

Development of High-accuracy Strain Sensors using Topology Optimization

Tom Jonckers

Thesis voorgedragen tot het behalen
van de graad van Master of Science
in de ingenieurswetenschappen:
bouwkunde, optie Civiele techniek

Promotoren:

Prof. dr. ir. E. Reynders
Prof. dr. ir. G. De Roeck
Prof. dr. ir. M. Schevenels

Assessoren:

Prof. dr. ir. T. Geernaert
Ir. C. Van hoorickx

Begeleider:

Ir. C. Van hoorickx

© Copyright KU Leuven

Without written permission of the thesis supervisors and the author it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilize parts of this publication should be addressed to Faculteit Ingenieurswetenschappen, Kasteelpark Arenberg 1 bus 2200, B-3001 Heverlee, +32-16-321350.

A written permission of the thesis supervisors is also required to use the methods, products, schematics and programs described in this work for industrial or commercial use, and for submitting this publication in scientific contests.

Zonder voorafgaande schriftelijke toestemming van zowel de promotoren als de auteur is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot Faculteit Ingenieurswetenschappen, Kasteelpark Arenberg 1 bus 2200, B-3001 Heverlee, +32-16-321350.

Voorafgaande schriftelijke toestemming van de promotoren is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Preface

Deze masterproef is voor mij het hoogtepunt van mijn opleiding tot burgerlijk ingenieur bouwkunde. Ik had deze thesis, en bij uitbreiding mijn hele opleiding, niet tot een goed einde kunnen brengen zonder enkele zeer belangrijke mensen.

Ten eerste zou ik heel graag mijn promotoren prof. Edwin Reynders, prof. Guido De Roeck en prof. Mattias Schevenels willen bedanken voor hun uitstekende begeleiding. Hun advies en inbreng was onontbeerlijk voor het voltooiën van deze thesis.

Ook mijn begeleider ir. Cédric Van hoorickx zou ik willen bedanken. Hij heeft me steeds op weg geholpen bij het verfijnen van mijn topologische optimalisatie algoritmes.

Next I would like to thank the other members of the FWO research project *Robust vibration-based damage identification with a novel high-accuracy strain measurement system*: prof. Thomas Geernaert, ir. Urszula Nawrot, prof. Francis Berghmans, dr. ir. Ben De Pauw and ir. Dimitrios Anastasopoulos. Their experience in developing the strain upscaling sensor package proved to be invaluable for my own work. Furthermore, their expertise allowed me to get familiar with sensing techniques in which I lacked knowledge prior to this thesis.

Tijdens het praktische gedeelte van mijn thesis werd ik steeds bijgestaan door Tom Henskens. Daarvoor zou ik hem ook willen bedanken.

Tot slot wil ik ook zeker mijn familie bedanken. Mijn ouders hebben mij bijgestaan doorheen mijn volledige opleiding. Zonder hun steun en toeverlaat kon dit eindwerk niet tot een goed einde gebracht worden.

Tom Jonckers

Contents

Preface	i
Abstract	v
Samenvatting	vi
List of Figures and Tables	vii
List of Symbols	xii
1 Introduction	1
1.1 Problem statement	1
1.2 State of the art	2
1.3 Objective	3
2 Standard topology optimization	5
2.1 Introduction to topology optimization	5
2.2 Alterations to the model	7
2.3 Obtained design	15
2.4 Conclusions	19
3 Length scale control	21
3.1 Improvement of the model	21
3.2 Obtained design	25
3.3 Conclusions	27
4 Fibre in pure tension	29
4.1 Improvement of the model	29
4.2 Obtained design	31
4.3 Conclusions	32
5 Combination of length scale control and fibre in pure tension	33
5.1 Improvement of the model	33
5.2 Obtained design	34
5.3 Conclusions	35
6 Selection and discussion of the final designs	37
6.1 Influence of the position of the measurement section	37
6.2 Final designs	39
6.3 Conclusions	45
	iii

7 Numerical validation	47
7.1 Static analysis	47
7.2 Dynamic analysis	53
7.3 Buckling analysis	58
7.4 Conclusions	62
8 Experimental validation	63
8.1 Introduction to fibre optic sensing using fibre Bragg gratings	63
8.2 Preparation	64
8.3 Discussion of the fibre mounting system	69
8.4 Static experiments	73
8.5 Dynamic experiments	81
8.6 Further research	87
8.7 Conclusions	90
9 Conclusions	93
A MATLAB code: Standard Topology Optimization	97
A.1 Input Parameters	97
A.2 Optimization Code	98
B MATLAB code: Length scale control	103
B.1 Input Parameters	103
B.2 Optimization Code	104
C MATLAB code: Fibre in pure tension	111
C.1 Input Parameters	111
C.2 Optimization Code	112
D MATLAB code: Combination of length scale control and fibre in pure tension	117
D.1 Input Parameters	117
D.2 Optimization Code	118
E ANSYS input files	127
E.1 General input file	127
E.2 Input file static analysis	130
E.3 Input file dynamic analysis	131
E.4 Input file buckling analysis	132
F Production data of the optical fibres	133
Bibliography	137

Abstract

In this MSc thesis, high-accuracy strain sensors are developed. The aim of the sensor package is to perform very accurate, one dimensional strain measurements. Once the high-accuracy sensor package is fully developed, it can be used in applications such as force identification or structural health monitoring under ambient loading conditions.

In a first part of this thesis, the design of a transducer is made. This design is obtained using the topology optimization technique. Four different topology optimization models are constructed during this thesis. First, a model with very little constraints is made. Next, the complexity is increased, adding constraints for length scale control and imposing the optical fibre in the sensor package to be loaded in pure tension. Combining all models will result in the most complex topology optimization model. Ultimately, two final designs of the transducers are selected. One design will perform optimal in compression, the other in tension.

After the designs are finalized, a numerical validation of the transducers is performed. First, the results of the topology optimization are verified. Next, other properties of the transducers, which are not included in the topology optimization, are investigated. These properties include the occurring stresses, the eigenfrequencies and buckling modes of the transducers.

In a last part, an experimental validation is performed. A comparison is made between the numerically predicted and the measured performance. The upscaling factors which are obtained during the experiments are lower than the theoretical values. The difference is explained by the strain loss in the connection between the transducer and the optical fibre. This connection was not modelled in the numerical models, which means the loss could not be predicted. Lastly, further steps in the development of the strain upscaling sensor package are proposed, including some possible improvements and additional experiments.

Samenvatting

In deze masterproef worden hoogperformante reksensoren ontwikkeld. Het doel van de reksensoren is om zeer nauwkeurige, één dimensionale metingen uit te voeren. Wanneer dit sensor pakket volledig ontwikkeld is, zou het gebruikt kunnen worden voor toepassingen zoals krachtidentificatie en structurele gezondheidscontroles.

In het eerste deel van deze thesis wordt het ontwerp van de transducer gemaakt. Dit ontwerp wordt bekomen door gebruik te maken van topologische optimalisatie. Vier verschillende topologische optimalisatie modellen worden opgesteld in deze thesis. Het eerste model is vrij eenvoudig en bevat weinig beperkingen. Vervolgens wordt de complexiteit van het model opgedreven door beperkingen, zoals een minimale lengteschaal en zuivere trek in de vezel, toe te voegen. Het meest complexe model wordt bekomen door de verschillende beperkingen te combineren. Uiteindelijk worden er twee definitieve ontwerpen geselecteerd voor de transducers. Eén ontwerp is gemaakt om optimaal te presteren onder druk, het ander ontwerp werkt beter in trek.

Eens de definitieve ontwerpen bekomen zijn, wordt een numerieke validatie uitgevoerd. Eerst worden de resultaten bekomen tijdens de topologische optimalisatie geverifieerd. Vervolgens worden enkele eigenschappen gecontroleerd die niet in de topologische optimalisatie opgenomen waren. De spanningen in de transducers, alsook de eigenfrequenties en de knikmodes worden hierbij nagekeken.

In het laatste deel van deze thesis zal ook een experimentele validatie uitgevoerd worden. Er wordt een vergelijking gemaakt tussen de numeriek voorspelde en de opgemeten prestaties. De uitvergrotingsfactoren die bekomen worden tijdens de experimenten blijken lager te zijn dan de voorspelde waardes. Het verschil wordt verklaard door rekverliezen in de verbinding tussen de transducer en de optische vezel. Deze verbinding was niet gemodelleerd in de numerieke modellen, waardoor het verlies in rek niet voorspeld kon worden. Tot slot worden nog enkele toekomstige stappen in de ontwikkeling van de hoogperformante reksensoren aangeraden, waaronder mogelijke verbeteringen en toekomstige experimenten.

List of Figures and Tables

List of Figures

1.1	Example of a transducer (Design by KU Leuven, Structural Mechanics section and VUB, B-Phot section).	2
2.1	Principle of the density filter [26].	7
2.2	Dimension restrictions.	10
2.3	Design domain.	11
2.4	(a) Bolt connection of the transducer (Design by VUB, B-Phot section, picture blurred due to patent application) and (b) Clamping bracket of the optical fibre.	12
2.5	Boundary conditions at the bolt connection.	12
2.6	The heaviside function and the heaviside projection filter used for $\beta = 1$, $\beta = 10$ and $\beta = 25$	15
2.7	Influence of the value of the volume fraction on the upscaling factor. . .	16
2.8	Design obtained by the standard topology optimization algorithm. . . .	17
2.9	Hinges present in the design. Green circles indicate hinges which are acceptable in size, red circles indicate hinges which are too small (single node or one element hinges).	18
2.10	Formation of a lever, which will rotate the mounting platform and cause bending in the fibre.	18
2.11	Rotation of the clamping brackets, which results in bending of the fibre. .	19
2.12	Forces occurring in the optical fibre. At the top tension as well as bending, at the bottom only tension.	19
3.1	(a) Eroded ($\eta = 0.7$), (b) intermediate ($\eta = 0.5$) and (c) dilated ($\eta = 0.3$) designs of the force inverter as discussed in Wand et al. [39].	22
3.2	Threshold projection filters used in figure 3.1 for the (a) eroded ($\eta = 0.7$), (b) intermediate ($\eta = 0.5$) and (c) dilated ($\eta = 0.3$) designs. . .	22
3.3	Verification of the implementation of the algorithm by calculating the (a) eroded ($\eta = 0.7$), (b) intermediate ($\eta = 0.5$) and (c) dilated ($\eta = 0.3$) designs of the force inverter benchmark.	24
3.4	Relation of threshold parameter η to manufacturing error ϵ and filter radius r_{min}	25

3.5	(a) Eroded ($\eta = 0.69$), (b) intermediate($\eta = 0.5$) and (c) dilated ($\eta = 0.31$) designs of the transducer obtained by the modified robust topology optimization formulation based on erosion, intermediate and dilation projections.	26
3.6	Hinges present in the intermediate design.	26
4.1	Altered design domain to ensure pure tension in the fibre.	30
4.2	Design obtained by the algorithm which ensures pure tension in the optical fibre.	31
4.3	Design obtained with the mounting platform at the top of the design domain.	32
5.1	(a) Eroded, (b) intermediate and (c) dilated designs of the transducer obtained by the final, most robust topology optimization formulation.	34
5.2	(a) Eroded, (b) intermediate and (c) dilated designs of the transducer obtained by the final, most robust topology optimization formulation with the measuring section positioned at one quarter of the height of the transducer.	35
5.3	Detail of the hinged connection of the mounting platform.	35
6.1	Influence of the vertical position of the measurement section on the upscaling factor when using the topology optimization with length scale control.	38
6.2	Influence of the vertical position of the measurement section on the upscaling factor when using the topology optimization with length scale control and fibre in pure tension.	38
6.3	Intermediate designs with the measurement section at position (a) 5 and (b) 25.	39
6.4	(a) Eroded, (b) intermediate and (c) dilated designs selected topology for the transducer optimized in compression.	40
6.5	Design for the transducer optimized in compression after post-processing.	40
6.6	CAD drawing of the transducer optimized in compression.	41
6.7	Transducer optimized in compression after production.	41
6.8	Starting material distribution (a) before and (b) after flipping the connection arms.	42
6.9	(a) Eroded, (b) intermediate and (c) dilated designs obtained using the algorithm with length scale control for the transducer optimized in tension.	43
6.10	Design for the transducer optimized in tension after post-processing.	44
6.11	CAD drawing of the transducer optimized in tension.	44
6.12	Transducer optimized in tension after production.	45
7.1	Upscaling factor calculated in ANSYS for the design in (a) compression and (b) tension.	48
7.2	Design optimized in compression, (a) before and (b) after post-processing.	49
7.3	Upscaling factor after post-processing for the transducer optimized in (a) compression and (b) tension.	49

7.4	Displacements (a) in true scale and (b) enlarged 500 times for the design optimized in compression with 2000 μS in the optical fibre.	50
7.5	Displacements (a) in true scale and (b) enlarged 500 times for the design optimized in tension with 2000 μS in the optical fibre.	51
7.6	Graph of the displacement at the fixation node of the fibre as function of time for the design optimized in (a) compression and (b) tension.	51
7.7	Von Mises stresses in the transducer optimised in compression with 2000 μS in the optical fibre.	52
7.8	Von mises stresses in the transducer optimised in tension with 2000 μS in the optical fibre.	53
7.9	Placement of the four clamping brackets on the two mounting platforms of the transducer: two at the front and two at the back of the transducer.	55
7.10	Mode shapes of eigenmodes (a) one (60.7 Hz), (b) two (79.7 Hz), (c) three (180.2 Hz), (d) four (414.8 Hz) and (e) five (461.0 Hz) for the transducer design in compression.	56
7.11	Mode shapes of eigenmodes (a) one (67.2 Hz), (b) two (85.0 Hz), (c) three (221.1 Hz), (d) four (431.0 Hz) and (e) five (467.7 Hz) for the transducer design in tension.	57
7.12	Buckling shapes (a) one (-283.8), (b) two (75.0), (c) three (193.5), (d) four (464.7) and (e) five (508.5) for the transducer design in compression.	60
7.13	Buckling shapes (a) one (-222.5), (b) two (-212.6), (c) three (-115.1), (d) four (-42.1) and (e) five (188.0) for the transducer design in tension.	61
8.1	Working principle of fibre Bragg gratings [13].	64
8.2	Example of a reflected light spectrum of a fibre containing four FBGs.	64
8.3	Different components of the strain upscaling sensor package: (1) interrogater, (2) optical fibre, (3) transducer and (4) clamping brackets.	65
8.4	Configuration of the FBGs: (1) reference strain, (3) upscaled strain and (4) upscaled strain.	66
8.5	Eccentricity of the reference FBG.	67
8.6	Temperature FBG.	67
8.7	New location of the reference FBG.	68
8.8	New location of the clamping brackets of the reference FBG.	68
8.9	Graph with strains in the reference FBC in (a) eccentric and (b) centered configuration.	69
8.10	Clamping brackets used to install the fibre on the transducer.	70
8.11	Fibre clamping system in the configuration it was previously used in other experiments.	70
8.12	Deformation of the rubber while subjected to shear forces.	71
8.13	Experiment set-up to compare the performance of both rubbers.	72
8.14	Comparison of the strain loss of the old and new rubber.	72
8.15	Set-up of the static experiments.	74
8.16	Detailed view of the chain connection of the weights to avoid eccentric loading and bending of the transducer.	75

8.17 Measured strains during a first static test on the design optimized in compression: (a) all strains and (b) detail of the reference strains. . . .	76
8.18 Temperature change during the first experiment.	76
8.19 Relation between the (a) measured and (b) theoretical reference strain and the upscaled strains.	77
8.20 Results of a second static test on the design optimized in compression: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.	78
8.21 Results of a third static test on the design optimized in compression: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.	78
8.22 Results of a first static test on the design optimized in tension: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.	79
8.23 Results of a second static test on the design optimized in tension: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.	80
8.24 Results of a third static test on the design optimized in tension: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.	80
8.25 (a) Connection to the electromechanical shaker and (b) detail of the load cell.	81
8.26 (a) Applied force and (b) measured and theoretical upscaled strains for the dynamic test with a loading frequency of 2 Hz on the design optimized in compression.	82
8.27 Relation between the theoretical reference strains and the measured upscaled strains for the dynamic test with a loading frequency of 2 Hz on the design optimized in compression.	83
8.28 (a) Applied force and (b) measured and theoretical upscaled strains for the dynamic test with a loading frequency of 4 Hz on the design optimized in compression.	84
8.29 Relation between the theoretical reference strains and the measured upscaled strains for the dynamic test with a loading frequency of 4 Hz on the design optimized in compression.	84
8.30 (a) Applied force and (b) measured and theoretical upscaled strains for the dynamic test with a loading frequency of 2 Hz on the design optimized in tension.	85
8.31 Relation between the theoretical reference strains and the measured upscaled strains for the dynamic test with a loading frequency of 2 Hz on the design optimized in tension.	86
8.32 (a) Applied force and (b) measured and theoretical upscaled strains for the dynamic test with a loading frequency of 4 Hz on the design optimized in tension.	86

