{N} \sum_{i=1}^{N} (\lambda^a_i - \lambda_i)^2,
$$

(A.8)

which we minimize by adjusting the weights, and its standard deviation:

$$
s^e^2 = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} [(\lambda^a_i - \lambda_i)^2 - \bar{e}^2]^2}
$$

(A.9)
A selection of curves, including the wicket [64, 66, 65], the s-curve, and a skew curve, are used to give a visual impression of how well the curves corresponding with the results for the numerically and approximated values $\lambda$ and $\lambda^0$ agree. The curves are shown in Figure A.8. The agreement here is measured by the euclidian distance $d_i$ of the corresponding points on the curves and for these a mean distance

$$\bar{d} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{dx^2 + dy^2},$$

(A.10)

and standard deviation

$$s^d = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (d_i - \bar{d})^2}$$

(A.11)

analogous to Eq. A.8 and Eq. A.9 are calculated.

A.1 Approximation of shape parameter $\lambda$

Figure A.2 provides values of the normalized shape parameter $\lambda$ calculated by the numerical procedure solving the minimization problem connected with the constant stretching concept (Section 2.4). The circumferential distribution of the data $\lambda(\alpha, \rho = 90^\circ)$ is plotted in Figure A.6 and marked with solid circles. The solid line approaching them

![Figure A.6: Circumferential distribution of $\lambda$ for $\rho = 90^\circ$](image)
constant stretching concept

does not describe how well the data can be fitted by adjusting weights $w_i$ for the four normalized periodic circumferential shape functions $c_i$.

\begin{align*}
c_1 &= 1 \\
c_2 &= \frac{1}{2} (1 + \cos 4\alpha) \\
c_3 &= \sqrt{\left[ \frac{1}{2} (1 - \cos 4\alpha)^3 \right]} \\
c_4 &= \frac{1}{2} (1 - \cos 4\alpha)
\end{align*} \quad (A.12)

where $c_3$ and $c_4$ appear in their respective domains illustrated in Figure A.5 and vanish elsewhere.

Next we investigate the distributions along the straight lines in Figure A.2, use a finer division of $5^\circ$ for better resolution, and plot them in Figure A.7. We note the previous observation that the curves shown in there extend to the edges of the square data array, and that the

\textbf{Figure A.7: $\lambda$ along point symmetry lines}
diagonals are longer than the horizontal or vertical. The functions

\[ r_1 = 1 \]
\[ r_2 = \left(1 - \cos(2\sqrt{2} \rho □)\right) \]
\[ r_3 = \sqrt{(1 - \cos^2 \rho □)^3} \]
\[ r_4 = \left(1 - \cos \frac{1}{2} \rho □\right)^2 \]

are used to give the individual approximations

\[ \lambda^a_1 = r_1 \]
\[ \lambda^a_2 = r_1 + 0.0386r_2 \]
\[ \lambda^a_3 = 0.0386r_2 + 0.4260r_3 \]
\[ \lambda^a_4 = 0.0386r_2 + 6.9200r_4 \]

which fit the respective distributions plotted in Figure A.7 with the small errors whose measures are given in Table A.1.

<table>
<thead>
<tr>
<th>( \bar{c} ) [%]</th>
<th>( \bar{e} ) [%]</th>
<th>( \bar{\theta} ) [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0074</td>
<td>0.0007</td>
<td>0.0087</td>
</tr>
<tr>
<td>0.0009</td>
<td>0.0000</td>
<td>0.0015</td>
</tr>
<tr>
<td>0.0013</td>
<td>0.0015</td>
<td>0.0013</td>
</tr>
</tbody>
</table>

Table A.1: Approximation functions error in % constant stretching concept

The remaining task is to combine the circumferential periodicity depicted in Figure A.6 with the distributions along the straight lines shown in Figure A.7. Thus, we use the periodic shape functions \( c_i \) as carriers for the distributions Eq. A.14 and obtain the approximating
equation:
\[
\lambda^a(\alpha, \rho_{\square}) = c_1 \quad r_1 \\
+ c_2 \cdot 0.0386r_2 \\
+ c_3 \cdot (0.0386r_2 + 0.426r_3) \\
+ c_4 \cdot (0.0386r_2 + 6.920r_4)
\]  \hspace{1cm} (A.15)

The formal deviation of Eq. A.15 from Eq. A.6 is because of the circumstance that the radial function \( r_2 \) contributes to not only \( \lambda^a_2 \) but also \( \lambda^a_3 \) and \( \lambda^a_4 \).

The good agreement between the numerically obtained and the approximated values \( \lambda \) and \( \lambda^a \) lets expect the impressive similarity of curves demonstrated with Figure A.8. From it one obtains that the four shapes share the same endpoint positions and differ only by the angles specified at points \((0, 0)\) and \((0, 2)\): the more and less strongly curved \( S \) shapes correspond with \( \beta_1 = \beta_2 = -90^\circ \) and \( \beta_1 = \beta_2 = -45^\circ \), respectively, and the two curves with points in the positive \( x \) range only correspond with \( \beta_1 = 0^\circ, \beta_2 = 90^\circ \) and \( \beta_1 = 90^\circ, \beta_2 = -90^\circ \). For each shape two curves are plotted and the dotted lines indicate the numerically obtained \( \lambda \) values. The agreement with the respective solid lines is excellent and it is only in the case of the wicket shape \( \beta_1 = 0^\circ, \beta_2 = 180^\circ \) where the eye can notice that the curves do not exactly match. The visual impressions are confirmed by the statistical data listed in Table namely the mean values Eq. A.10 and standard deviations Eq. A.11 of the distances between the points of the compared curves.
### A.1 Approximation of shape parameter $\lambda$

<table>
<thead>
<tr>
<th>$\gamma_1, \gamma_2$</th>
<th>$-90, -90$</th>
<th>$-45, -45$</th>
<th>$0, 90$</th>
<th>$90, -90$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$ [%]</td>
<td>0.145</td>
<td>0.049</td>
<td>0.050</td>
<td>0.389</td>
</tr>
<tr>
<td>$s^e$ [%]</td>
<td>0.022</td>
<td>0.008</td>
<td>0.008</td>
<td>0.060</td>
</tr>
</tbody>
</table>

Table A.2: Statistical measures in % of approximation error of curves shown in Figure A.8.
Appendix B

EoGraph optimization

The EoGraph project is an environment written in C++. The program includes the necessary definitions of classes for the complex-shaped beams, their objects, material data, etc. It also contains a number of specific operators for mutation and recombination. Furthermore, the custom parts of the EA are included in the program. The executable file eoGraphOpt-seq starts the optimization run.