8.33	Relation between the theoretical reference strains and the measured upscaled strains for the dynamic test with a loading frequency of 4 Hz on the design optimized in tension.	87
8.34	Altered fibre mounting mechanism: the addition of a second clamping bracket with a dummy fibre between the two brackets.	88

List of Tables

3.1	Objective values of the force inverter benchmark case.	24
7.1	Eigenfrequencies of the design in compression.	54
7.2	Eigenfrequencies of the design in tension.	54
7.3	Load multipliers for buckling of the design in compression.	59
7.4	Load multipliers for buckling of the design in tension.	59
8.1	Properties of fibre one.	65
8.2	Properties of fibre two.	66
8.3	Summary of the static experiments on the transducer optimized in compression.	79
8.4	Summary of the static experiments on the transducer optimized in tension.	81
8.5	Summary of the dynamic experiments on the transducer optimized in compression.	85
8.6	Summary of the dynamic experiments on the transducer optimized in tension.	87

List of Symbols

Symbols

β	Smoothness parameter
ϵ	Manufacturing error
η	Threshold parameter
c	Objective function
E_0	Young's modulus of the solid material
E_e	Young's modulus of an element
E_{min}	Artificial stiffness of an empty element to avoid the stiffness matrix from becoming singular (10^{-9} MPa)
F	Global force vector
f_v	Volume fraction
k	Spring stiffness
K	Global stiffness matrix
K_e	Element stiffness matrix
l_{out}	Output node selection vector
p	Element stiffness penalization factor
r_{min}	Density filter radius
U	Global displacement vector
U_{ero}	Displacement vector of the eroded design
U_{int}	Displacement vector of the intermediate design
U_{dil}	Displacement vector of the dilated design
V_0	Total volume of the design domain
V_{mean}^*	Maximum mean value of the volume fraction
$V(x)$	Material volume in the design domain
x	Vector containing the element densities x_e
x_e	Element densities
\tilde{x}_e	Intermediate density of an element
\bar{x}_e	Physical density of an element
$\bar{x}_{e,ero}$	Vector containing the physical element densities of the eroded design
$\bar{x}_{e,int}$	Vector containing the physical element densities of the intermediate design
$\bar{x}_{e,dil}$	Vector containing the physical element densities of the dilated design

Chapter 1

Introduction

1.1 Problem statement

The measurement of strains in structures is an important aspect when monitoring a structure. Today optical fibres are getting increasingly popular. They have many advantages such as their flexibility, their accuracy, their ability to measure over long distances and their applicability in harsh environments. In the past, they have mostly been used to measure strain in a quasi-static case. However, it would be very useful to be able to measure dynamic strains as well. Some interesting applications of the dynamic strain data would be health monitoring, fatigue analysis and force identification. In fact, from this data one is able to not only detect damage, but also localize and quantify it. This could be very useful to be able to investigate older structures and determine the acceptable load and remaining life span of the structure.

If a structure is damaged, the local and overall stiffness will be lower than observed in an undamaged structure. This decreased stiffness will result in a change in the eigenfrequencies, mode shapes and damping ratios, which can be calculated from the dynamic strain data. An easy way to obtain these data would be to measure the dynamic strain under ambient excitation. The problem with this is the fact that the ambient excitations, by for example a pedestrian or car on a bridge, only result in very small strains. Therefore, very accurate strain measurements are required. A strain accuracy of at least $0.1 \mu\text{S}$ is required under ambient loading. These accuracies are hard to obtain using the currently available strain measurement techniques, such as optical fibre strain sensors with Bragg grating (FBG). The accuracy could be increased in another way. When dealing with dynamic strains, the average strains (macrostrains) over a larger length of up to 1 meter are of interest. For this reason, one could design a mechanical device which concentrates the total deformations over a long length (up to 1 meter) onto a small measurement section on which the optical fibre is attached. In this way, as much of the deformations as possible are concentrated on the FBG, with a gauge length of 2 cm. This results in a local strain upscaling. In figure 1.1, an example of such a *strain upscaling sensor package* or *transducer* is displayed.

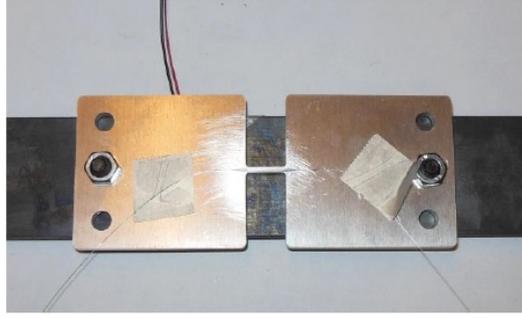


Figure 1.1: Example of a transducer (Design by KU Leuven, Structural Mechanics section and VUB, B-Phot section).

1.2 State of the art

In light of structural health monitoring using dynamic strain measurements, a lot of research has already been conducted, including by the Structural Mechanics section at the KU Leuven. During this research, the eigenfrequencies, mode shapes and modal curvatures are examined on their applicability when assessing possible damage on structures. This is done by comparing these characteristics for a damaged structure and an undamaged theoretical model. By updating the material properties of the structure in the theoretical model at places where the structure is damaged, a correspondence of the eigenfrequencies, mode shapes and modal curvatures is obtained. The updated material properties can be used to assess the health of the structure. This dynamic analysis proves to be a very powerful tool to investigate the health of a structure as already confirmed by research conducted in laboratory conditions, as well as on real-life structures. Interesting research on this subject has been published by Abdel Wahab et al. [1], Teughels et al. [37], Unger et al. [38], Reynders et al. [25] and Casas et al. [4], among many others. Research on the application of fibre optic measurements for structural health monitoring has been published by for example Glisic and Inaudi [13].

To improve this health monitoring, a higher measurement accuracy could be useful. The accuracy can be increased by improving the accuracy of the optical fibre itself, which is outside the scope of this MSc thesis, or by mechanical upscaling of the strain. This upscaling is already discussed by Suzhen Li and Zhishen Wu.[34] Mechanical strain upscaling can be obtained by developing a sensor package which concentrates the strain over a small measurement section. An ideal tool to develop this sensor package is topology optimization. Using topology optimization, one can find the optimal distribution of material in order to achieve a strain upscaling which is maximized. The concept of topology optimization has been meticulously explained by Sigmund [31] and Andreassen et al. [2]. These papers clarify how the available material is distributed over the design domain in order to maximize or minimize the objective function.

The design of the transducer will be a mechanical system. The design of mechanical systems using topology optimization has already been researched on a much smaller scale, namely for the development of MEMS (micro-electro-mechanical systems). These are compliant mechanisms which try to minimize or maximize the output displacement, given an input displacement. This is very interesting in light of this MSc thesis, given that the objective of this thesis is to develop a sensor package which maximizes the strain over a measurement section. Research on this topic has been published by Sigmund [30, 33], Pedersen et al. [24] and Maute et al. [23]. Lastly, additional constraints can be implemented in order to obtain a robust design. This is for example interesting to avoid an infinitely small hinge, which cannot be manufactured. The implementation of robust topology optimization algorithms have been previously researched by Wang et al. [39] and Schevenels et al. [27].

1.3 Objective

The objective of this MSc thesis is to design a new strain upscaling sensor package. This will include the tasks of topologically optimizing the sensor package and testing it in the laboratory.

In a first stage, the transducer will be designed. This will be done using topology optimization. The main goal of this optimization is to achieve a strain upscaling which is as large as possible. However, other effects will be considered as well in order to obtain a robust design. These effects include length scale control and loading of the fibre in pure tension. The design should meet the following requirements:

- The overall length over which strains are measured is equal to 500 mm.
- The length of the measurement section where the optical fibre will measure must be at least 20 mm.
- The width of the transducer should not be larger than half the length over which the strains are measured. This means the width should remain smaller than 250 mm.
- The dimensions of the hinges formed in the design should be larger than the thickness of the transducer.
- The fibre should be loaded in pure tension.
- The stresses in the transducer should remain below the yield limit, avoiding plastic deformations.
- All eigenfrequencies of the transducer should remain above 50 Hz.
- The transducer should not be sensitive to buckling.

In a second stage, when the design of the transducer is finalized, the sensor package will be manufactured by laser cutting and tested in the laboratory. The static and dynamic performance of the transducer will be validated during experiments.

Chapter 2

Standard topology optimization

In this chapter, a first design of the transducer will be made, entirely based on the method discussed by Andreassen et al. [2]. Therefore their 88 lines of MATLAB code will be used as a basis for the topology optimization of the transducer. The working principle of that algorithm will be discussed first. Next, some significant changes will be made in order for the algorithm to be applicable for designing the transducer. These changes include defining a different objective function, redefining the design domain and implementing a threshold projection filter based on Wang et al. [39].

2.1 Introduction to topology optimization

When designing a structure, the design must always meet some specific objectives. In civil engineering, these objectives are typically the requirement to withstand a certain load and not to deform too much under this load. Topology optimization is a technique which distributes a certain amount of material over the design domain in an optimized way, in order to meet the specified objective.

The topology optimization algorithm used in this thesis, is based on the 88 lines of MATLAB code presented by Andreassen et al. [2]. This algorithm will be used as a starting point to which multiple changes and extensions will be added. Therefore the basic building blocks of the topology optimization algorithm of Andreassen et al. will be briefly mentioned.

The topology optimization algorithm is an iterative algorithm. Each iteration the design is improved until an optimal design is reached. During each iteration a finite element analysis of the design is performed, in order to determine the stiffness and the displacements of the design. This finite element analysis is based on a four node element with two degrees of freedom per node: a horizontal and a vertical translation.

Once the finite element analysis is performed, the next step is to determine the

objective function. Typically an objective function looks as follows:

$$\begin{aligned}
 \min_x & : c(x) \\
 \text{subject to} & : KU = F \\
 & : \frac{V(x)}{V_0} \leq f_v \\
 & : 0 \leq x \leq 1
 \end{aligned} \tag{2.1}$$

In this equation, x is the vector with the element densities x_e , ranging from 0 (void) to 1 (solid). These element densities will be altered to obtain an optimal objective value c . K is the global stiffness matrix, U is the global displacement vector containing the displacements in all degrees of freedom and F is the global force vector. $V(x)$ is the material volume and V_0 is the total volume of the design domain. The ratio of these parameters must be smaller than a prescribed volume fraction f_v . The optimum of the objective function is determined using the optimality criteria method.

The volume fraction is the maximal value of the ratio of the filled elements to the void elements. A volume fraction of 0.2 for example means that maximum 20% of all elements are allowed to be completely filled with material. However, if it turns out that the use of less material (for example 15%) results in a better objective value the algorithm will automatically use less material.

The vector x contains the design variables. These values will be optimized in order to obtain a final design. The element densities x_e , which are stored in vector x , are related to the Young's modulus E_e of the element. This relation is called the *modified Solid Material with Penalization* approach, or modified SIMP approach. This approach is formulated in equation 2.2.

$$E_e(x_e) = E_{min} + x_e^p(E_0 - E_{min}) \tag{2.2}$$

Where E_{min} is a very small stiffness (10^{-9} MPa) to avoid the stiffness matrix from becoming singular. E_0 is the real Young's modulus of a solid material. From this equation one can see that when an element is void ($x_e = 0$) the stiffness of that element will be nearly zero, when an elements is solid ($x_e = 1$) the stiffness is equal to the real stiffness of the solid material. A stiffness is also assigned to elements with a density between 0 and 1. However, these densities must be avoided as much as possible, as they do not exist in the physical world. For this reason a penalization factor p is introduced, which will penalize densities between 0 and 1. Typically a value of $p = 3$ is used in topology optimization.

A final important building block to mention is the density filter. A common problem in topology optimization is the existence of checkerboard patterns. A checkerboard pattern is a distribution of material in which solid and void elements alternate. Applying a density filter will eliminate checkerboard patterns by averaging the element densities over a number of elements in a circular filter region (hence filter radius),

see figure 2.1. However, this will again introduce a number of gray elements at the edge between the solid and void regions.

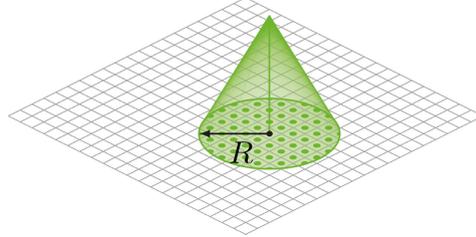


Figure 2.1: Principle of the density filter [26].

As previously mentioned the optimum of the objective function is determined using the optimality criteria method. In the paper of Andreassen et al. [2] it was already mentioned that this method cannot be applied for topology optimization problems with more than one constraint. Given that the more robust topology optimization, which will be used in chapters 3 to 5, requires multiple constraints, a more versatile optimization algorithm will be implemented. Andreassen suggested the use of the method of moving asymptotes (MMA) introduced by Svanberg [35].

2.2 Alterations to the model

Some alterations will be made to the standard topology optimization algorithm of Andreassen et al. [2] which was described previously.

2.2.1 Objective function

The objective of the topology optimization of the transducer is to enlarge the strain as much as possible and concentrate it over the measurement section. The simplest way to implement this, is to maximize the displacements in the node where the optical fibre is attached. This node is displayed in red in figure 2.3. The objective function of the optimization algorithm is the following:

$$\begin{aligned}
 \min_x : c &= l_{out}^T U + 100 \\
 s.t. : KU &= F \\
 &: \frac{V(x)}{V_0} \leq f_v \\
 &: 0 \leq x \leq 1
 \end{aligned} \tag{2.3}$$

In this equation, c is the objective value, l_{out} is the vector containing the degree of freedom (DOF) for which the displacement should be optimized and U is the vector

containing the displacements in all degrees of freedom.

The optimal value of this objective function will be calculated using the method of moving asymptotes (MMA), implemented in MATLAB by Svanberg.[35] When using the MMA algorithm, the objective value in more robust topology optimization is not allowed to become negative. Hence a constant term of 100 is added in equation 2.3. To be consistent and to be able to compare results, this constant term is also added for the standard topology optimization method. More explanation on the importance of this constant term will follow in chapter 3. This constant term will not have an influence on the obtained design, nor on the upscaling factor.

It is also worth noting that the MMA algorithm is implemented in such a way that only a minimum can be obtained. This poses a problem, as the goal of the transducer is to enlarge the displacements as much as possible. There are two possible solutions for this problem. The first one is to place a minus sign in front of the objective function. That way a minimum will be transformed into a maximum. This implementation will work for the standard topology optimization algorithm discussed in this chapter, however, it will not work for the more robust topology optimization algorithms from chapters 3 to 5. Therefore, a second solution is preferred: changing the sign of the input force. When this force is positive (the transducer is loaded in tension) the optimization algorithm will converge towards a minimal value, meaning the optical fibre will be in compression. When the input force is negative (the transducer is loaded in compression) the optimization algorithm will also converge towards a minimal value, meaning the optical fibre will be in compression. These two designs will perform differently, both having their advantages. The optical fibre can only be used in tension, meaning the first design will be very useful at locations where compression is expected, as compression on the transducer will result in tension in the fibre. This is for example the case at the top of a simply supported beam. The second design will be more useful at a location where tension is expected, as tension on the transducer will result in tension in the optical fibre. This is for example the case at the bottom of a simply supported beam.

To obtain the optimum of the objective function, the first derivative of this function with respect to the design variable x , containing element densities x_e , must be calculated. Due to the dependency on the displacements U this derivative is hard to obtain. The most effective way for calculating the derivative of the objective function, is to use *the adjoint method* as suggested by Bendsøe and Sigmund.[3] This way the derivatives of the displacements are never calculated explicitly. The technique involves using an altered notation of the objective function:

$$\hat{c} = l_{out}^T U + 100 + \lambda(KU - F) \quad (2.4)$$

The extra term in equation 2.4 is equal to zero, since the equilibrium equation states $KU = F$, where K is the global stiffness matrix, U is the global displacement vector

and F is the global force vector. The derivative of equation 2.4 with respect to x is:

$$\frac{\partial \hat{c}}{\partial x} = \lambda^T \frac{\partial K}{\partial x} U + (l_{out} + \lambda K) \frac{\partial U}{\partial x} \quad (2.5)$$

In this equation, the term λ can be chosen freely. It is however very interesting to chose $\lambda \equiv -K^{-1}l_{out}$ as this will cause the second term in equation 2.5 to be equal to zero, resulting in the final equation:

$$\frac{\partial \hat{c}}{\partial x} = \lambda^T \frac{\partial K}{\partial x} U \quad (2.6)$$

This means only the global stiffness matrix K has to be derived with respect to x in order to obtain the derivative of the objective function. Global stiffness matrix K is defined as:

$$K = (E_{min} + x_e^p(E_0 - E_{min}))K_e \quad (2.7)$$

Where K_e is the element stiffness matrix. The derivative of the global stiffness matrix K with respect to x is equal to:

$$\frac{\partial K}{\partial x_e} = px_e^{p-1}(E_0 - E_{min})K_e \quad (2.8)$$

By calculating the derivative of the objective function as defined in equation 2.6, the optimum of the objective function is determined and the material is distributed in an optimized way over the design domain.

2.2.2 Design domain

The goal of the transducer is to concentrate the displacements over a larger length onto a smaller length, which results in the upscaling of the strains. The decision has been made to concentrate the displacements over a total length of 500 mm onto a measurement section of 20 mm. A further decrease in size of the measurement section would increase the upscaling of the strains. However, due to the nature of the optical fibre, this further decrease in size is impossible. The optical fibre contains Fibre Bragg Gratings (FBG), which measure the strain. The FBG's have a length of 8 mm. The position of this FBG in the fibre is only known to within a few millimetres. To ensure the FBG is indeed located within the measurement section a tolerance must be maintained. For this reason a minimum length of 20 mm is required for the measurement section. Another limitation to the design is the width constraint. The width of the transducer should remain smaller than or equal to half of the length of the transducer. This constraint is reasonable given that the transducer is designed to measure one dimensional, longitudinal strain. This last demand is also due to practical and aesthetical reasons. A transducer with a larger width would be more difficult to attach to for example a bridge. All the restrictions in dimensions are illustrated in figure 2.2.

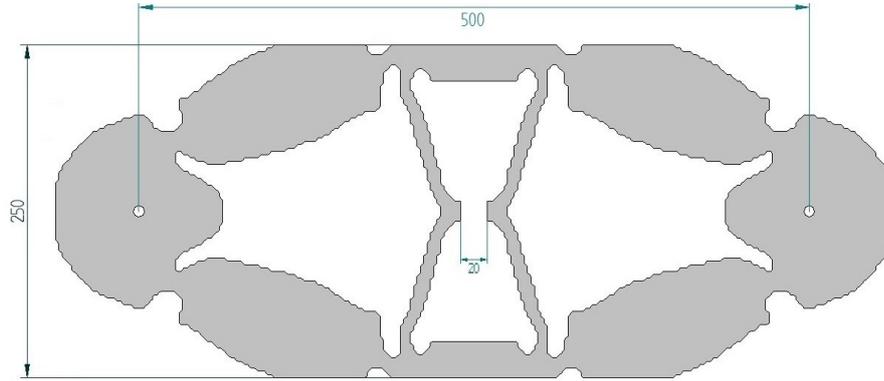


Figure 2.2: Dimension restrictions.

Symmetry of the transducer is a very important aspect to consider when defining the design domain. The transducer is attached to the structure, which is being measured, at two locations, one on each end of the transducer. Both connections exert an equal but opposite force on the transducer. When the measurement section is constrained to be located in the middle of these two connections, it is clear a vertical symmetry plane can be implemented for this transducer. This symmetry plane can be seen in figure 2.3 on the left side. The implementation of a symmetry plane will reduce the size of the design domain by half. This reduces the computational effort when optimizing the topology of the transducer, yet it also reduces the design freedom. A trade-off has to be made between the design freedom and the computational effort. The restriction in design freedom is considered to be sufficiently small, justifying the implementation of the vertical symmetry plane. One could argue that this half structure could be reduced even further to a quarter structure. However, this would further reduce the design freedom. Furthermore horizontal symmetry of the transducer is not guaranteed nor required. Although the vertical position of the measurement section is assumed to be in the middle at the beginning, this might not be the most optimal position. The influence of this position will be discussed further in chapter 6. The implementation of a second, horizontal, symmetry plane would inhibit this analysis.

The implementation of the symmetry plane significantly reduces the size of the design domain. Initially, a length of 750 mm (displacements measured over 500 mm plus an over length of 250 mm) was required. This can now be limited to 375 mm due to symmetry. The width of the design domain remains 250 mm. The design domain will be divided into square elements, with a size of 2.5 mm by 2.5 mm, resulting in a grid of 150 by 100 elements. The size of the elements has been chosen based on two grounds. On the one hand, a smaller element size will result in a better transducer design, especially taking into account the minimal length scale of solid areas, which will be discussed in chapter 3. It will be shown that for this minimal length scale of solid areas, the elements should be smaller than or equal to 2.5 mm by 2.5 mm. On the other hand, a larger element size will result in a faster calculation time.

Considering the argument of calculation time, an element size smaller than 2.5 mm by 2.5 mm is not advisable, especially not for the more robust topology optimization which will be discussed in chapters 3 to 5.

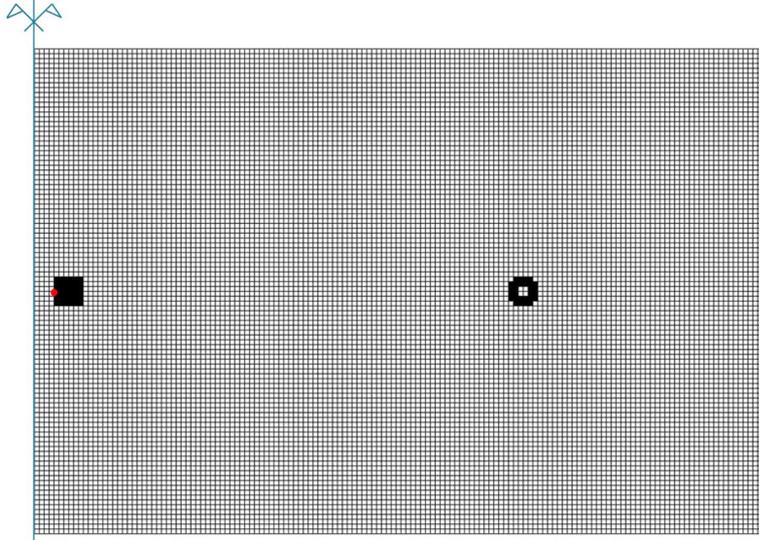


Figure 2.3: Design domain.

In order to attach the transducer to the structure, as well as the optical fibre measuring the strain to the transducer, additional constraints must be imposed to certain elements. First of all, the transducer is attached to the structure it is monitoring using bolts (see figure 2.4a). This means a hole in the transducer must be present for the bolt to fit through. The elements surrounding the bolt must be solid at all time, ensuring the forces can be transferred from the bolt to the transducer. This is implemented into the topology optimization by defining active and passive elements, as suggested by Andreassen et al. [2]. The implementation can be found in lines 56 to 63 in the MATLAB code provided in appendix A.2. An active element means the element will be solid (a density of 1) at all times. Active elements are used for the elements around the bolt. A passive element means the exact opposite, it will be completely empty at all times (a density of 0). These passive elements are used for the elements at the bolt hole. The same technique is used to implement the fixation of the optical fibre. The optical fibre is connected using a clamping bracket, as shown in figure 2.4b. In order for the bracket to be installed, a mounting platform of 15 mm by 15 mm must be present in the design of the transducer. This is implemented using active elements in lines 48 to 55 in the MATLAB code in appendix A.2. The active elements are also visible in figure 2.3 as black elements.

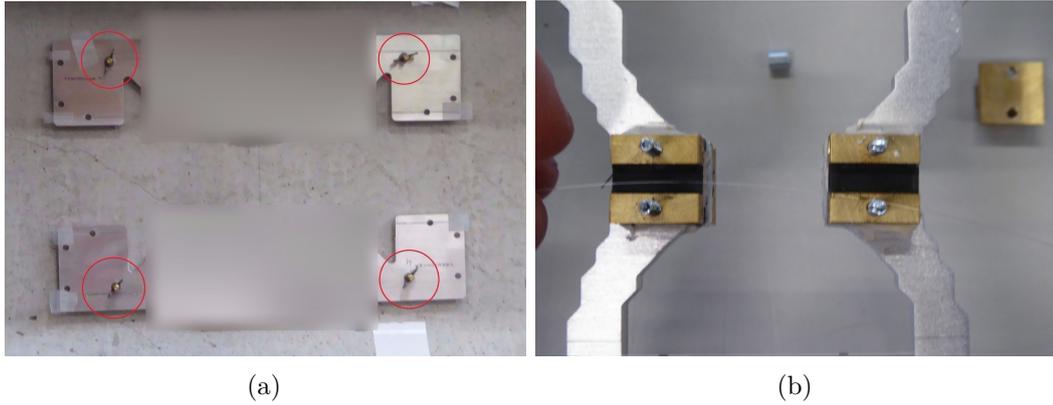


Figure 2.4: (a) Bolt connection of the transducer (Design by VUB, B-Phot section, picture blurred due to patent application) and (b) Clamping bracket of the optical fibre.

Another important aspect which has to be defined in the design domain are the boundary conditions. As previously mentioned, there is a symmetry plane present at the left boundary of the design domain. To define the symmetric boundary condition, all horizontal displacements in the nodes of this symmetry plane should be prevented. Vertical displacements are allowed however. This is implemented in line 42 in the MATLAB code in appendix A.2. Also at the location of the bolt hole, some boundary conditions are implemented. These boundary conditions are displayed in figure 2.5. In all nodes of the active elements around the bolt, vertical displacements are prevented. Horizontal displacements, however, are allowed. This way, the transducer will be loaded in pure longitudinal tension or compression.

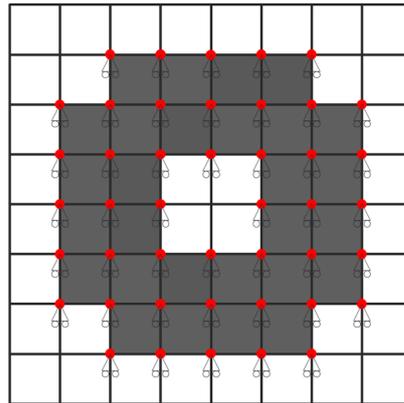


Figure 2.5: Boundary conditions at the bolt connection.

A last aspect of the design domain which is discussed, are the input and output displacements. In all nodes of the active elements around the bolt, a longitudinal input

displacement is defined. This input displacement is provided by the structure to which the transducer is attached. As a result of the way the topology optimization algorithm is implemented in MATLAB, it is not allowed to define an input displacement constraint, since it is designed for an input force constraint. An easy way to solve this problem is suggested by Bendsøe and Sigmund.[3] The implementation of an input displacement is done by attaching a spring to the input node, on which a force is applied. If one knows the spring stiffness and the input displacement, one can easily determine the force that is required to obtain that specific input displacement. To obtain a realistic value of the spring constant of the input spring, the characteristics of the structure to which the transducer is attached are of importance. A concrete beam with a cross section of 300 mm by 300 mm is assumed for this calculation. The spring stiffness of the beam can be calculated using the following formula:

$$\begin{aligned}
 k &= E \frac{A}{l} \\
 &= 30 \text{ kN/mm}^2 \frac{300 \text{ mm} \cdot 300 \text{ mm}}{250 \text{ mm}} \\
 &= 10\,800 \text{ kN/mm}
 \end{aligned}
 \tag{2.9}$$

Where E is the Young's modulus of concrete, A is the area of the cross section of the beam and l is the length of the spring. Note that, due to this spring stiffness, the results of the topology optimization will be, in theory, depending on the structure it is attached to. However, the spring stiffness of the beam is much greater than the stiffness of the aluminium transducer. Therefore the exact value of this spring stiffness will not have an influence on the resulting design. The same design would be obtained when using the spring stiffness of a concrete beam of for example 1000 mm by 1000 mm. It is also important to note that this is the total spring stiffness of the concrete beam. When implementing this into the topology optimization algorithm, a spring is implemented in multiple nodes, which results in multiple springs working in parallel. In order to obtain the spring stiffness of each individual spring, the total spring stiffness must be divided by the number of springs working in parallel.

During the topology optimization, a strain of 100 μS is assumed in the beam. This value has been selected to ensure the stability of the optimization algorithm. A smaller strain would inhibit topology optimization, as the displacements become too small, destabilizing the algorithm. The exact value of the input displacement does not have a direct effect on the obtained design, due to the linear elastic formulation of the topology optimization. The strain in the beam is equal to a displacement at the bolt connection of:

$$\begin{aligned}
 u &= 100 \mu\text{S} \cdot 250 \text{ mm} \\
 &= 0.025 \text{ mm}
 \end{aligned}
 \tag{2.10}$$

To obtain this displacement the following input force is required:

$$\begin{aligned}
 F &= k \cdot u \\
 &= 10\,800\text{kN/mm} \cdot 0.025\text{ mm} \\
 &= 270\text{kN}
 \end{aligned}
 \tag{2.11}$$

It is again important to note that the exact value of this input force has no influence on the obtained design. However, a representative value must be defined in order for the topology optimization to work.

The optical fibre can be modelled as a spring as well. This spring is attached at the middle of the mounting platform of the fibre. This is displayed in figure 2.3 by the red node. The optical fibre has a Young's modulus E of 79 GPa, a radius r of 62.5 μm and a length l of 20 mm. Considering the symmetry plane, the length reduces to 10 mm. This results in the following spring stiffness:

$$\begin{aligned}
 k &= E \frac{A}{l} \\
 &= 79\text{ kN/mm}^2 \frac{\pi \cdot (0.0625\text{ mm})^2}{10\text{ mm}} \\
 &= 0.0908\text{ kN/mm}
 \end{aligned}
 \tag{2.12}$$

2.2.3 Implementation of the threshold projection filter

A common problem when using the 88 lines of code algorithm is the occurrence of gray elements after density filtering. Gray elements have a density between 0 (void) and 1 (solid). However, it is physically impossible to produce such an element, since a material is either solid or void in real-life. Intermediate values do not exist. To solve this problem, the paper of Andreassen et al. [2] suggests the implementation of a black-and-white projection filter. The threshold projection filter suggested by Wang et al. [39] will be used to obtain this black-and-white projection. The black-and-white projected densities, called physical densities, are obtained from the threshold projection filter formulated in equation 2.13.

$$\bar{x}_e = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{x}_e - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))}
 \tag{2.13}$$

Where \bar{x}_e are the physical densities of the elements, \tilde{x}_e are the densities after density filtering called intermediate densities, η is the threshold parameter and β is the smoothness parameter.

Threshold parameter η determines the threshold above which the densities will be projected to solid elements. This parameter is set to $\eta = 0.5$, which means intermediate densities below 0.5 will be projected towards void elements, while intermediate densities above 0.5 will be projected towards solid elements.

Smoothness parameter β determines how well the heaviside function is approximated. With $\beta = 1$ no filtering will take place, when β approaches infinity the filter function will converge towards the heaviside function. The smoothness parameter β will be gradually increased throughout the iterations of the optimization, starting at 1 and increasing by 1% each iteration. This gradual increase ensures the stability of the iterative process. It also means that the black-and-white filtering will get more crisp each iteration. The heaviside function and the approximation of the threshold projection filter are illustrated in figure 2.6, as well as the influence of the β parameter.

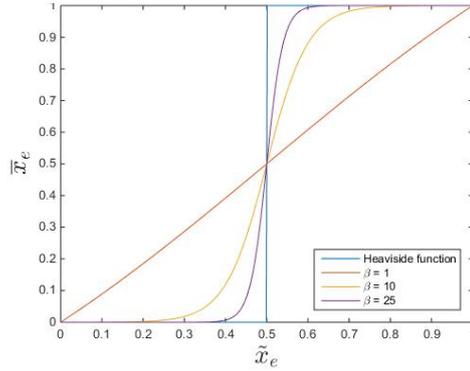


Figure 2.6: The heaviside function and the heaviside projection filter used for $\beta = 1$, $\beta = 10$ and $\beta = 25$.

2.3 Obtained design

In the previous section all changes to the 88 lines of code provided by Andreassen et al. [2] were discussed. In this section, the adapted standard topology optimization will be executed.

2.3.1 Parameters

There are four important parameters which have to be chosen for the topology optimization to work: volume fraction f_v , penalization factor p , filter radius r_{min} and smoothness parameter β .

Volume fraction f_v In the topology optimization of the transducer a volume fraction f_v of 50% is used. The reason for this is that a further increase of the volume fraction would not significantly increase the upscaling factor of the strains. However, it would have a negative effect on the eigenfrequencies of the transducer. These eigenfrequencies are of importance, since the aim of the transducer is to measure dynamics strains. Therefore the eigenfrequencies of the transducer should remain outside of the measuring range, for them not to interfere with the measurements.

The influence of the volume fraction on the upscaling factor is also displayed in figure 2.7.