B.1 Data sources for the geometry

Three files exist for the definition of the genotype, two of which are needed to start an optimization run. The first one is the eoGraphOpt.Data file which contains the coded geometry information for the problem. This file is always needed. The other files are the generationsX.sav file which contains the whole genotype for a population and the eoGraphOpt.initPop file which contains a template of the genotype for the creation of a first generation. The contents of all three files are linked, meaning that a specific eoGraphOpt.Data file only works with either an appropriate eoGraphOpt.initPop file or an equivalent generationsX.sav file. For the ease of use all files can be created with the graphical user interface CompliantGraphPreproc preventing typing errors, described more thoroughly in Appendix C.
B.1.1 The eoGraph.Data file

This file contains several sections containing mainly data for the definition of the design space as well as for predefined elements, FEM boundary conditions, material data, etc. Every section includes a counter \texttt{count} defining how many entries will be read from that corresponding section. If the \texttt{count} number is smaller than the entries in the section, the extra data will be skipped. If the \texttt{count} number is larger than the number of entries, an error will occur.

In the \texttt{[eoGraphOpt]} section the design space is set. So far the program only works for 2D problems, so the \texttt{z-length} value has to be set to 0.

In the \texttt{[eoGraphOpt/Polynom/positive(negative)]} section, the rectangular design space can further be constraint by confining polynomial functions that define the upper and the lower limit for the \texttt{y-coordinates}.

\texttt{[eoGraphOpt/staticVertices]} contains the numbering, \texttt{x-}, \texttt{y-}, \texttt{z-} coordinates and geometric boundary conditions (BCs) for the vertices. For every BC a boolean defines if it is active (1) or inactive (0). Every structural problem needs at least one input (displacement, rotation, force or moment) and it also has to be at least statically determinate. The structure is the following:
\begin{verbatim}
vertex<index> = ' <index> <x-coord> <y-coord> <z-coord> ... 
disp <dx-act> <dx> <dy-act> <dy> <dz-act> <dz> ... 
rot <rx-act> <rx> <ry-act> <ry> <rz-act> <rz> ... 
load <lx-act> <lx> <ly-act> <ly> <lz-act> <lz> ... 
mom <mx-act> <mx> <my-act> <my> <mz-act> <mz>'
\end{verbatim}

\texttt{[eoGraphOpt/staticEdges]} contains the information for the complex-shaped beams. The first entry again is the beam number, the second and third entries define the vertices that define the beam endpoints. Fourth and fifth entries are the \texttt{β}-angle values. Entries 6 to 9 contain the thickness values. The 10th represents the layer number. Entry 11 defines the material number of the beam as listed below and the last entry defines whether the beam can change its thickness, \texttt{β}-angles and material or if it can even be removed. A value of 0 means that the beam cannot be changed or removed. A value of 1 means that the beam can change its thickness distribution,
β-angles or material from one generation to another and a value of 2 denotes that the beam can change thickness etc. and even be removed from one generation to another. The structure is the following: 

\[
\text{edge<index> = ' <index> <label1> <label2> <β1> <β2> ... <t1> <t2> <t3> <t4> <layer#> <mat#> <stat> '}
\]

[\text{eoGraphOpt/StaticFixNodes}] contains the coordinates of points where springs can be attached to. They do not serve as endpoints for edges. The entries have following structure: 

\[
\text{fixnode<index> = ' <index> <x-coord> <y-coord> <z-coord> '}
\]

[\text{eoGraphOpt/StaticSprings}] contains the information for springs. Springs can be added for example to represent an output resistance or to limit the input/output displacements. The first entry in the spring data is the spring number, the second entry must be a fixed vertex, the third entry is a fixed node, the fourth is the spring constant (units are \([\text{N/mm}]\)). The structure is thus the following: 

\[
\text{spring<index> = ' <index> <vertex#> <fixnode#> <spring k> '}
\]

[\text{eoGraphOpt/BorderLine}] defines lines in the design space which may not be crossed by any edge. This is useful if some area of the design space must be kept free from any edge. For example: if the task is to design a gripper mechanism, one would not want any edges in the area between the splits. The structure of the borderlines has the following structure: 

\[
\text{bline<index> = ' <index> <x-coord 1> <y-coord 1> ... <x-coord 2> <y-coord 2> '}
\]

[\text{eoGraphOpt/MaterialData}] contains a catalog of different materials. The first entry is the name of the material, the second entry is the elastic modulus \(E\) in [MPa] the third entry is the Poisson ratio \(ν\) [-] and the fourth entry is the density \(ρ\) of the material in [Ton/mm\(^3\)]. The structure is the following: 

\[
\text{mat<index> = ' <name> <E> <ν> <ρ> '}
\]

[\text{eoGraphOpt/BorderLine}] defines horizontal or vertical boundary constraint lines in the design space. If nodes move on the lines including a certain tolerance, the respective boundary constraint is applied
to them. For example: nodes on symmetric lines get the symmetric constraint. In case of loads, the forces and moments are divided by the number of nodes on the boundary line, to avoid increasing input loads. Dx, Dy, Rz are the displacements or rotation and Fx, Fy, Mz are the forces or moment in the respective directions. The preceding boolean defines, whether the subsequent value is activated or not. The structure is the following:

\[
\text{moveBC<index> = 'index' bool x-coord bool ... y-coord bool ... Dx bool Dy bool ... Rx bool ... Fx bool Fy bool ... Mz ...}
\]

### B.1.2 The eoGraphOpt.initPop file

The `eoGraphOpt.initPop` file contains the prototype of a genotype. The following data are stored:

- The first line `\section{eoPop}` is needed in the program in order to know where to start to read in the data.

- The second line contains a number describing how many individuals are listed in the section `eoPop`. In the `eoGraphOpt.initPop` file this number is always 1.

- The third line contains a flag which is always set to INVALID in the `eoGraphOpt.initPop` file. This means that the genotype must be evaluated and its fitness is undetermined.

- The fourth line tells the program how many vertices are in each genotype. Warning: This number must be equal to the number of vertices defined in the `eoGraphOpt.param` file, otherwise the program will launch an error!

- The following section in the `eoGraphOpt.initPop` file contains the genes for all the coordinates of a fixed number of vertices including their respective upper and lower limits as well as the mutation parameters \(\epsilon\) and \(\sigma\) for every coordinate. Note: the cyclic property is set inactive for all coordinates. A numeric flag at the end of every vertex-line defining whether the respective vertex is allowed to move and must be connected
B.1 Data sources for the geometry

to an edge (movable=0) or not (movable=2). Movable=1 defines that the vertex must be connected, but it is also allowed to move.

A vertex has following genotype structure:
< label >
< double gene x-coordinate > ,
< double gene y-coordinate > ,
< double gene z-coordinate > ,
< movable >

- The last section of the file contains the prototype of a single edge. The prototype is needed for the generation of new edges. It defines the limits as well as the mutation parameters. The status of new edges is always set to 2. The status describes whether an edge cannot change vertices, thickness, angles nor material (status=0), whether it can change thickness, angles and material (status=1), or whether it can change everything and even be removed (status=2).

An edge has following genotype structure:
< int gene label1 > ,
< int gene label2 > ,
< double gene beta1 > ,
< double gene beta2 > ,
< double gene t1 > ,
< double gene t2 > ,
< double gene t3 > ,
< double gene t4 > ,
< int gene layer > ,
< int gene material > ,
< status >

Important notes when using eoGraphOpt.initPop for the initialization of a population

There are some advantages and disadvantages when the eoGraphOpt.initPop file is used for initialization, as opposed to the generationsX.sav file:
• Only one edge-prototype can be used. There is, for example, no way to define some edges that can switch between materials 1 and 2, and other edges that can switch between materials 3 and 4.

• The individuals of the population will always be initialized using some random initialization. A desired full genotype cannot be given, but there is a workaround: A desired geometry can be achieved with the `eoGraphOpt.Data` file.

• Because all edges in a population have the same genotype prototype the optimization program can use all the operators that work on the edges. This includes the split operator and the join operator.

B.1.3 The `generationsX.sav` file

The `generationsX.sav` file contains similar information to the `eoGraphOpt.initPop` file. The difference is that it contains not only genes of a single prototype individual, but it contains the genes of a whole population. Therefore the second line can contain a number different from 1. Another difference is that multiple prototypes for the edges can be used.

Important notes when using `generationsX.sav` for the initialization of a population

Some important points have to be respected when a population is initialized with a `eoGraphOpt.startGen` file.

• It is possible to generate a desired start geometry that can be used in the optimization program without prior random initialization.

• The number of generated start individuals is not crucial. If a start population is needing more individuals, then the extra individuals can be generated through initialization. If a start population needs less individuals then only the desired subset will be used.

• Because of the aforementioned reason it is thus possible to generate a subset with a desired geometry (genotype) plus another
subset which is randomly initialized. Initialization can always be performed with a linear distribution throughout the given limits of every gene, or it can be performed with a Gauss-distribution around the given geometry (genotype).

- Instead of the \texttt{generationsX.sav} file, any \texttt{.sav} file from previous optimization runs can be used for the initialization.

\subsection*{B.1.4 Data source for the definition of the fitness terms}

The fitness terms are defined in the file \texttt{eoGraphOpt.fitTerms}. This file basically contains the information how a value, which is desired to be minimized, is mapped onto the total fitness of every individual. There are several ways to map values, such as weight, deflection, energy etc. onto a fitness. Every single fitness term is weighted and the sum of the weighted single fitness terms finally defines the total fitness. The total fitness has always to be minimized. For values which are desired to be maximized it is either possible to take the negative value or the inverse value.

Any geometry or FEM value, such as weight or deflection can be subjected to side constraints. A side constraint could be for example a maximum weight or a maximum number of edges. Another example could be a desired deflection in one point.

For a genetic algorithm it would be inefficient to discard all the solutions that violate one or more side constraints. Even if an individual violates one or more side constraints it can contain the almost perfect solution. In the next generation the offspring of the individual will be slightly mutated and it might be a very good solution to the problem not violating any side constraints. Simply giving a very high fitness value to those individuals violating one or more side constraints is therefore not a good idea. It is much better to generate "soft" boundaries, one could say a "gray zone". If an individual has a value within a gray zone, its fitness will be penalized on the severity of the side constraint violation. For more information about the mapping of the objective and constraints referred to König [56]

It is only mentioned here that \texttt{eoGraphOpt.fitTerms} files can be viewed with the postprocessing MATLAB tool \textit{EoGraph postprocessing} (Appendix D).
B.2 Data Source for the running parameters of an optimization

The parameters for an optimization run are in the file CompliantGraph-seq.param. This parameter file is divided in several sections:

**General section**

The parameter `--seed` can be activated if an exact copy of subsequent runs is desired. A seed value controls the generation of random numbers. If the seed is set active, the machine will always generate the same random numbers throughout a run. This is usually helpful when debugging the program.

**Evolution Engine section**

This section contains information on the population size, the selection process, the number of offspring and the replacement process.

`--selection` defines the selection process. Possible values are:

**DetTour;** the deterministic tournament selection process takes $T$ individuals randomly out of the population. The best of these $T$ individuals is selected as fertile parent.

**StochTour;** the stochastic tournament selection process is equivalent to the deterministic tournament but it takes the $t$'th part of a population into the tournament.

**Roulette;** the roulette tournament is a special kind of selection process. The domain $[0, 1]$ is divided into $N$ unequally sized parts, where $N$ is the number of individuals in a population. The sizes of the parts are relative to the fitness values of the individuals. The parts are then split between all the individuals whereas the fittest individual gets the greatest part and the least fit individual gets the smallest part of the domain. A random number between 0 and 1 is then generated. The individual who's part contains that random number is selected as fertile parent. The fittest individual has thus the largest probability to be chosen as parent.
**Ranking:** the ranking selection process sorts the population according to their fitness. The fittest individual gets the highest probability to be chosen as parent.

**Sequential:** the sequential selection process chooses the first $M$ individuals in a list. $M$ is the number of offspring defined in `--nbOffspring`. The list can be unordered or ordered regarding the fitness of the individuals.

`--replacement` defines the replacement process. Possible values are:

**Comma:** the comma replacement process replaces the parent individuals by the offspring individuals. If `--nbOffspring` is equal to 100% or equal to `--popSize` then all the parents are replaced.