Analysis of the graph shows two volume fractions f_v (0.4 and 0.45) with a sig-

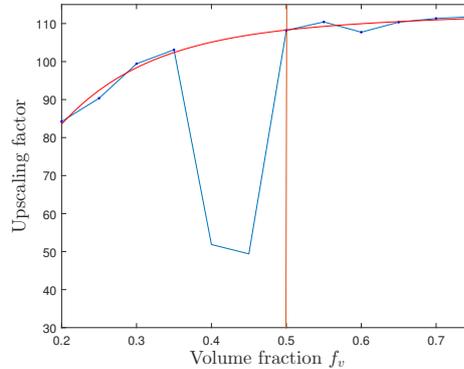


Figure 2.7: Influence of the value of the volume fraction on the upscaling factor.

nificantly lower upscaling factor. These are two designs for which the topology optimization algorithm got stuck in a local optimum. Apart from these two values, the upscaling factor gradually increases until the volume fraction is equal to 0.5.

Penalization factor p Typically a factor $p = 3$ is used in topology optimization.[2] A smaller value will not ensure a black and white solution, while a larger value might cause the algorithm to become less stable. The value of $p = 3$ will therefore be used for the topology optimization of the transducer.

Filter radius r_{min} The exact value of filter radius r_{min} has to be determined by trial-and-error. The paper of Andreassen et al. [2] suggests a filter radius of 0.03 times the width of the design domain. Since the design domain is 100 elements wide, this is equal to 4.5 elements in this case. This value also proves to deliver good results for the topology optimization of the transducer. Therefore the value of $r_{min} = 4.5$ will be selected.

Smoothness parameter β The value of smoothness parameter β should be determined by trial-and-error. A maximum value of $\beta = 25$ has proven to result in crisp edges throughout the design of the transducer, with only one or less gray elements between the void and solid regions. A further increase of smoothness parameter β will not result in a crisper edge, but will cause the algorithm to become unstable. Therefore, a maximum value of $\beta = 25$ is selected.

2.3.2 Resulting design

After the selection of the values of the parameters, the topology optimization of the transducer is executed. It takes a total of 382 iterations for the algorithm to converge

towards an optimized solution. The calculation requires about 360 s or 6 minutes. The obtained design is displayed in figure 2.8.

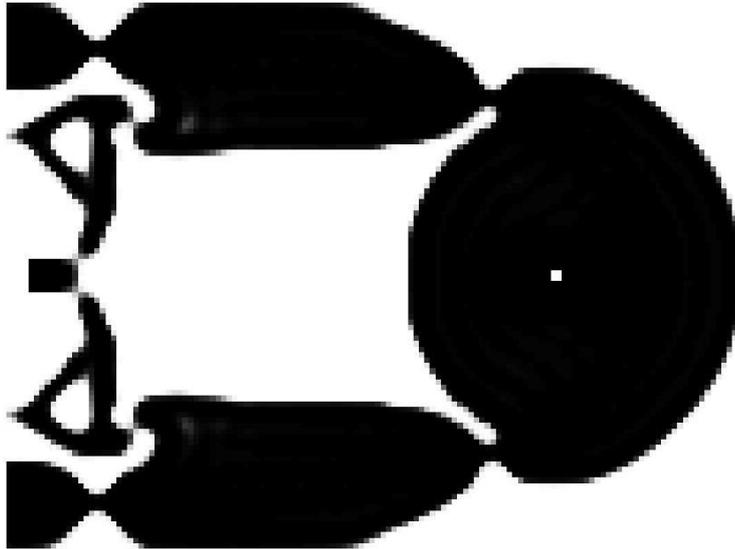


Figure 2.8: Design obtained by the standard topology optimization algorithm.

This design has a very good upscaling factor of 180. The resulting design is a mechanism containing 10 hinges as is visible in figure 2.9.

2.3.3 Possible improvements

The design obtained by the standard topology optimization algorithm provides a good starting point. There are however two major problems with this design. The first and biggest problem is the presence of single node hinges. This problem is visualised in figure 2.9. The implementation of the topology optimization algorithm allows for the full transfer of force from one element to another through one node. In practice, this is of course impossible. The production of a single node hinge is not possible, as a node has an infinitesimally small dimension. Furthermore, no force could transfer through this node from one element to the other. The solution to this problem is to ensure a minimum length scale of the solid areas. This is achieved by a method suggested by Wang et al. [39] and is implemented in chapter 3

A second problem with the standard topology optimization algorithm is the force which is applied to the optical fibre. The design displayed in figure 2.8 has a horizontal symmetry plane, meaning the fixation point of the optical fibre will only move horizontally. No rotation will occur. This results in only axial forces in the optical fibre, which is what the optical fibre is designed to measure. However, this is never

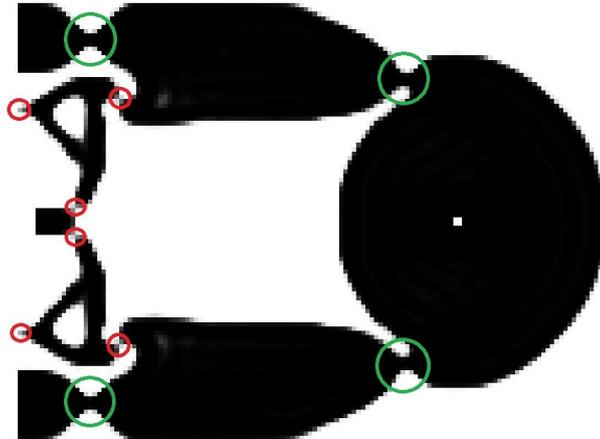


Figure 2.9: Hinges present in the design. Green circles indicate hinges which are acceptable in size, red circles indicate hinges which are too small (single node or one element hinges).

guaranteed by the standard optimization algorithm. For example by changing the position of the mounting platform in the vertical direction, a lever may be formed (see figure 2.10), resulting in a horizontal displacement as well as a rotation of the mounting platform. This rotation of the mounting platform, as well as the bending of the fibre is visualised in figure 2.11. The rotation will result in an axial force and a bending moment on the fibre, as displayed in figure 2.12. It would be a great improvement to the design of the transducer to avoid the occurrence of bending moments on the fibre. This addition will be implemented in chapter 4. The approach to obtain this addition is based on the book of Bendsøe and Sigmund.[3]

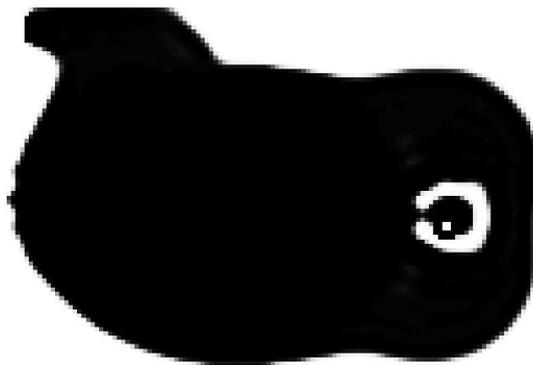


Figure 2.10: Formation of a lever, which will rotate the mounting platform and cause bending in the fibre.

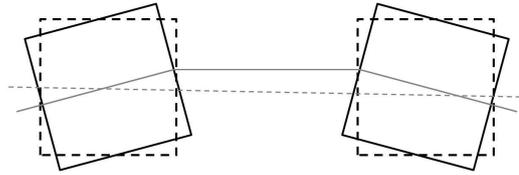


Figure 2.11: Rotation of the clamping brackets, which results in bending of the fibre.



Figure 2.12: Forces occurring in the optical fibre. At the top tension as well as bending, at the bottom only tension.

2.4 Conclusions

In this chapter, the most basic algorithm for the design of the transducer was implemented. This algorithm is based on the 88 lines of code suggested by Andreassen et al. [2]. Some aspects were altered in such a way that the algorithm could be used for the development of the transducer, for example the design domain and the objective function. The implementation of the method of moving asymptotes as well as the threshold projection filter have been introduced to allow the increase of complexity when making the step towards future, more robust topology optimization algorithms.

By executing the topology optimization algorithm, a first, primitive design was obtained. This design already showed a high upscaling of the strain, with an upscaling factor of 180. Some future improvements were suggested, in order to obtain a more robust design. The first improvement is the requirement of a minimum length scale in order to avoid single node hinges. The second improvement is the requirement of a pure axial tension force in the optical fibre, inhibiting bending from occurring. These improvements will be implemented in chapters 3 to 5.

Chapter 3

Length scale control

As suggested in chapter 2 length scale control is an important improvement to the design obtained by standard topology optimization. By introducing length scale control, the occurrence of single node hinges can be avoided. This will result in a design which can be produced and tested. The changes to the model are based on the paper of Wang et al. [39]. After improving the model a new, more robust design is obtained, which will be discussed in more detail.

3.1 Improvement of the model

3.1.1 Applicability of the technique

The modifications which will be made in this chapter are based on a methodology suggested by Wang et al. [39]: Modified robust topology optimization formulation based on erosion, intermediate and dilation projections. Their paper discusses a way to optimize a design taking into account possible manufacturing errors. The result is a design which will still perform as desired even if manufacturing errors occur. This is accomplished by requiring a minimum length scale, which is larger than the manufacturing error, for the solid and void areas. This technique can be applied to the design of the transducer, albeit with some minor modifications. The manufacturing error when laser cutting the transducer is relatively small (< 1 mm). The limiting factor is the size of the hinges. As a rule of thumb, the dimensions of the hinges should not be smaller than the thickness of the transducer. The transducer has a thickness of 5 mm, meaning a minimum length scale of 5 mm will be required using the technique suggested by Wang et al.[39]

3.1.2 Alterations to standard topology optimization

The modified robust topology optimization formulation based on erosion, intermediate and dilation projections will optimize three different designs. A first design is the eroded design, based on the assumption a uniform production error is made along all the edges reducing the solid areas in size and enlarging the void areas. The second design is the intermediate design, which has no manufacturing errors and

which is ultimately the design that will be produced. The third and final design is the dilated design, in which a uniform manufacturing error is assumed along all edges, enlarging the size of the solid areas and reducing the size of the void regions between them. As an illustration, figure 3.1 shows the different designs for the force inverter discussed in the paper of Wang et al. [39].

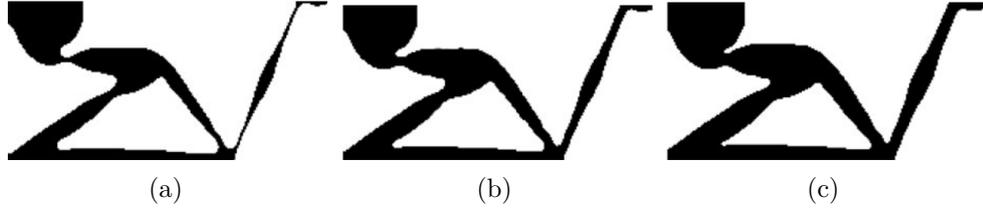


Figure 3.1: (a) Eroded ($\eta = 0.7$), (b) intermediate ($\eta = 0.5$) and (c) dilated ($\eta = 0.3$) designs of the force inverter as discussed in Wand et al. [39].

The three different designs are obtained using the threshold projection filter which was already briefly mentioned in chapter 2. The threshold parameter η initially was set to 0.5, meaning all densities smaller than 0.5 would be projected towards 0, other densities would be projected towards 1. In this chapter, the full potential of the threshold projection filter will be exploited. By changing the value of the threshold parameter η the occurrence of manufacturing errors can be simulated: an increase of the threshold parameter η will mimic the eroded design as more densities will be projected towards zero, a decrease will do the exact opposite and will resemble the dilated design. The different thresholds for the different designs of figure 3.1 are displayed in figure 3.2.

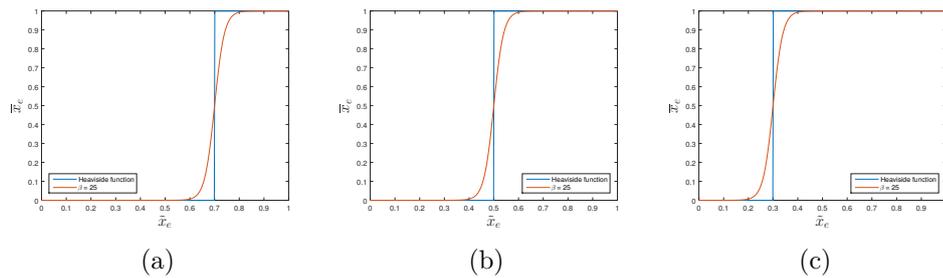


Figure 3.2: Threshold projection filters used in figure 3.1 for the (a) eroded ($\eta = 0.7$), (b) intermediate ($\eta = 0.5$) and (c) dilated ($\eta = 0.3$) designs.

The different designs will be optimized simultaneously using a *min-max* formulation of the optimization problem. Each iteration the least optimal design will be selected by the MMA algorithm and that design will be optimized. To obtain this,

the objective function must be modified to include the necessary extra constraints. This is done in equation 3.1. Due to the extra constraints, the optimality criteria based optimizer originally implemented by Andreassen et al. [2] cannot be used. The MMA algorithm by Svanberg [35] has to be used instead. When the min-max approach is used in combination with the MMA algorithm of Svanberg, the objective values are not allowed to become negative. Otherwise, the wrong design would be selected for optimization. To avoid this problem, a constant factor of 100 has been added to the objective function, as previously mentioned in chapter 2. The selection of the design to be optimized is also the reason the introduction of a minus sign in front of the objective function, to convert a minimum into a maximum, was not an option. If the minus sign would be used, the most optimal design would be selected for optimization instead of the least optimal design.

$$\begin{aligned}
 \min_x : \max & \{l_{out}^T U_{ero} + 100; l_{out}^T U_{int} + 100; l_{out}^T U_{dil} + 100\} \\
 \text{s.t.} : & KU_{ero} = F \\
 & : KU_{int} = F \\
 & : KU_{dil} = F \\
 & : f_v = \frac{\sum \bar{x}_{e,ero} + \sum \bar{x}_{e,int} + \sum \bar{x}_{e,dil}}{3} \leq V_{mean}^* \\
 & : 0 \leq x \leq 1
 \end{aligned} \tag{3.1}$$

Where U_{ero} , U_{int} and U_{dil} are vectors containing the displacements of the eroded, intermediate and dilated design respectively, $\bar{x}_{e,ero}$, $\bar{x}_{e,dil}$ and $\bar{x}_{e,del}$ are vectors containing the physical element densities of the eroded, intermediate and dilated design respectively and V_{mean}^* is the maximum value of the volume fraction which is allowed. Note that the volume constraint f_v is imposed on the mean value of physical densities \bar{x}_e of the three designs. This is different than suggested by Wang et al. [39], where the volume constraint is imposed on the dilated design. A volume constraint imposed on the mean of the three designs has proven to result in less gray elements. The paper of Wang et al. [39] also suggests to update the maximal value of the volume fraction V_{mean}^* every 20 iterations, so the volume fraction of the intermediate design always approximates the maximal value of the volume fraction. This is not implemented because it would reduce the stability of the algorithm, resulting in more gray elements.

One last, important remark has to be made on the min-max approach. The intermediate design, which will ultimately be manufactured, is rarely the least optimal design. This means most of the time the eroded or dilated design will be optimized. Therefore, performance of the intermediate design will decline. However, one has to keep in mind the goal of the technique is to ensure a minimum length scale, making single node hinges impossible. This goal is certainly reached by the min-max approach.

3.1.3 Implementation in the MATLAB code

The optimization with length scale control requires some alterations to the standard topology optimization algorithm. The new robust topology optimization algorithm is given in appendix B.2. The main difference is the fact that three different designs have to be composed. For each design, a separate finite element analysis has to be performed. This is done in lines 162-174. The objective function and the derivatives must be calculated for the three different designs as well. This is done in lines 175-206. The last major change is the way the MMA algorithm works, especially the implementation of the extra constraints. These constraints and their derivatives are defined in lines 126-133.

3.1.4 Verification of the implementation

In order to ensure the implementation in MATLAB yields correct results, the force inverter is optimized as a benchmark case. The resulting designs are displayed in figure 3.3.

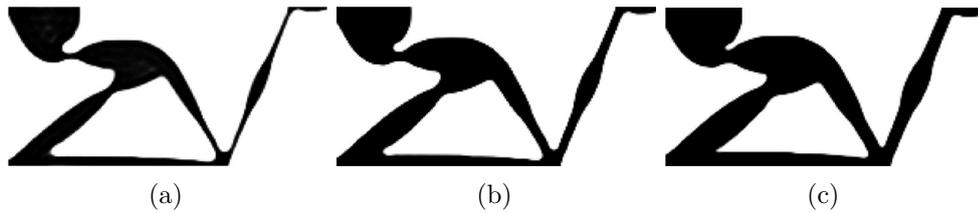


Figure 3.3: Verification of the implementation of the algorithm by calculating the (a) eroded ($\eta = 0.7$), (b) intermediate ($\eta = 0.5$) and (c) dilated ($\eta = 0.3$) designs of the force inverter benchmark.