**Plus:** the Plus replacement process chooses the $N$ fittest individuals from the parent and the offspring generation. $N$ is again the number of individuals in the population. This form of replacement is called **elitism** because the best individual survives an unlimited number of generations until another individual reaches a better fitness.

**EPTour:** the Evolutionary Programming Tournament is a stochastic tournament replacement scheme. Each individual (among the $2N$ parents plus offspring) encounters $T$ random opponents, increasing its score by one point if it has a better fitness. The $N$ individuals having the highest scores are passed to the next generation. The EP replacement scheme is always elitist. [56].

**SSGAWorst, SSGADet, SSGAStoch:** the Steady State Genetic Algorithms replacement processes replace the whole parent population with the offspring.

`--weakElitism` can be activated when the best parent individual shall replace the worst offspring individual. This mechanism ensures convergence for the best fitness value, even when Comma or any of the SSGA replacement schemes are set.
Fitness Evaluation section

This section contains the paths to the \texttt{eoGraphOpt.Data} and the \texttt{eoGraphOpt.fitTerms} files. The parameter \texttt{FEMtool} defines, which FEM tool, ANSYS or FELyX, should be employed. If \texttt{felyx} is chosen, linear modelling is applied, and if \texttt{ansys} is determined, nonlinear modelling is automatically used. Furthermore an error-fitness value can be assigned if an individual cannot be evaluated. This can be the case when the genotype of an individual causes an error in the FEM computation. The error-fitness is also the maximum fitness value that gets assigned, even if the weighted sum of all fitness terms exceeds this value.

Initialization section

This section contains several information.

- \texttt{NumVertices} is an important parameter. It must correlate with the value of vertices in \texttt{eoGraphOpt.initPop} or \texttt{generationsX.sav}.

- \texttt{BeamEvTp} defines the spline interpolation scheme for the edge thickness definition.

- \texttt{minEdgeThickness} defines the minimal edge thickness. Smaller values resulting from the spline interpolation are overruled by this value.

- \texttt{fspring} is a spring constant multiplication factor.

- \texttt{IntPt} defines the number of integration points for the evaluation of the complex-shaped beam stiffness matrix.

- \texttt{EdgeNumMax} defines the maximum number of edges that an individual can have.

- \texttt{EdgeNumMin} defines the minimum number of edges that an individual must have.

- \texttt{minEdgeLength} defines the minimum edge length.

- \texttt{CheckCross} this parameter controls the way that edge crossings and beam overlapping at the corners is handled. A 0 value does not
check any crossing or overlapping. A value of 1 for this parameter activates the checking of crossings. A value of 2 activates the regulation of overlapping regions. Overlapping corners of adjacent edges cannot be computed efficiently by FELyX. Therefore it can be advantageous to limit the overlapping areas of adjacent beams. The length of the beam that is allowed to be overlapped by any adjacent beam is controlled by \texttt{--OverlappRadius}. The effect of \texttt{--noOverlappRatio} is shown in figure B.1

\begin{figure}[h]
\centering
\includegraphics[width=0.7\textwidth]{OverlapEffect.png}
\caption{The effect of the OverlappRadius parameter.}
\end{figure}

\texttt{--NbCheckPt} defines the number of points by which crossing is checked along every edge.

\texttt{--LimitCross} defines the maximum number of crossings that can occur in an individual. To further minimize crossings it is advised to include the number of crossings in the fitness function.

\texttt{--RefMass} is the referential mass used to compute the mass ratio that is often used in the fitness function.

\texttt{--initType} defines whether the initial population is generated uniformly within the possible limits of the genes or whether it is initialized around the given \texttt{eoGraphOpt.initPop} or \texttt{generationsX.sav} with a gaussian distribution.

\texttt{--nbFeasibleIndis} defines the percentage of individuals in the initial population that must be feasible, i.e. their fitness value is below \texttt{--error-fitness} and all the fitness terms have values below the cutoff value.
–**loadPath** contains the path and file to either a *eoGraphOpt.initPop* or a *generationsX.sav* file.

–**runInitialization** controls whether an initialization is run or whether a whole population is continued or run from a *.sav* file. If *eoGraphOpt.initPop* is used for initialization, then this parameter must have a value of 1.

–**recomputeFitness** older computation runs and their *.sav* files contain information on the fitness values of the individuals. If a new population or new fitness definitions are used then the old fitness values are obsolete.

–**keepLoaded Indis** defines whether the individuals coming from the *.sav* file are kept in the initial population after initialization.

**Output section**

This section controls the output to file and screen during an optimization run.

–**saveFrequency** every $F$ generation is stored into a *.sav* file.

–**resDir** contains the path and file to where the results are stored.

–**eraseDir** checks whether the content of the result directory will be deleted prior to the optimization.

–**plots** if plots is set to 1 then the fitness values of all the fitness terms are displayed graphically to the screen.

–**printFile** checks whether a statistics file will be stored or not. The statistics file stores the fitness values of the best individual as well as the average fitness values of each generation into a *.eoStatistics.sav* file in the result directory.

**Persistence section**

The **--status** defines the path and file to where the CompliantGraph-seq.param file will be copied.
B.2 Data Source for the running parameters of an optimization

B.2.0.1 Reordering section

Reordering is a method to order the numbering of vertices and edges in such a way that vertices and edges that lie close to each other also have similar numbers. This has the advantage when mutation occurs. For example if an edge mutates its first vertex from 5 to 6 a smaller change in the phenotype will occur if the vertices are reordered. This increases the linearity in the mutation process: a small change in the genotype generates a small change in the phenotype.

Stopping criterion section

The user of the eoGraphOpt program can control when an optimization run shall be stopped. The entries are self explanatory. Whenever the first criterion is met, the optimization will stop.

Variation Operators section

This section controls the probabilities of applying a certain mutation and cross-over operators. A detailed description of every operator is not given here. Instead, a short explanation how the variation process is performed will be given:

Every individual has a probability \( p_{\text{Cross}} \) (at the bottom of the list) to be chosen in a first selection to perform a recombination. The individuals selected for recombination then undergo a further selection process. The \( x_{\text{-rate}} \)-operators define the relative rate with which selected individuals will actually undergo a recombination step. Which cross-over operator is finally called on an individual is determined by the single probability values of the cross-over operators. After crossover is performed, all individuals have the probability \( p_{\text{Mut}} \) (at the bottom of the list) to be chosen in the first selection for mutation. The \( M_{\text{-rate}} \)-operators define the relative rate with which selected individuals will actually undergo a mutation step. Which mutation operator is called is again determined by the single probability values of the \( p_{\ldots\text{mutation}} \)-operators. Figure B.2 shows the variation process graphically.
every individual of entire population

cross-over

pCross 1-pCross

no cross-over

for every X-operator

for every M-operator

Figure B.2: The variation process.
Appendix C

EoGraph preprocessing tool

It can be tedious to generate data files for an optimization run, since writing and editing the data files by hand is error-prone. Therefore a graphical user interface (GUI) for MATLAB has been implemented. It concurrently writes the \textit{EoGraph.Data}, the \textit{EoGraph.initPop} and the \textit{generations0.sav} files (see Figure C.1), which serve as input files of the optimization. (Appendix B). It is possible to start the setting up of the optimization from a \textit{modelX.matlab} result file. Thereby, the preprocessing tool reads an existing structural design and allows to adjust the optimization setup according to the imported design. Furthermore,
the preprocessing tool visualizes fitness terms of an optimization run (EoGraph.fitTerms) and allows to store as well as to load input data (EoGraph_handles.mat). The layout of the preprocessing tool displayed in Figure C.2 is divided into areas with different functionalities:

(1) Graphical display

(2) Loading of already existing files or result files from previous optimizations

(3) Editing vertex and edge properties

(4) Categories of input data

(5) Area to set data of the respective category

(6) Brief explanation of the respective input data

(7) Legend to the graphical display

Figure C.2: Layout of the preprocessing tool.
A sample output of the graphical display is illustrated in Figure C.3. The figure shows the design space \((x = (0, 250), y = (0, 250))\) which is further limited in \(y\)-direction by a polynomial function indicated by the shaded gray area. It also shows that the current geometry has 14 vertices, 13 of which are fixed in their position and vertex 14 has a range from \((x, y) = (25, 50)\) to \((x, y) = (100, 80)\), indicated by the dashed white lines across the vertex. There are 17 edges, all of the same material, some fixed, some with free thicknesses and angles. Edge 16 is also removable. Finally there are 5 bright green borderlines which may not be crossed by the edges.
C.1.1  Save and load input data

The user has the possibility to save the entered data before writing the input optimization files. When saving, all the data are stored in an EoGraph_handles.mat file which can be reloaded later on. It is also possible to import a .matlab file from a previous optimization run. Since the *.matlab file contains only the phenotype, the user is prompted in pop up windows to specify additional information such as upper and lower limits, mutation parameters and so on. After importing a file the user may add or change the initial geometry.

The clear all data button basically closes the main window and restarts the program.

C.1.2  Edit vertices and edges

Vertices and edges can be edited. To do so, the user has to press on one of the two edit buttons shown in Figure C.2. After picking a vertex or edge using the mouse a window will pop up showing all the information of the vertex or edge respectively. That allows to alter that information.

Deleting an existing vertex, an edge or a support geometry is not possible with these buttons. Instead the user can save the current state and do a restart and import the EoGraph_handles.mat file of the previous session. He will then be given the option to delete geometries.

Another option in the edit area is a zooming function for the axes with a reset button which is basically a replot. The export figure button opens the axes in a better printable color scheme in a new window. From that window the user has the option to export the graph to an image file.

C.1.3  Input data categories

This section provides a more detailed description of the use of the input categories and their functionalities. Nine steps or nine categories, respectively, are necessary for the generation of the input data:

Step 1: Setting up of design space

Figure C.4 displays the design space category. It is advised to first
set the design space and then set the design space limitations. Design space limitations are computed by polynomial functions defined by the coefficients $p_0$ to $p_9$.

**Step 2: Definition of the vertices**

The second category specifies the adding of the vertices. Figure C.5 shows the user interface in more detail. The user sets new vertices by mouse or by entering the coordinates. Optionally, the vertices can be given a range within which they are allowed to move during the evolution process. Furthermore it can be determined whether the vertices must be connected by edges or may not. Already added vertices cannot be edited in this section. An extra button in the edit area has this purpose.

Finally the total number of vertices can be set. If this number exceeds the number of preset vertices then the remaining vertices will be randomly generated within the allowed design space. These vertices
will be freely movable. Adding a maximum vertices number is mandatory before saving data to files. The number can be changed at any time and has to be confirmed by pressing the set button.

**Step 3: Definition of the edges**

Initial edges can be generated in the third category as shown in Figure C.6. *Vertex 1* and *vertex 2* denote the vertices between which the edge will be placed. *Beta 1* and *beta 2* are the relative angles at the vertices and \( t_0 \) to \( t_3 \) are thicknesses at the support points for the spline interpolation of the thickness distribution. See section 2.2
for more information on the definition of the complex-shaped beam element. With the material drop down menu, the user can set the material for the edge. For every value, the user is free to set an upper and a lower limit. The last option defines whether an edge has fixed values, whether it can change angles, thicknesses and material during evolution or whether it can even change vertices and be removed. This setting overrules the limits defined underneath the values if it is set to all fixed.

**Step 4: Setting up of FEM boundary conditions**

FEM constraints are set to existing fixed vertex without range. By clicking on a radio button, the user can activate and deactivate constraints. FEM constraints can be changed any time with this mask by typing the number of the vertex in the top box and setting the appropriate constraints (Figure C.7).
Step 5: Adding support nodes

Support geometries are artificial nodes, springs and borderlines. Fixed nodes are used as connection points for springs. They may be added by mouse or by entering the coordinates. The input mask of the support nodes is displayed in Figure C.8.

Step 6: Adding artificial springs

Springs are added to simulate a certain stiffness or resistance. Springs are always connected between a fixed vertex and a fixed support node and a spring constant $k_1$ is assigned to them.

Step 7: Definition of borderlines

Borderlines define lines that are not allowed to be crossed by any edge and keep areas free of edges. A borderline is defined by the coordinates of two points as shown in Figure C.10.

Step 8: Visualization of the fitness terms

A further feature of the preprocessing tool is the visualization of the fitness definition which is defined in an EoGraph.fitTerms file. The routine reads in the file and plots it to the screen. Changes of the fitness terms can be made by pressing the Edit fitness Def. and editing the opened text file. Figure C.11 shows the fitness visualization mask.