By comparing figures 3.1 and 3.3, and the objective values in table 3.1, one can see the paper of Wang et al. [39] and the implementation of this thesis yield almost the same results. Therefore, one can conclude the implementation works as desired.

Table 3.1: Objective values of the force inverter benchmark case.

Design	Objective values	
	Wang et al. [39]	Own implementation
Eroded	97.85	98.01
Intermediate	97.70	97.82
dilated	97.85	98.01

3.2 Obtained design

After the alterations to the model are implemented, the more robust topology optimization algorithm will be executed. In this section the new value of the threshold parameter η will be discussed. Next, the obtained design will be discussed and some important remarks will be made.

3.2.1 Parameters

The only parameter which changes compared to the standard topology optimization algorithm is threshold parameter η . Due to the three different designs, three different values for η will be used. The η value is determined as discussed by Wang et al. [39] and Schevenels et al. [27]. They suggest the threshold parameter η is function of the manufacturing error ϵ and the filter radius r_{min} . This relation is depicted in figure 3.4.

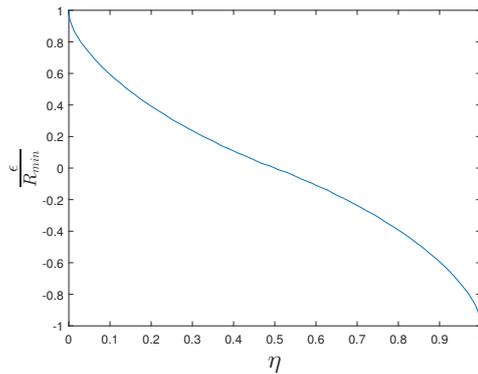


Figure 3.4: Relation of threshold parameter η to manufacturing error ϵ and filter radius r_{min} .

Based on this relationship, and given that $\epsilon = 2.5$ mm (2.5 mm artificial manufacturing error at each side of the hinge) or 1 element and $r_{min} = 4.5$ elements, η is chosen to be equal to 0.31 for the dilated design. This means the threshold for the eroded design will be at $(1 - \eta) = 0.69$. The threshold for the intermediate design is by definition located at $\eta = 0.5$. Due to the artificial manufacturing error of $\epsilon = 2.5$ mm or 1 element, the element size should be 2.5 mm or smaller.

3.2.2 Resulting design

The optimization of the new, more robust design takes 424 iterations. This is about 11% more iterations compared to the 382 iterations it took for the standard topology optimization. The calculation time has drastically increased, from 360 s for the standard topology optimization algorithm to 996 s or 16.6 minutes for the new,

3. LENGTH SCALE CONTROL

robust algorithm. This is an increase of about 177%. The three obtained designs are displayed in figure 3.5.

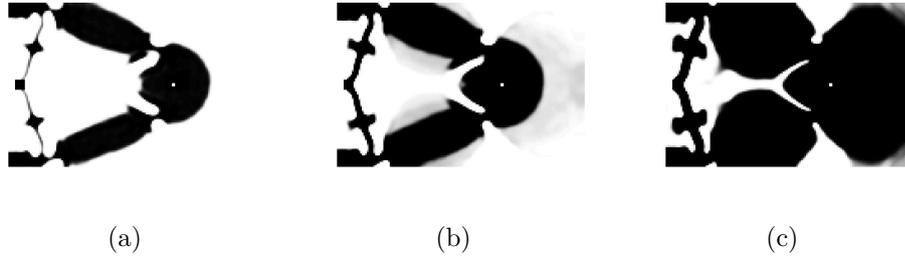


Figure 3.5: (a) Eroded ($\eta = 0.69$), (b) intermediate ($\eta = 0.5$) and (c) dilated ($\eta = 0.31$) designs of the transducer obtained by the modified robust topology optimization formulation based on erosion, intermediate and dilation projections.

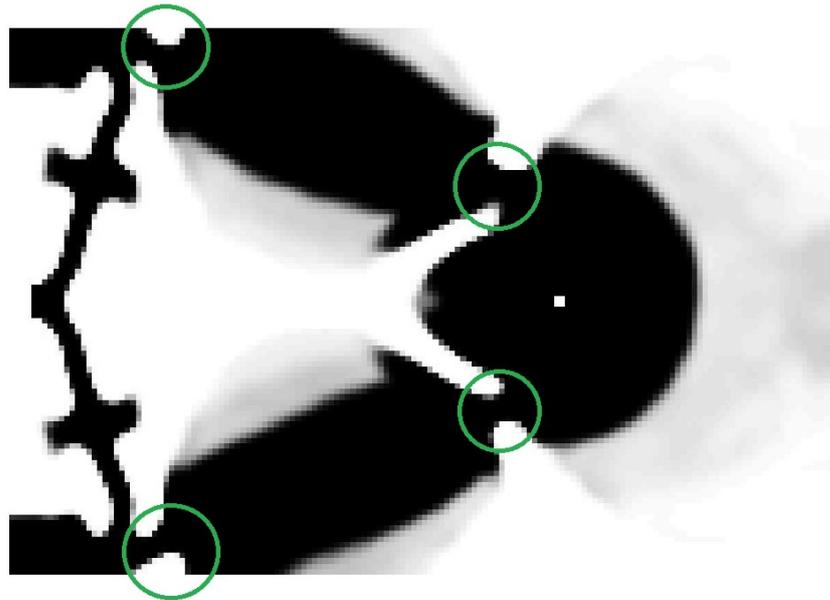


Figure 3.6: Hinges present in the intermediate design.

The intermediate design has an upscaling factor of 109, which is a decrease of about 39% compared to the standard topology optimization. The eroded and dilated design both have an upscaling factor of 73. Keep in mind the intermediate design is the one which will ultimately be manufactured. The two other designs are just artificial designs in order to guarantee the minimum length scale required. The intermediate design is visible in more detail in figure 3.6. The figure shows that the

number of hinges has been reduced from 10 to 4. Furthermore, all hinges are larger than 5 mm in width, which was the objective of this more robust design.

From figure 3.6 it is visible the number of gray elements has increased. These elements will not disappear by increasing the smoothness parameter β , but can easily be removed in a post-processing phase.

3.3 Conclusions

In this chapter a more robust topology optimization algorithm was implemented. This algorithm, based on the paper of Wang et al. [39], is designed to ensure a minimum length scale. That way the occurrence of single node hinges is prevented. The algorithm works as it was intended to, resulting in no single node, or even single element hinges. This more robust design comes at a cost, with a decrease in upscaling factor of 39%. The design obtained by the standard topology optimization algorithm cannot be manufactured due to the small hinges. Therefore the significant decrease in upscaling factor is acceptable.

Chapter 4

Fibre in pure tension

As mentioned earlier in chapter 2 it could be of interest to ensure only axial tension forces are applied to the optical fibre, as it is designed to measure axial strain. This improvement, which is based on a suggestion made in the book of Bendsøe and Sigmund [3], will be discussed in this chapter. After the improvements are made, a new design will again be obtained and discussed. The alterations to the model discussed in this chapter are in reference to the standard topology optimization discussed in chapter 2 and not to the robust topology optimization of chapter 3. The combination of the alterations in chapter 3 and this chapter will be discussed in chapter 5.

4.1 Improvement of the model

4.1.1 Alterations to standard topology optimization

In order to ensure the fibre is only loaded in tension, prohibiting bending from occurring, rotation of the mounting platform of the fibre must be prevented. This means the top of the mounting platform must have the same horizontal translation as the bottom of the platform. The most efficient and effective way to implement this is to formulate this as a min-max problem, as suggested by Bendsøe and Sigmund.[3] This adjustment demands a small change in the design domain. Instead of one node for which the displacement will be optimized (figure 2.3), the displacements will be optimized for two nodes, one at the top and one at the bottom of the mounting platform. This is also visible in figure 4.1. The new objective function of the min-max problem is given in equation 4.1. Two different output-node selection vectors $l_{out,1}$ and $l_{out,2}$ are defined in this objective function.

$$\begin{aligned} \min_x : \max & \{l_{out,1}^T U + 100; l_{out,2}^T U + 100\} \\ \text{s.t.} : & KU = F \\ & : f_v = \frac{\sum \bar{x}_e}{V_0} \leq V^* \\ & : 0 \leq x \leq 1 \end{aligned} \tag{4.1}$$

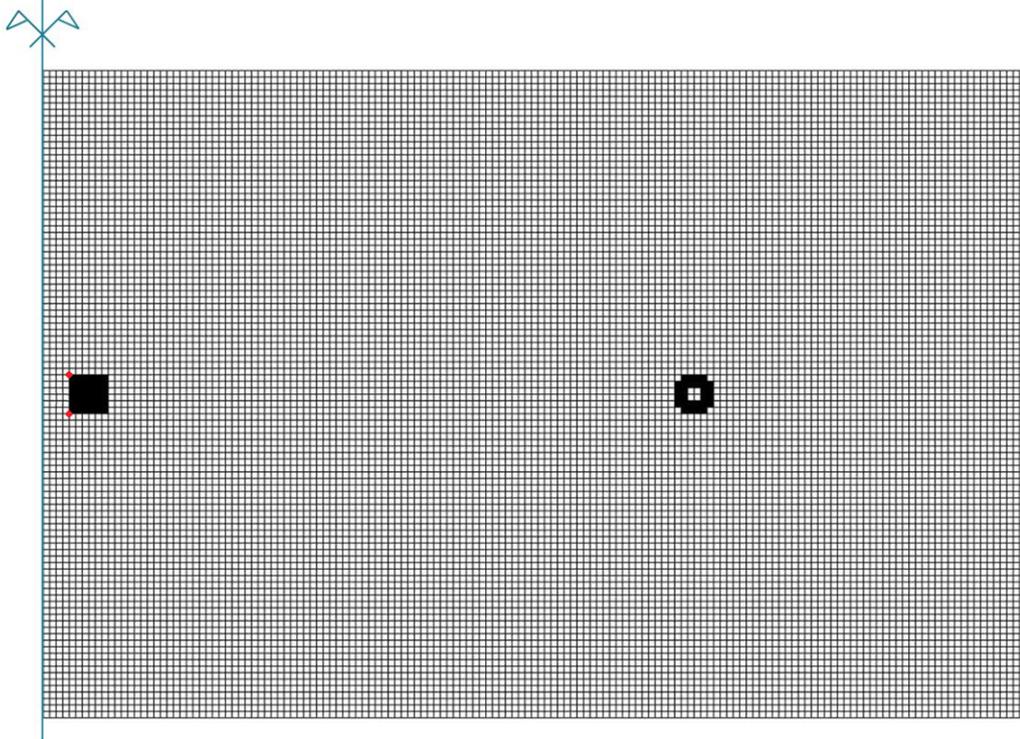


Figure 4.1: Altered design domain to ensure pure tension in the fibre.

The working principle of the min-max problem statement is very similar to the min-max problem in chapter 3. This will therefore not be discussed in more detail. There is however one important difference. Whereas the min-max problem in chapter 3 was used to optimize three different designs, the min-max problem used in this chapter only optimizes one design. The min-max problem is simply used to select for which output-node selection vector l_{out} the problem should be optimized.

4.1.2 Implementation in the MATLAB code

Some changes in the MATLAB code of the standard topology optimization are required in order to realize the proposed improvements in this chapter. The improved topology optimization algorithm can be found in appendix C.2. The first change is the definition of not one, but two output-node selection vectors $l_{out,1}$ and $l_{out,2}$. This is done in lines 73-74 of the algorithm. Next the objective function and the derivative has to be calculated for both output-node selection vectors. These are determined in lines 147-164. Lastly, the values of the objective function, as well as their derivatives will be entered in the constraints of the MMA algorithm. This is done in lines 116-121.

4.2 Obtained design

In this section the results obtained by the improved topology optimization algorithm will be discussed. As all parameters remain the same as was the case for the standard topology optimization algorithm, no parameters will be discussed like in previous chapters.

The algorithm takes 377 iterations to converge towards the final design. This is approximately the same number of iterations as the standard topology optimization algorithm, with 382 iterations. The time it takes to calculate the design has however increased from 360 s to 495 s or 8.25 minutes. This is an increase of about 38%, indicating each iteration takes longer to calculate. The obtained design is displayed in figure 4.2. The upscaling factor of this design is equal to 198. This is 10% higher

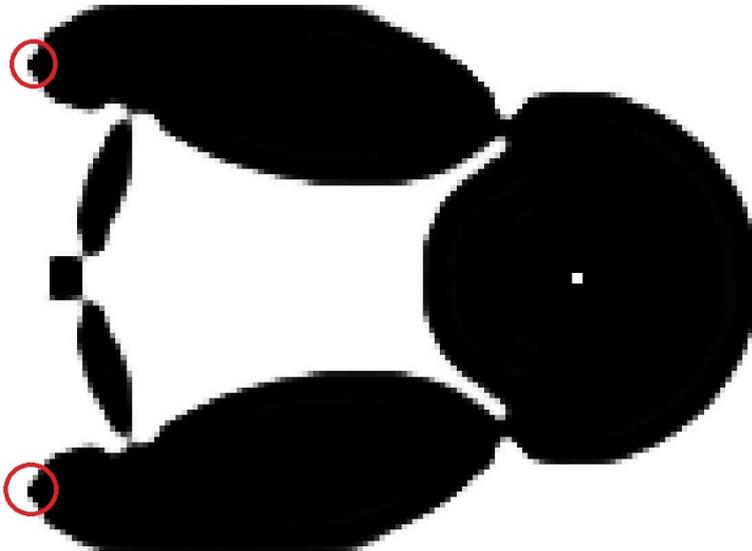


Figure 4.2: Design obtained by the algorithm which ensures pure tension in the optical fibre.

than the design obtained by the standard topology optimization algorithm. It clearly indicates the standard topology optimization algorithm was stuck in a local optimum. One could argue the design in figure 4.2 is also stuck in a local optimum. The fact is that if the hinges indicated in figure 4.2 are moved more towards the top and the bottom of the design domain respectively, the levers will increase in size, and ultimately the upscaling factor will increase.

The main goal is however to ensure the optical fibre is loaded in pure tension. This is the case in figure 4.2, but this is mainly due to the presence of a horizontal symmetry plane. When the mounting platform of the fibre is moved upwards to the

top of the design domain, this symmetry plane is not present any more. Nevertheless, figure 4.3 clearly shows the fibre is still loaded in pure tension. This is due to the hinged connection of the mounting platform to the rest of the transducer. This means the improved model works as it is supposed to. The design displayed in figure 4.3 has an upscaling factor of 149. Notice the difference with figure 2.10, where the mounting platform was connected rigidly.

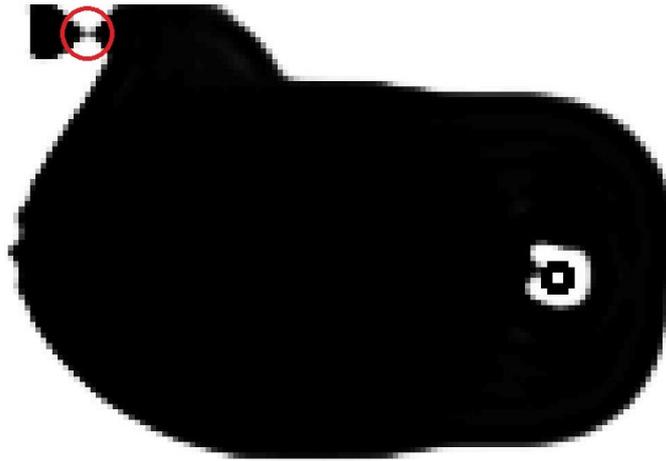


Figure 4.3: Design obtained with the mounting platform at the top of the design domain.

Figures 4.2 and 4.3 show that it is possible for single node or single element hinges to develop using this algorithm. This is due to the fact that no length scale control is implemented in this model. This will be solved in chapter 5.

4.3 Conclusions

In this chapter the model was changed to ensure only pure tension could occur in the optical fibre. It was shown that, by implementing a min-max approach, it was possible to prevent bending from occurring in the fibre. This is due to the formation of a hinged connection either at the two corners of the mounting platform for a symmetric design, or at the middle of the mounting platform for an asymmetric design, ensuring no rotation occurs. This proves the improved algorithm works as desired.

Chapter 5

Combination of length scale control and fibre in pure tension

In chapter 2 a first standard topology optimization algorithm was discussed and a couple of problems with the obtained design were raised. In chapter 3 and 4 both problems were addressed individually. In this chapter, the solutions to these problems will be combined, resulting in a final and most robust topology optimization algorithm.