The number of fitness functions applied is only needed when working with the generations0.sav file. It describes the number of fitness terms that count towards the fitness of an individual.
C.1 Graphical layout

Figure C.8: Category; support nodes

Figure C.9: Category; springs

Figure C.10: Category; borderlines

Figure C.11: Category; fitness data
Step 9: Saving the data to the input files

There are several options which the user can choose when saving data to files. The first option is the number of individuals to be created. This number is only of importance when working with the EoGraph.fitTerms file. The Copy files to directory can be used when the data files shall be copied to specific directory than the current working directory in MATLAB. Optionally the user can choose to add an omnipotent edge to the genotype of every individual. An omnipotent edge is an edge that defines limits of the edge properties. The edge will be added to the generations0.sav file. Figure C.12 shows the options available when saving to files.

![Figure C.12: Category; save to files](image)

C.1.4 Legend to the graphical display

The legend shows the color code to the graph in the axes. A red vertex characterizes a vertex which may be unconnected from an edge. A blue vertex must always connect to an edge. A square vertex denotes that it has a FEM constraint other than $dz = 0$. Edges are also characterized by a color code. A full line means that the edge is fixed. A dashed line stands for an edge that can change angle, thickness and material within the given limits during evolution. A dot-dashed line means that the edge can change every value or even be removed. Every material also has a specific color.
Appendix D

EoGraph postprocessing tool

The implementation of the EoGraph postprocessing tool have had several objectives. The first objective was to visualize the EoGraph optimization result, whereas the final outcome of the optimization is a modelX.matlab result file as displayed in Figure D.1. The result file contains information such as the objective value, constraint values and the corresponding fitness values as well as parameters for the complex shape beam representation. These parameters specify the used thickness distribution function, the applied minimal thickness and the radius of the non-restricted areas of overlapping elements (Section 3.4). Furthermore, the result file encodes the graph-based beam structure including the FEM data such as the nodes and their displacements and rotations, the elements and its properties, boundary conditions as well as the list of materials.

The second objective of the postprocessing tool was to provide an interface which allows to quickly transfer the beam structure into a manufacturable design. A Visual Basic Script is used to load the structure into CATIA [89] providing two different modes; automatic and user-assisted generation of the CAD model (Section 4.2). Later on, the CAD model may be loaded into a commercial FEM program for further analysis.
The legend of Figure D.1:

(1) Objective and constraint values as well as the corresponding fitness values

(2) Complex-shaped beam representation parameters

(3) Nodes and their coordinates and displacements

(4) Complex-shaped beam elements and their properties

(5) Loads and boundary conditions

(6) List of materials
Figure D.2: Postprocessing graphical user interface

The graphical user interface functionalities consist of a graphical display (1), several plot functions (2) and functions (3) which increase the user’s comfort such as loading of result files, re-plotting and displaying of the encoded result data. In addition, many plot options (4) are provided such as plotting of the non-displaced and displaced nodes, plotting of the displacement directions, mirroring function, and displaying of the boundary condition as well as of the node, material and layer numbers. A further feature is the possibility to change the thickness distribution function (5). The graphical user interface automatically chooses the thickness distribution function applied during optimization. In a later step, the user may intend to adapt the thickness
distribution function. The optional visualization functions (6) only affect the graphical display and not the plot functions. For example, it zooms in regions of interest, or it opens new figures, or it exports the graphical display to an image file. In addition, the plot fitness function visualizes the objective and constraint mapping functions as well as the corresponding the weighted fitness addends as shown in Figure D.3. This is often very helpful when setting up a new fitness function, in order to check the appropriateness of the fitness function.

Figure D.3: Visualization of the fitness mapping functions and fitness addends (red circle)

The last functionality of the EoGraph postprocessing tool (7) is the possibility to export the beam structure into a commercial FEM or CAD program. The functionality Export Ansys automatically generates a so-called apdl-file (ANSYS Parametric Design Language) consisting of the whole FEM model, which includes nodes, elements,
loads, boundary conditions as well as material properties. This file can be directly imported into ANSYS for further analysis. The functionality **Export Catia** writes a text file that encodes the geometry of the structure. Later on, this file can be read by a Visual Basic script implemented in CATIA.

### D.2 EoGraph postprocessing-Catia interface

Figure D.4 displays the interface implemented in CATIA. Two modes, the automatic and the user-assisted mode, are available, which are explained in Section 4.2. After defining both the name and the path of the text file that encodes the geometry, the user can run the Visual Basic Script by pressing the **Run** button.

![Figure D.4: EoGraph postprocessing-Catia interface](image)
D.2 EoGraph postprocessing-Catia interface

Initial Population Evaluation

\[ F(p) = \sum w_i D_i(p) \]

Fitness

Selection

Replacement

Recombination

Mutation

Replacement

Selection

Figure D.5: Optimization loop
# List of Tables

1.1 General features of compliant mechanisms/structures ... 7

2.1 Shape parameter $L$ and curve length $l$ of the curves plotted in Figure 2.6 .... 32

2.2 Alternately meshed beam geometries .......... 47

3.1 Edge/Beam properties corresponding to Figure 3.2 ... 53

3.2 Vertex/Node properties corresponding to Figure 3.2 ... 53

3.3 Number of checks for overlapping ............... 68

3.4 Three levels of beam complexity and two levels of optimization problem complexity ................. 74

3.5 Inverter optimization; best individuals consisting of three beams ......................... 75

3.6 Inverter optimization; best and average of the fifteen best individuals using fixed nodes ............ 77

3.7 Inverter optimization; best and average of the fifteen best individuals using moving nodes ............ 77

3.8 Gripper optimization; best and average of the fifteen best individuals ......................... 80

3.9 Gripper optimization; geometrical linear and nonlinear analysis ........................... 87

3.10 Inverse gripper optimization; single layer versus multiple layers .................................... 90

3.11 Material properties .................................. 91

3.12 Enhanced mapping; force inverter .................. 100

4.1 Comparison of the versions; using geometrically linear and nonlinear modelling .................. 109
4.2 Parallel gripper with limited overlap .................. 110

5.1 Objective values corresponding to Figure 5.6. ........ 121
5.2 Objective function values using geometrical linear and nonlinear modelling. .................. 123
5.3 Objective function values of the modified structure using nonlinear modelling. .................. 124

A.1 Approximation functions error in % constant stretching concept ........................................ 147
A.2 Statistical measures in % of approximation error of curves shown in Figure A.8 .................. 149
## List of Figures

1.1 Da Vinci’s catapult and crossbow drawings [1] ........... 1
1.2 Minimal invasive surgery gripper (source: EMPA) ..... 2
1.3 Common compliant devices; a binder clip, paper clips, backpack latch, lid eyelash curler, and nail clipper [3] .... 3
1.4 Ice hockey stick ........................................... 4
1.5 Long bow ................................................... 4
1.6 Active Aeroelastic Wing (AWW) project [4] .............. 4
1.7 Micro gripper ................................................ 4
1.8 Macro gripper .............................................. 4
1.9 Definition of the terms .................................... 6
1.10 Pseudo-rigid deflection path approximation; a) Small-length flexural pivot, b) Pseudo-rigid model [3] ....... 9
1.11 Force inverter according to Bendsoe ..................... 10
1.12 Ground structure (left) [32] versus homogenization approach (right) ...................................................... 13
1.13 Example of a basic Evolutionary optimization loop ... 16
1.14 Shape control .............................................. 21

2.1 Parameters of the beam; a) uniform, b) variable thickness beam .................................................... 24
2.2 Comparison of the deformation mechanisms of a beam element depending on the length to thickness ratio .... 26
2.3 Geometrically nonlinear displacement path; a) uniform beam, b) variable thickness beam, c) curved, variable thickness beam .................................................... 28
2.4 Curved, variable-thickness beam parameters ............. 28
2.5 Centerline defined by start- and endpoint parameters and factor L ................................... 29
2.6 Influence of shape parameter $L$; a) $L=0$, b) $L=2$, c) constant stretching and d) constant curvature concept . . . 31
2.7 Shape dependence on parameter $L$ using the same thickness function $t(\sigma)$ and the same $\beta_1$ and $\beta_2$; a) $L=0$, b) $L=2.7$, c) constant stretching concept .................. 34
2.8 Coordinates for defining the centerline shape through endpoints .................................... 36
2.9 Plane element in a curved bar ........................................ 38
2.10 Definition of Function $Y(y)$ (after [74]) ...................... 39
2.11 Coordinates for defining the centerline shape through endpoints .................................... 43
2.12 Meshed beam geometry; a) coarse mesh, b) medium mesh and c) fine mesh .................. 46
2.13 Connection to the main node ........................................ 46
2.14 Mesh dependency; fine and coarse mesh ...................... 47
2.15 Mesh modification; a) original, b) modified mesh ....... 48
2.16 Alternatively meshed gripper ................................. 48
3.1 Optimization disciplines ............................................ 50
3.2 Beam representation (left) and the corresponding graph representation (right) .................... 52
3.3 Example for int_gene and double_gene ....................... 55
3.4 Influence of the mutation parameters $\varepsilon$, which denotes the range of the uniform mutation of an unbounded gene, the standard deviation $\sigma$ of the Gaussian mutation $f(v)$, the bounds $l_{lower}$ and $l_{upper}$ as well as the cyclic properties option .................. 56
3.5 Sample variable length graph genotype ....................... 56
3.6 Split and merge operators; a) split, b) original and c) merged beam structures .................. 58
3.7 Curved, variable-thickness sizing operators .................. 59
3.8 Mapping of the objective value ................................. 61
3.9 Mapping of the constraint value ............................... 61
3.10 Non-crossing centerlines (left) and overlapping of restricted zones (right) .................... 63
LIST OF FIGURES

3.11 Beams divided into restricted and non-restricted overlapping zones ........................................ 63
3.12 Sample layout consisting of thirteen elements ................................................................. 64
3.13 a) circular bonding box, b) bonding box consisting of two circle segments ........................ 65
3.14 Circular bounding boxes ........................................................................................................ 66
3.15 Determination of the intersection point ................................................................................... 67
3.16 Circle segments bounding boxes ............................................................................................ 68
3.17 The graph-based optimization incorporating three libraries (BGL, EO and eoUnigene) as well as the FELyX FE codes ........................................................................................................ 69
3.18 Inverter optimization setup .................................................................................................... 70
3.19 Best solution employing three beams for fixed and moving nodes; a) conventional, b) variable-thickness, c) curved, variable-thickness ................................................................. 75
3.20 Best solution using fixed and moving nodes; a) conventional, b) variable thickness, c) curved, variable-thickness ........................................................................................................ 76
3.21 Solution dependent on spring factor, a) $f = 0.01$, b) $f = 1$, c) $f = 10$, and on spring ratio, d) $k_{in} : k_{out} = 1 : 8$, e) $1 : 1$, f) $8 : 1$ .................................................................................................................. 78
3.22 Gripper optimization setup ..................................................................................................... 79
3.23 Gripper optimization using conventional beams ................................................................. 81
3.24 Gripper optimization using curved beams with a quadratic thickness function .................... 81
3.25 Gripper optimization using curved beams with a cubic thickness function ............................ 82
3.26 Geometric nonlinearity .......................................................................................................... 83
3.27 Linear and nonlinear analysis ................................................................................................. 84
3.28 Newton Approach .................................................................................................................. 85
3.29 Large deflection grippers ........................................................................................................ 86
3.30 Layers .................................................................................................................................. 88
3.31 Single-layer inverse gripper ..................................................................................................... 89
3.32 Multi-layer inverse gripper ...................................................................................................... 89
3.33 Thee-dimensional CAD model ............................................................................................. 90
3.34 Multi-material solutions ........................................................................................................ 91
3.35 Multi-material, inverse gripper .............................................................................................. 92
3.36 Deformed beam element ....................................................................................................... 92
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.37</td>
<td>Mass dependance on the bending stiffness</td>
<td>94</td>
</tr>
<tr>
<td>3.38</td>
<td>Strain dependance on the bending stiffness for a given displacement $u_0$</td>
<td>94</td>
</tr>
<tr>
<td>3.39</td>
<td>CAD model</td>
<td>95</td>
</tr>
<tr>
<td>3.40</td>
<td>Enhanced mapping procedure</td>
<td>97</td>
</tr>
<tr>
<td>3.41</td>
<td>Enhanced mapped structure</td>
<td>98</td>
</tr>
<tr>
<td>3.42</td>
<td>Meshed structure: three of thirteen regions failed to mesh according to the enhanced mapping concept.</td>
<td>99</td>
</tr>
<tr>
<td>3.43</td>
<td>Force inverter optimization; a) complex-shaped beam representation, b) FEM of the optimization, c) Finely meshed CAD model</td>
<td>100</td>
</tr>
<tr>
<td>4.1</td>
<td>Visualization of the structural behavior</td>
<td>102</td>
</tr>
<tr>
<td>4.2</td>
<td>Graph-based design environment</td>
<td>103</td>
</tr>
<tr>
<td>4.3</td>
<td>Optimized gripper</td>
<td>105</td>
</tr>
<tr>
<td>4.4</td>
<td>Contour nodes connected by splines</td>
<td>106</td>
</tr>
<tr>
<td>4.5</td>
<td>Manually generated CAD model of the parallel gripper</td>
<td>107</td>
</tr>
<tr>
<td>4.6</td>
<td>Automated CAD model of the parallel gripper</td>
<td>107</td>
</tr>
<tr>
<td>4.7</td>
<td>Gap between two beams and added chamfer</td>
<td>108</td>
</tr>
<tr>
<td>4.8</td>
<td>Parallel gripper with limited overlaps</td>
<td>110</td>
</tr>
<tr>
<td>4.9</td>
<td>Rapid prototyped parallel gripper</td>
<td>111</td>
</tr>
<tr>
<td>5.1</td>
<td>The backrest clasping around the driver’s back (top view)</td>
<td>115</td>
</tr>
<tr>
<td>5.2</td>
<td>A car seat concept based on two columns and rib-like compliant structures</td>
<td>116</td>
</tr>
<tr>
<td>5.3</td>
<td>Center single column versus two columns (top view)</td>
<td>117</td>
</tr>
<tr>
<td>5.4</td>
<td>Colored driver dummy (green: target back section)</td>
<td>118</td>
</tr>
<tr>
<td>5.5</td>
<td>The axis-symmetric setup; initial shape (outlined), target shape (dashed),</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>the driver’s body (gray), the forbidden area (horizontally lined), external</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loads $F_a$, $F_m$ and $F_r$, and boundary conditions.</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>The target curve (outlined) and the difference of $\Delta y_{err}$</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>optimization (dotted) and $\Delta y_{nstd}$ formulation (dashed).</td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>The optimized topology; target shape (dashed), displaced nodes (grey)</td>
<td>122</td>
</tr>
<tr>
<td>5.8</td>
<td>Manual modification of the optimized topology</td>
<td>123</td>
</tr>
<tr>
<td>5.9</td>
<td>Deformation of the final rib; target shape (dashed), displaced structure (colored)</td>
<td>124</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