5.1 Improvement of the model

To combine the two models developed in the previous chapters, the two min-max problems will have to be combined. On the one hand there is the min-max problem of the length scale control algorithm which optimizes one of three different designs. On the other hand there is the min-max problem of the fibre in pure tension, which selects one of two output-node selection vectors for which the design will be optimized. Combining these two min-max problems means that one of the three designs will be selected to be optimized for one of the two output-node selection vectors. The design domain displayed in figure 4.1 will be used in the combined algorithm. The objective function is given in equation 5.1.

$$\begin{aligned} \min_x : \max & \{l_{out,1}^T U_{ero} + 100; l_{out,2}^T U_{ero} + 100; l_{out,1}^T U_{int} + 100; \\ & l_{out,2}^T U_{int} + 100; l_{out,1}^T U_{dil} + 100; l_{out,2}^T U_{dil} + 100\} \\ s.t. : & KU_{ero} = F \\ & : KU_{int} = F \\ & : KU_{dil} = F \\ & : f_v = \frac{\sum \bar{x}_{e,ero} + \sum \bar{x}_{e,int} + \sum \bar{x}_{e,dil}}{3} \leq V_{mean}^* \\ & : 0 \leq x \leq 1 \end{aligned} \tag{5.1}$$

The implementation in the MATLAB code is very similar to previous chapters. The Matlab code can be found in appendix D.2. Due to the fact three designs will be optimized, the finite element analysis must be performed three times, once for every design. These calculations are performed in lines 171-183. Next the objective functions and their derivatives must be calculated. This is done six times per iteration, one time for each combination of design and output-node selection vector. These calculations are implemented in lines 184-239. Lastly, the six different values of the objective functions and their derivatives are implemented in the constraints of the MMA algorithm in lines 129-142.

5.2 Obtained design

The results of the most complex and robust topology optimization algorithm will be discussed. All parameters remain the same as in chapter 3.

The designs obtained after 410 iterations are displayed in figure 5.1. This is an increase of about 7% compared to the 382 iterations needed for the standard topology optimization. The calculation time has increased from 360 s to 1309 s or 21.8 minutes, which is an increase of 264%. The upscaling factor of the intermediate design is a little under 110, which is approximately the same as the upscaling factor obtained in chapter 3.

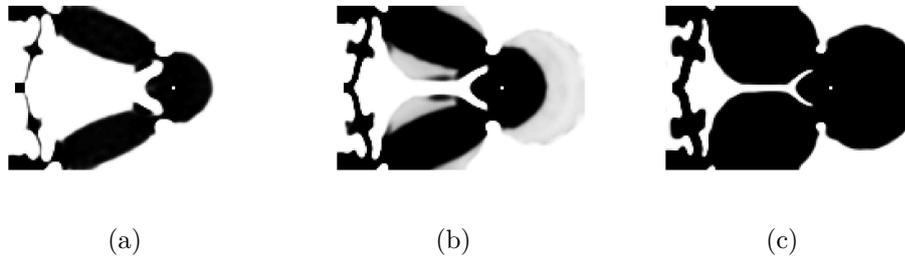


Figure 5.1: (a) Eroded, (b) intermediate and (c) dilated designs of the transducer obtained by the final, most robust topology optimization formulation.

There is however a drawback to this complex model. Not only does it take a longer time to calculate, it is also less stable than previous models, increasing the probability of gray elements and strange designs. This is visible in figure 5.2b, where a solid area is formed which is not connected to the other solid areas. Furthermore, the requirement of pure tension in the optical fibre cannot be guaranteed, due to the optimization of three different designs. This is for example the case in figure 5.2. It is clear that for the eroded design, the fibre will be loaded in pure tension. However, the size of the hinge has drastically increased in the intermediate design (see figure 5.3), as is required by the length scale control. This means rotation of the mounting

platform will occur and the fibre will experience bending.



Figure 5.2: (a) Eroded, (b) intermediate and (c) dilated designs of the transducer obtained by the final, most robust topology optimization formulation with the measuring section positioned at one quarter of the height of the transducer.

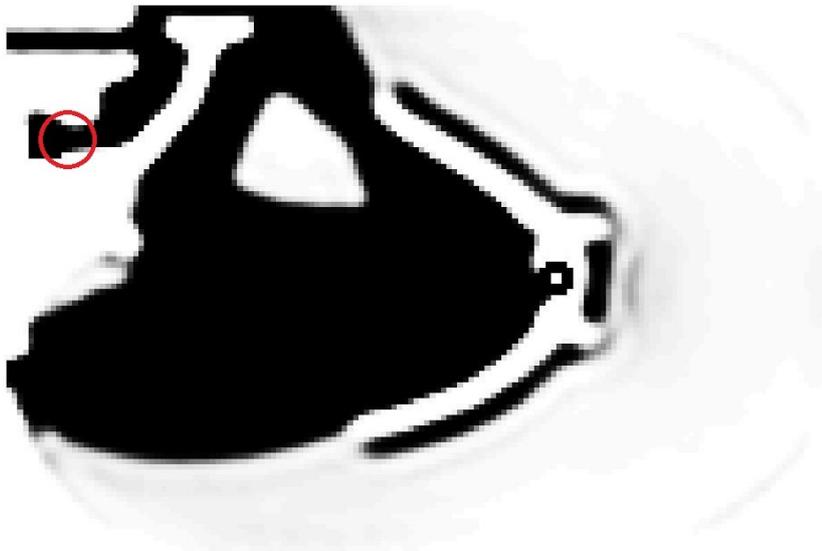


Figure 5.3: Detail of the hinged connection of the mounting platform.

It is obvious from these results that, while the algorithm might work in some cases, it has its limitations. Therefore it is important to always investigate whether or not the intermediate design meets all requirements. For the design in figure 5.1 this is the case, for the design in figure 5.2, it is not.

5.3 Conclusions

In this chapter the most complex and robust model was discussed. It is the combination of the models discussed in chapter 3 and 4. Although those models both

work fine individually, the combination turned out to be rather difficult. The most important problem is the requirement of pure tension in the optical fibre not being guaranteed in all cases. Therefore it is important to always analyse the resulting design of this algorithm in full detail, as it will not always perform as required. In some cases this model does work as it should and is very useful.

Chapter 6

Selection and discussion of the final designs

In chapters 2 to 5, four different topology optimization algorithms were discussed, ranging in complexity. No final design was selected however. In this chapter, the influence of the position of the measurement section will be discussed. Once this position is fixed, the final designs can be made and discussed.

6.1 Influence of the position of the measurement section

In the previous chapters, the vertical position of the measurement section was always defined at the middle of the transducer. It was already mentioned in chapter 2 this might not be the optimal position. In this section, the influence of the vertical position of the measurement section is discussed and a final decision on the location will be reached.

In this analysis, the vertical position of the measurement section will range from (counting from the top) element 5 to element 50 with an increment of 5 elements for each new optimization. This is from the top to the middle of the design domain. A measurement section at the bottom half of the design domain will not be discussed, as the resulting designs will be the same, only mirrored vertically. The design with the measurement section 5 elements from the bottom, will be the mirror image of the design with the measurement section 5 elements from the top. The upscaling factor of both designs will be identical. The result of the calculation for topology optimization with length scale control is given in figure 6.1.

One can see from this graph that the position of the measurement section has a significant influence on the obtained upscaling factor, with values ranging from 96 to 125. It can also be observed that only one position results in a smaller upscaling factor than the value obtained with the measurement section at the middle (represented by the vertical line). All other designs result in a larger upscaling of the strains.

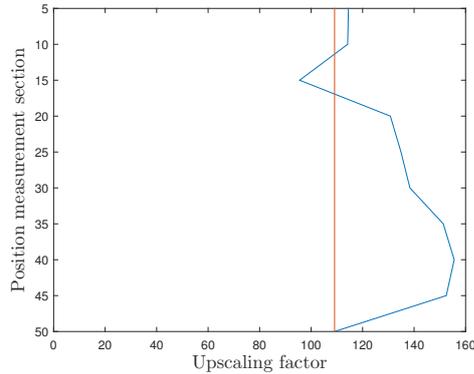


Figure 6.1: Influence of the vertical position of the measurement section on the upscaling factor when using the topology optimization with length scale control.

This result is only valid for the topology optimization algorithm without fibre in pure tension. When considering the most complex topology optimization algorithm, combining the length scale control and fibre in pure tension, a different result is found. This result is visible in figure 6.2.

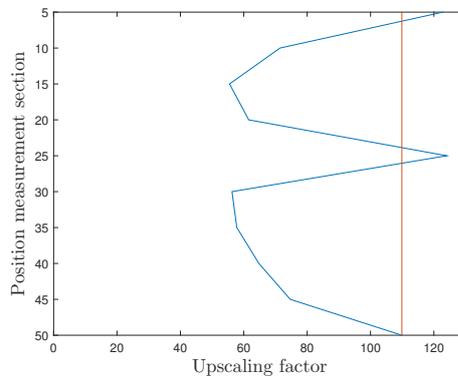


Figure 6.2: Influence of the vertical position of the measurement section on the upscaling factor when using the topology optimization with length scale control and fibre in pure tension.

This graph leads to a totally different conclusion. The strain upscaling when the measurement section is at the middle of the design domain (element 50) is large compared to other positions. Only for the measurement section at position 5 and 25 a larger upscaling factor is found. The values of the upscaling factor range from 56 to 124. A closer inspection of the designs obtained with measurement sections at position 5 and 25 reveal that these designs do not completely meet the constraint of pure tension in the fibre. This is visible in figure 6.3. As stated in chapter 5, one must

always check if this constraint has been met. This is clearly not the case for both designs. Therefore, the conclusion is reached the best position of the measurement section is at the middle of the transducer.



Figure 6.3: Intermediate designs with the measurement section at position (a) 5 and (b) 25.

6.2 Final designs

In the previous section, the conclusion was reached that the measurement section should be located in the middle of the design domain. Taking into account this decision, the final designs can be composed. As mentioned in chapter 2, two different designs can be made. A first design will result in tension in the fibre when the transducer itself is loaded in compression. This means the transducer will perform best when it is (quasi-)statically loaded in compression, as the optical fibre can only measure strains in tension. This is for example the case at the top of a simply supported beam. The second design will result in tension in the fibre when the transducer is loaded in tension as well. This means the transducer will perform best when it is (quasi-)statically loaded in tension. This is for example the case at the bottom of a simply supported beam. Both designs will be discussed in the next sections.

6.2.1 Design optimized in compression

In chapters 2 to 5, each of the four topology optimization algorithms was already executed for the design optimized in compression, with the measurement section located at the middle of the design domain. This means all four possible designs are already displayed in chapters 2 to 5. Two designs can be excluded, namely the design resulting from the standard topology optimization and the design obtained by the *fibre in pure tension* algorithm. Both designs feature single node hinges which cannot be manufactured. This leaves two designs. As mentioned in chapter 5, the algorithm with length scale control and the fibre in pure tension, results in approximately the same transducer design as the algorithm with only length scale control. The upscaling factor is roughly the same. However, calculation time is

significantly longer and, more importantly, slightly more gray elements occur when using the most complex algorithm. Therefore, the design obtained in chapter 3 by the algorithm with length scale control is preferred. The eroded, intermediate and dilated designs are displayed in figure 6.4. Keep in mind the intermediate design is the design which will be manufactured.

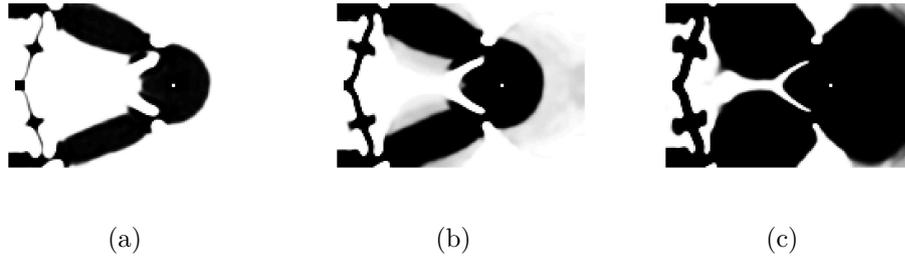


Figure 6.4: (a) Eroded, (b) intermediate and (c) dilated designs selected topology for the transducer optimized in compression.

The upscaling factor of the intermediate design is equal to 109. The design displayed in figure 6.4 does still require some manual post-processing. First all gray elements are removed. Secondly some black elements are removed manually. The elements which are removed contribute very little to the upscaling factor, while having a negative effect on the eigenfrequencies of the transducer. Therefore it is better to remove these elements. Lastly, the symmetry of the transducer is implemented, mirroring the design around the symmetry plane. The result of this post-processing is visible in figure 6.5.



Figure 6.5: Design for the transducer optimized in compression after post-processing.

In a last step, the edges of the elements will be partially smoothed, and a CAD drawing is made. It is from this CAD drawing, the transducer will be produced by laser cutting. This CAD drawing is also visible in figure 6.6, along with some important dimensions. Figure 6.7 shows the transducer optimized in compression after production.

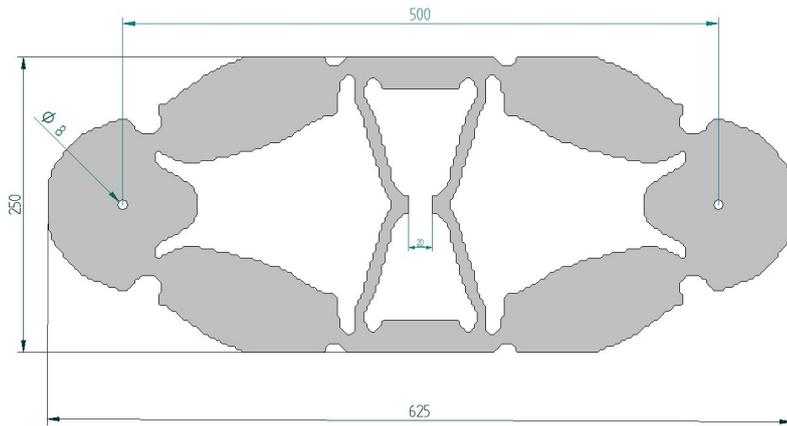


Figure 6.6: CAD drawing of the transducer optimized in compression.

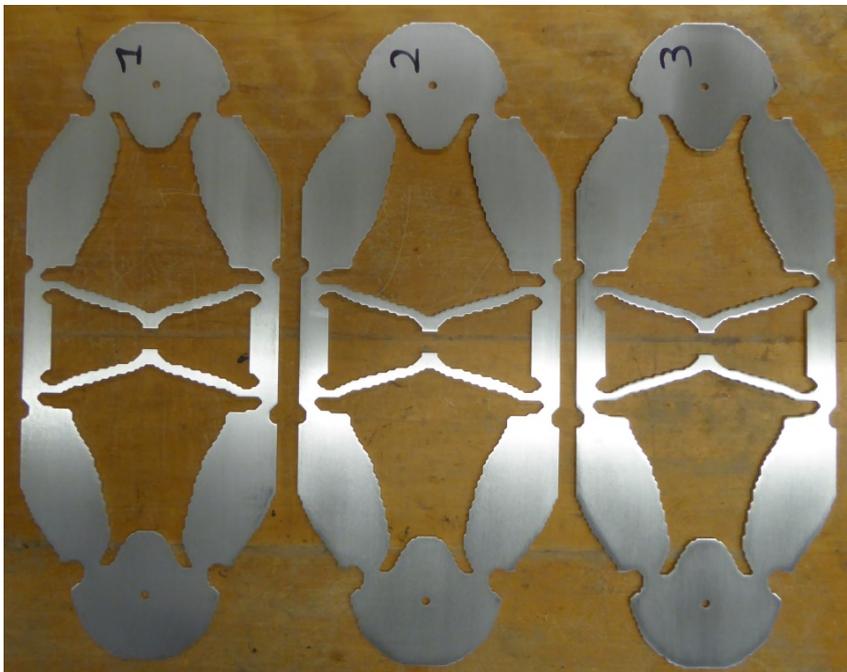


Figure 6.7: Transducer optimized in compression after production.

6.2.2 Design optimized in tension

The topology optimization algorithms can be slightly altered to calculate a design which performs optimal in tension. To achieve this, the load should be inverted to the negative direction. This is done by replacing line 70 (for standard topology optimization and length scale control) or line 71 (for fibre in pure tension and the combination) by:

```
70 F = sparse(inDof,1,-270/length(inDof),2*(nely+1)*(nelx+1),1);
```

The algorithm for the design in tension is very sensitive to the starting distribution of material over the design domain. While optimizing the design in compression, a homogeneous distribution of material over all elements was used (except in the passive or active elements). This start distribution does not work very well while optimizing the design in tension, as the optimization will rapidly converge to a local optimum. To solve this, a good material distribution is used to start the optimization. The best material distribution at hand is the design obtained while optimizing in compression. To make the design work in tension, some slight alterations to the material distributions must be made, flipping the direction of the connection arms of the mounting platform. This alteration is visible in figure 6.8.

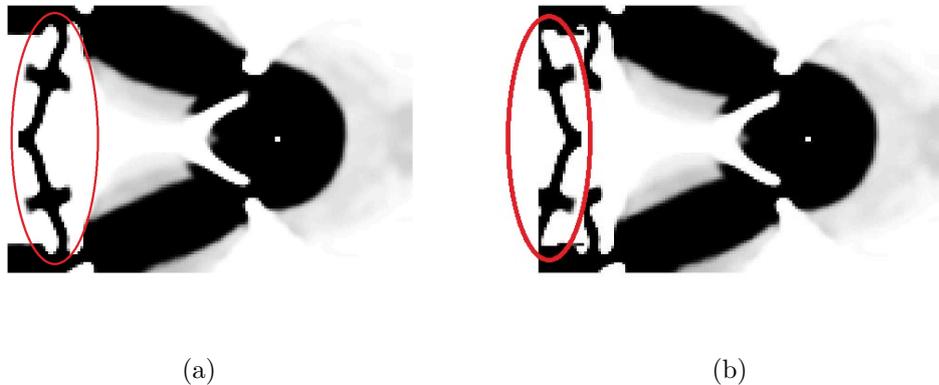


Figure 6.8: Starting material distribution (a) before and (b) after flipping the connection arms.

To implement this, the design obtained in compression is imported, the elements of the connection arms are selected and flipped around. From this material distribution, the optimization can start. In the MATLAB code, line 11 must be replaced by the following lines:

```

11 inputfile = strcat(savingDirectory, 'xPhysCompression.txt');
12 fileID = fopen(inputfile);
13 inputData = fscanf(fileID, '%f');
14 fclose(fileID);
15
16 z = reshape(inputData, nely, nelx);
17 flipMat = z(10:90, 4:20);
18 flipMat = fliplr(flipMat);
19 z(10:90, 1:17) = flipMat;

```

From the four different topology optimization algorithms, two of them result in a design which cannot be manufactured (standard topology optimization and fibre in pure tension) due to single node hinges. The designs obtained by the length scale control algorithm are displayed in figure 6.9. The intermediate design contains a lot of gray elements, which will be removed during post-processing. The intermediate design has an upscaling factor of 91. This algorithm results once again in a horizontally symmetric design, which ensures the fibre is loaded in pure tension. The most complex topology optimization algorithm discussed in this thesis results in the same design.

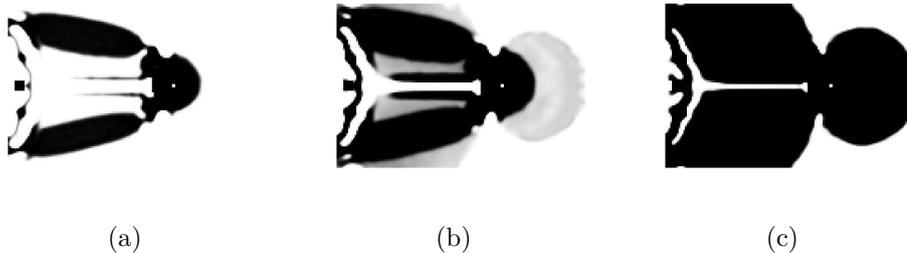


Figure 6.9: (a) Eroded, (b) intermediate and (c) dilated designs obtained using the algorithm with length scale control for the transducer optimized in tension.

Post-processing will be executed in a similar way as for the design in compression. The final design after post-processing is displayed in figure 6.10. The CAD drawing of the design optimized in tension is depicted in figure 6.11. Figure 6.12 shows the transducer optimized in tension after production.

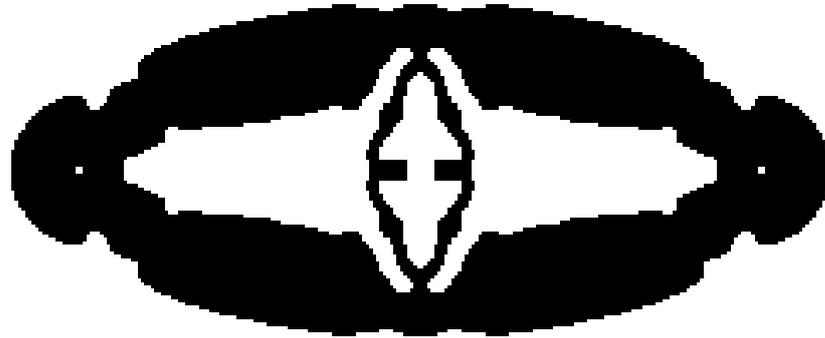


Figure 6.10: Design for the transducer optimized in tension after post-processing.

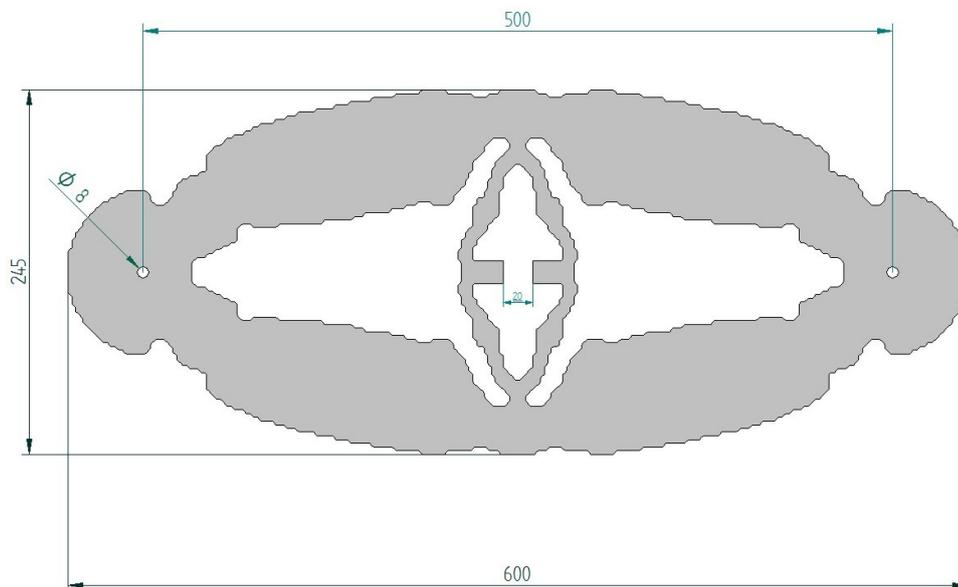


Figure 6.11: CAD drawing of the transducer optimized in tension.

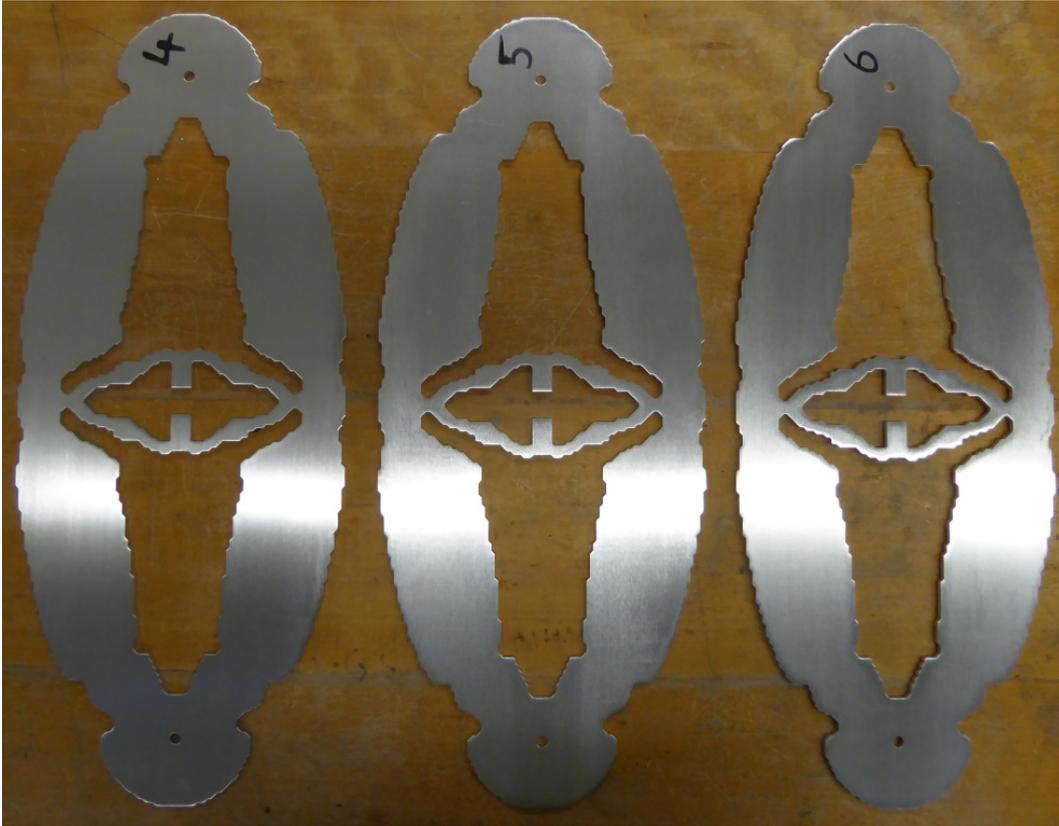


Figure 6.12: Transducer optimized in tension after production.

6.3 Conclusions

In this chapter, the influence of the position of the measurement section was discussed. The decision was made to place this section at the middle. Afterwards two final designs were selected, one performs optimal in compression, the other in tension. These designs will be produced and tested.

Chapter 7

Numerical validation

In previous chapters, a design was made using topology optimization in MATLAB. In this chapter, the behaviour of the designs will be checked in a more versatile finite element program: ANSYS. First the static results from MATLAB will be verified. Next, other important aspects of the design which are not included in the MATLAB algorithm will be checked: displacements and stresses in the transducer, dynamic behaviour and buckling loads. The log files for this numerical validation are available in appendix [E](#).

7.1 Static analysis

7.1.1 Verification of results

During the topology optimization an optimized distribution of material was obtained in MATLAB. This distribution is now loaded into ANSYS. During the loading of the design, some filtering of the densities is already performed. Elements with a density smaller than 0.5 will be assumed to be void. Elements with a larger density will be assumed to be solid. This is already a small alteration of the design, as intermediate values of the densities were still possible in MATLAB, even with the implementation of the black-and-white filter. Especially for the design optimized in tension this will change the performance because this design does still contain a significant amount of gray elements. However, the filtering in ANSYS better approximates the physical reality, as intermediate values of the densities do not exist.

Another difference between the implementation in ANSYS and MATLAB is the element used for the analysis. During topology optimization a very basic, four node element is used as defined in the 88 lines of code by Andreassen et al. [\[2\]](#). ANSYS is a more versatile, commercial finite element package, which contains more complex elements. During the numerical verification SHELL63 elements are used to model the transducer. It is also worth noting that during the numerical validation a geometrical non-linear analysis is performed, while during the topology optimization in MATLAB elastic linear behaviour was assumed.

7. NUMERICAL VALIDATION

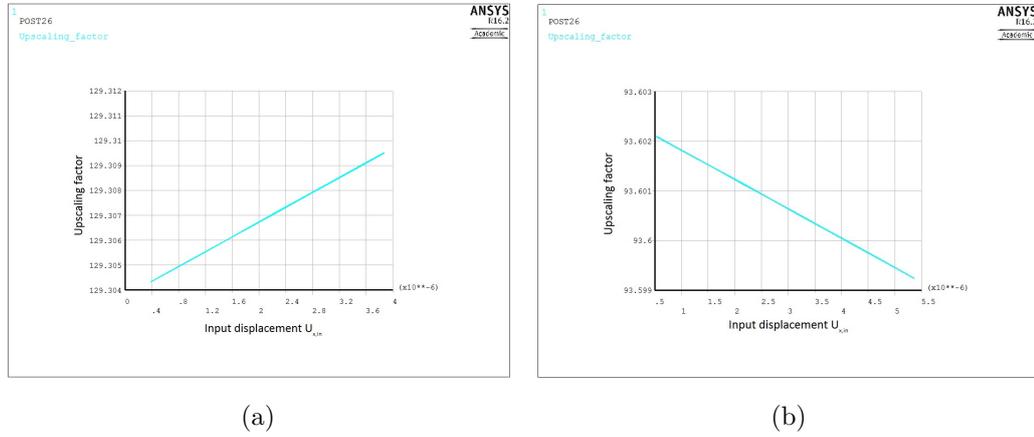


Figure 7.1: Upscaling factor calculated in ANSYS for the design in (a) compression and (b) tension.

Once the design is imported into ANSYS, the numerical verification of the results can be performed. Like during topology optimization, the upscaling factors of the two designs are calculated. In figure 7.1 the upscaling factor is displayed as a function of the imposed input displacement. One can see the upscaling factors do not remain completely constant for increasing input displacements. The difference is very small, with an increase of 0.005% for the design in compression and a decrease of 0.05% for the design in tension. This change is negligible. A more notable difference is the value of the upscaling factors. During topology optimization upscaling factors of 109 and 91 were found for the design in compression and tension respectively. The upscaling factors obtained during numerical validation are 129 for the design in compression, and 94 for the design in tension. This is an increase of 18% and 3%. These differences can be explained by the difference in complexity of the two models. As mentioned before, the use of the more complex SHELL63 elements as well as the geometrical non-linear analysis will have an influence on the performance of the transducer. The fact that the increase for the transducer optimized in tension (3%) is lower than the increase for the transducer optimized in compression (18%) can be explained by the larger percentage of gray elements in this design. These gray elements are filtered out when the design is imported into ANSYS, thereby influencing the performance of the design.

Although the numerical verification of the transducer yields slightly different upscaling factors, it can be concluded from this analysis that the working principle of the design remains the same. This numerical verification proves that the transducer does indeed enlarge the displacements which are imposed, resulting in an upscaling of the strains.

7.1.2 Post-processing

As mentioned before in chapter 6, some post-processing of the design is performed. The reason for this post-processing is the presence of elements which contribute very little to the upscaling factor, but have a negative effect on the eigenfrequencies of the transducers. Most of these elements are located on the connection arms of the mounting platforms, as shown in figure 7.2. The effect on the eigenfrequencies of the transducer will be discussed in the dynamic analysis of the transducer, further in this chapter. In this section it is already proven that the deleted elements do indeed contribute very little to the upscaling factor.

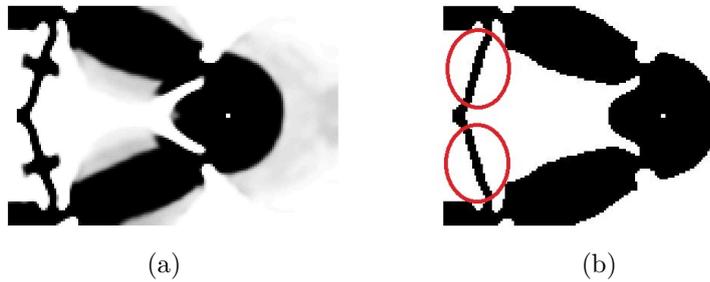


Figure 7.2: Design optimized in compression, (a) before and (b) after post-processing.

From the graphs in figure 7.3 one can see the upscaling factor is not really affected by the post-processing of the design. The upscaling factor of the design optimized in compression remains exactly the same at 129.3. The upscaling factor of the design optimized in tension does decrease a little from 93.6 to 93.3. This loss in upscaling factor is not significant, especially as this post-processing does improve the dynamic behaviour of the transducers.

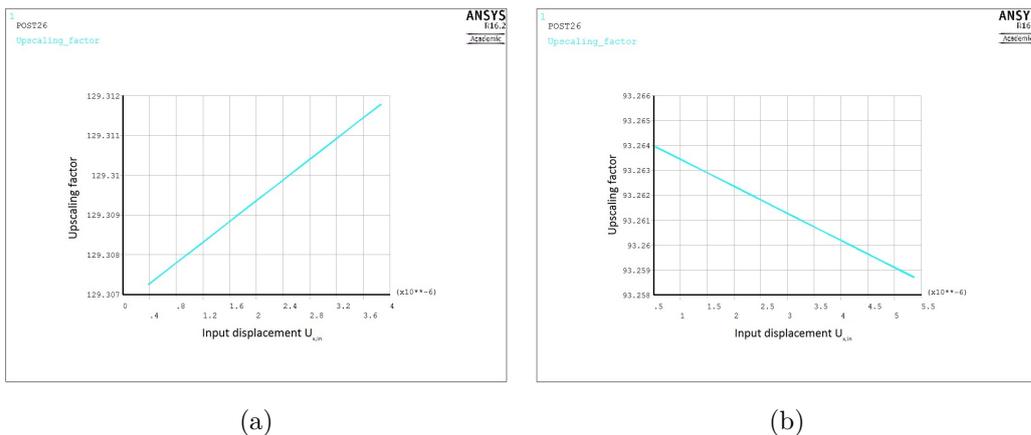


Figure 7.3: Upscaling factor after post-processing for the transducer optimized in (a) compression and (b) tension.