5.10 Rib-like compliant mechanisms; a) initial state, b) actuated ......................................... 125
5.11 Inelastic, tubelike air cushion ................................................................. 125
5.12 Actuation; a) initial state, b) actuated ......................................................... 126
5.13 Air cushion model ..................................................................................... 126
5.14 Actuation line force $F_a$ and distance $d$ versus pressure $p$ plot ............................... 127
5.15 Adaptive car seat concept ............................................................................ 128
5.16 Adaptation to driving situation; turning left ................................................... 129
5.17 Nonlinear contact simulations; a) small b) big male/female ............................... 129

6.1 Increasing complexity and efforts .................................................................... 133
6.2 Adaptive car seat concept and a conventional system ........................................ 135
6.3 Aspects concerning the industrial application of compliant systems .................... 136
6.4 Hingeless medical gripper for minimally invasive surgery ..................................... 139

A.1 Definition of relative angles $\gamma_1$ and $\gamma_2$ ................................................. 141
A.2 Shape parameter $\lambda$ data constant stretching concept ...................................... 142
A.3 Range of $\gamma_1$ and $\gamma_2$ underlying the data in Figure A.2 ............................ 143
A.4 Cylinder coordinates $\alpha$ and $\rho$ ................................................................. 143
A.5 Domains of circumferential functions, compare Figure A.6 ................................... 144
A.6 Circumferential distribution of $\lambda$ for $\rho = 90^\circ$ ........................................... 145
A.7 $\lambda$ along point symmetry lines ...................................................................... 146
A.8 Curves obtained by concept c and its approximation ......................................... 148

B.1 The effect of the OverlapRadius parameter ....................................................... 161
B.2 The variation process ....................................................................................... 164

C.1 EoGraph preprocessing input and output files .................................................... 165
C.2 Layout of the preprocessing tool ....................................................................... 166
C.3 axes ................................................................................................................. 167
C.4 Category; design space ..................................................................................... 169
C.5 Category; vertices ............................................................................................ 170
C.6 Category; edges ............................................................................................... 171
C.7 Category; FEM boundary conditions .................................................................. 172
C.8 Category; support nodes .................................................................................. 173
C.9 Category; springs ............................................................................................ 173
C.10 Category; **borderlines** ........................................ 173
C.11 Category; **fitness data** ........................................ 173
C.12 Category; **save to files** ..................................... 174
D.1 modelX.matlab result file .......................................... 176
D.2 Postprocessing graphical user interface .......................... 177
D.3 Visualization of the fitness mapping functions and fitness
   addends (red circle) .................................................. 178
D.4 EoGraph postprocessing-Catia interface ........................ 179
D.5 Optimization loop .................................................... 181
Bibliography


Own publications


Curriculum Vitae

Name: Michael Sauter
Date of Birth: June 21, 1977
Nationality: Switzerland
Address: Rychenbergstrasse 16
           CH-8400 Winterthur

EDUCATION

Ph.D., Centre of Structure Technologies, ETH Zurich  Apr04 – Feb08

Diploma thesis, Alcan, Neuhausen  Oct02 – Feb03
Subject: - Thermoplastic rivets

Study Mechanical Engineering, ETH Zurich  Oct98 – Oct02
Subjects: - Aerospace and lightweight structures
          - Composite structures
          - Numerical fluid dynamics

WORKING EXPERIENCE

40% employment, New Generation Ice Hockey Stick  May05 – Feb07
Collaboration between Busch Composites SA and ETH Zurich
Responsible for the new design of the models 06/07

Employee, Rieter Automotive, Winterthur  May03 – Oct03
Construction and engineering

Internship, Alcan Airex, Altenrhein  Jul01 – Oct01
Construction and engineering

Internship, Spider Town, Tägerwil  Feb99 – Mar99
Construction