7.1.3 Displacements

The deformations of the design are of importance for the performance of the transducer. It is important that the transducer behaves linear elastically, especially at the location where the optical fibre is fixed. Geometric non-linearities may occur due to large displacements at the fixation point of the fibre. Particularly for the design optimized in compression, snap-through of the connection arms of the mounting platforms might be a problem.

In figures 7.4 and 7.5 the displacements of the design in compression and tension respectively are displayed. These figures are made with an input displacement which results in a strain of $2000 \mu\text{S}$ in the optical fibre. This is the advised maximum strain at which measurements can be executed with the fibres. These are therefore the maximum displacements which will occur under normal operational conditions. For the transducer optimized in compression, an input displacement of approximately -0.0038 mm will result in a strain of $2000 \mu\text{S}$ of tension in the fibre. For the transducer optimized in tension a strain of $2000 \mu\text{S}$ is reached by imposing a displacement of 0.0054 mm at the input nodes.

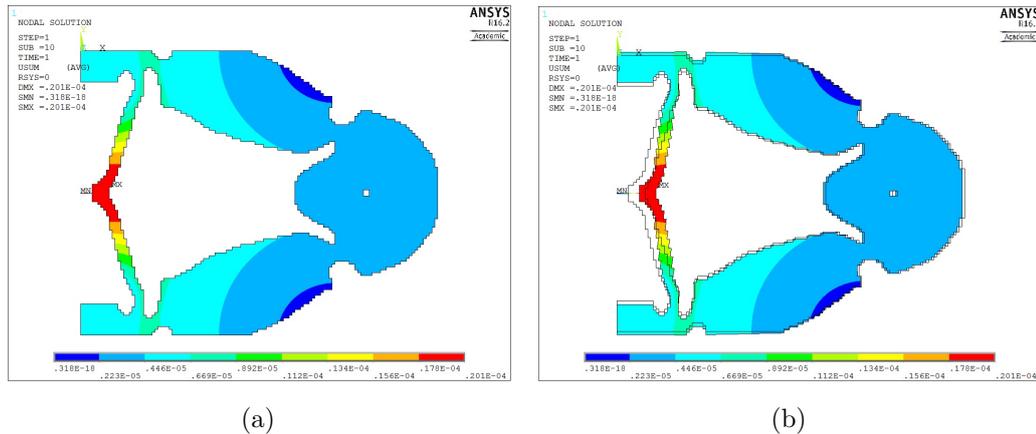


Figure 7.4: Displacements (a) in true scale and (b) enlarged 500 times for the design optimized in compression with $2000 \mu\text{S}$ in the optical fibre.

In subfigures 7.4a and 7.5a the true scale of the displacements is used. The displacements are very small, meaning the deformations of the transducers are not really visible on these figures. Therefore subfigures 7.4b and 7.5b are also added, in which the displacements are enlarged by a factor 500. This clarifies how the transducers deform under loading. The figures also show that the largest displacements take place at the mounting platforms. This is the behaviour the transducers are designed for.

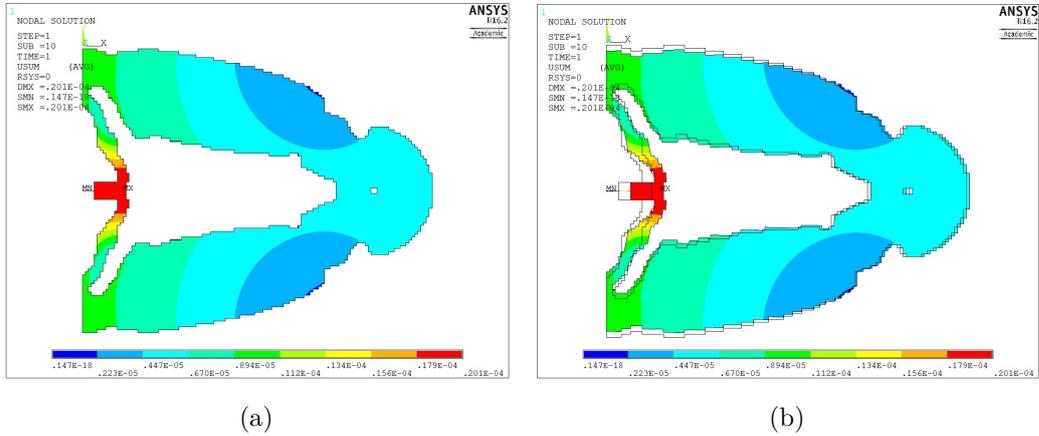


Figure 7.5: Displacements (a) in true scale and (b) enlarged 500 times for the design optimized in tension with $2000 \mu\text{S}$ in the optical fibre.

The fact that the displacements on true scale are not visible, already indicates that non-linearity due to large deformations will not be a problem. Due to these small displacements snap-through of the connection arms will also not be an issue. This is also confirmed by the graphs in figure 7.6. One can see that the displacement at the fixation point of the fibre increases linearly with increasing load, until a displacement of 0.02 mm is reached, which is equal to a strain of $2000 \mu\text{S}$ in the fibre. It can therefore be concluded that the transducer performs linearly under normal operational conditions.

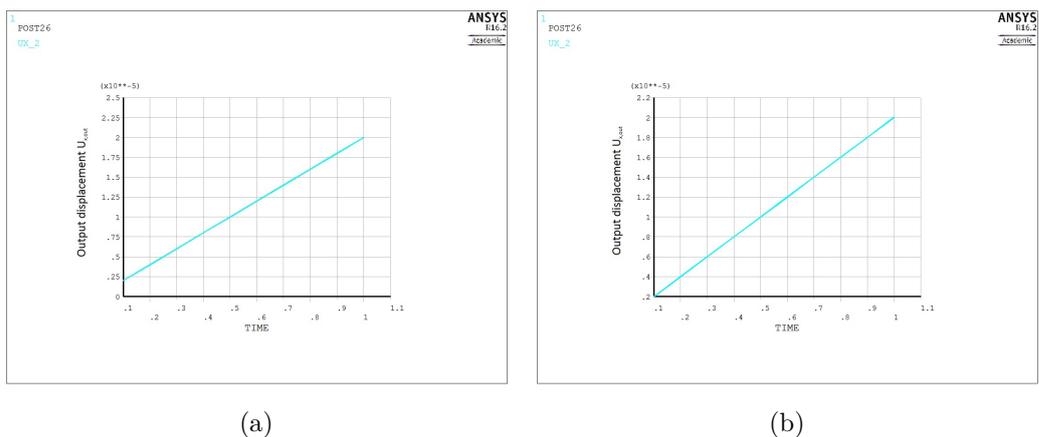


Figure 7.6: Graph of the displacement at the fixation node of the fibre as function of time for the design optimized in (a) compression and (b) tension.

7.1.4 Stresses

A last static check which is performed during the static analysis are the stresses in the transducers. In figures 7.7 and 7.8 the Von Mises stresses in the transducers optimized in compression and tension are displayed, again at the moment a strain of $2000 \mu\text{S}$ is present in the fibre. It is clear that the largest stress concentrations occur at the hinges and in the connection arms. This behaviour is as expected, as these parts have smaller dimensions, resulting in larger stresses. For the transducer optimized in compression, the highest stress is equal to 3.7 MPa. For the transducer optimized in tension, the highest stress is a bit higher with a value of 6.9 MPa. These values are still below the yield limit of aluminium which is equal to 95 MPa.[8] This means no plastic deformation of the transducer will occur under normal operational conditions.

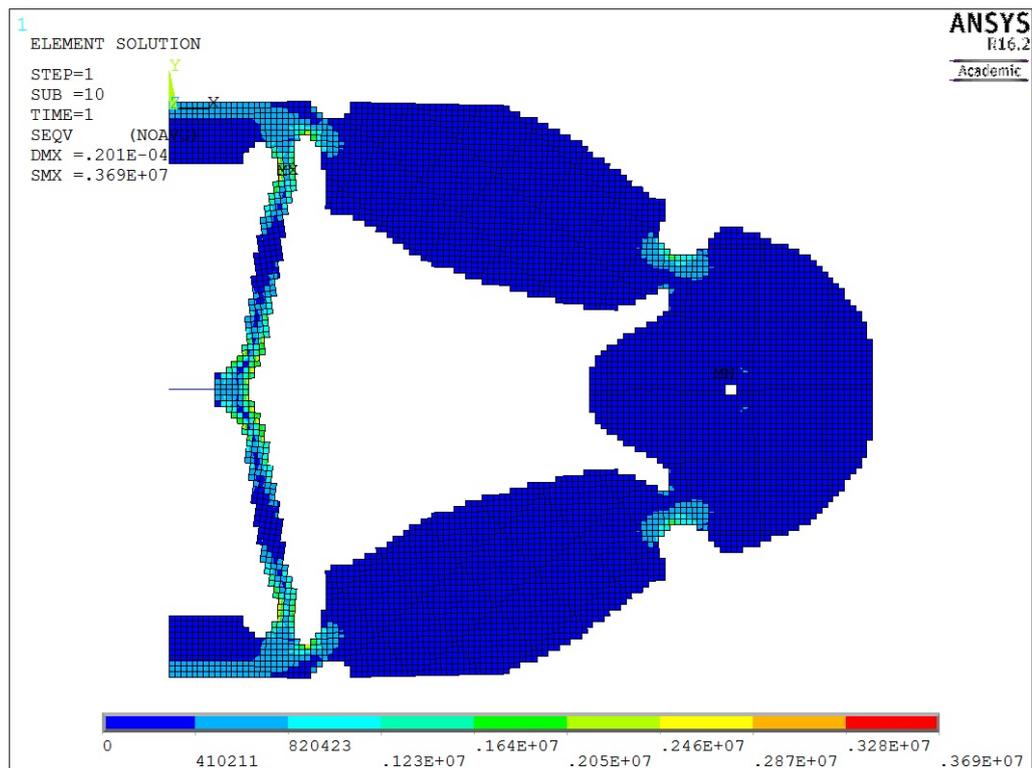


Figure 7.7: Von Mises stresses in the transducer optimised in compression with $2000 \mu\text{S}$ in the optical fibre.

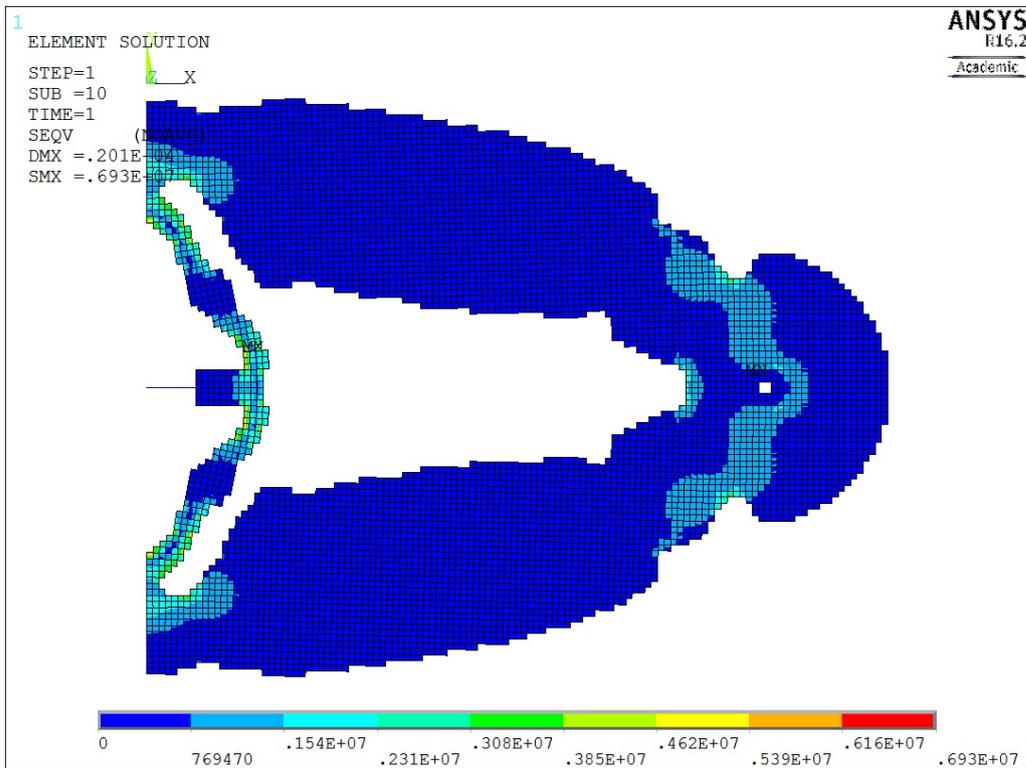


Figure 7.8: Von mises stresses in the transducer optimised in tension with $2000 \mu\text{S}$ in the optical fibre.

7.2 Dynamic analysis

In this section, the dynamic behaviour of the transducer is analysed. Since the transducer will be used for dynamic measurements as well, it is important that the eigenfrequencies of the transducer are high enough. Interference of these eigenfrequencies with the measurements, would limit the usability of the transducer. Therefore, all eigenfrequencies should remain above 50 Hz. If the eigenfrequencies would be lower than this value, dynamic behaviour of the transducer should be included in the topology optimization.

As mentioned before, some post-processing of the design was conducted, in order to remove elements which have a very small contribution to the upscaling factor, while having a negative impact on the eigenfrequencies of the transducer. A comparison is made between the designs before and after post-processing, to confirm the effect of the removed elements on the transducer. The first five eigenfrequencies in both cases are listed in tables [7.1](#) and [7.2](#).

Table 7.1: Eigenfrequencies of the design in compression.

Number	Eigenfrequencies [Hz]		
	Topology optimization	Post-processed	With brackets
1	66.2	69.5	60.7
2	78.3	79.7	79.7
3	268.2	285.2	180.2
4	415.1	414.8	414.8
5	477.3	486.9	461.0

Table 7.2: Eigenfrequencies of the design in tension.

Number	Eigenfrequencies [Hz]		
	Topology optimization	Post-processed	With brackets
1	70.8	73.7	67.2
2	83.3	85.0	85.0
3	243.2	321.6	221.1
4	255.4	431.0	431.0
5	318.3	470.2	467.7

In order to mount the optical fibre to the transducer, clamping brackets will be installed on the transducer. The mass of these brackets will also be modelled, as they have an influence on the eigenfrequencies. Each bracket weighs 20 g and measures 15 mm by 15 mm by 10 mm. On each mounting platform two of these brackets will be installed (see figure 7.9). The effect of these brackets on the values of the first five eigenfrequencies is also listed in tables 7.1 and 7.2. The shapes of the eigenmodes with clamping brackets are displayed in figures 7.10 and 7.11 for the design in compression and tension respectively.

The eigenfrequencies are quite high, ranging from 60.7 Hz to 461.0 Hz for the design in compression and 67.2 Hz to 467.7 Hz for the design in tension. This means the eigenfrequencies of both designs are above the required 50 Hz. The eigenfrequencies listed in tables 7.1 and 7.2 show that the post-processing of the design does indeed raise the eigenfrequencies of the transducers. For the lowest eigenfrequencies this increase is just a few Hz. For the higher eigenfrequencies the increase is even larger.

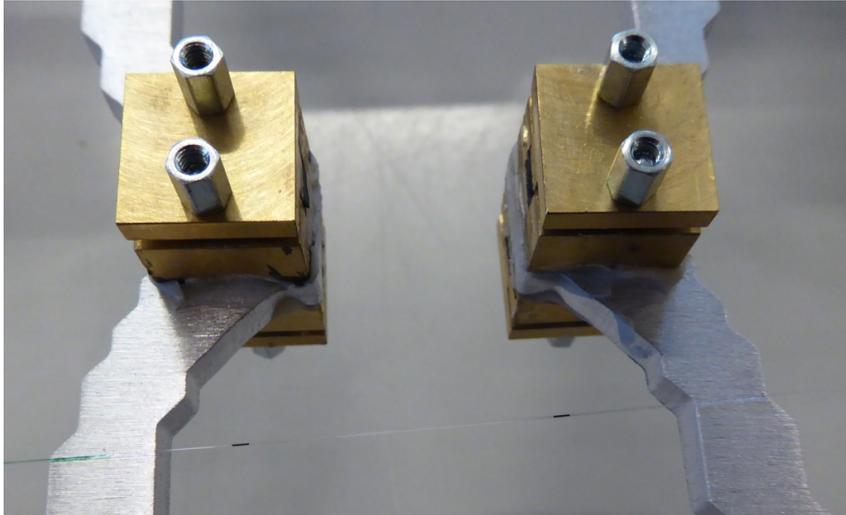


Figure 7.9: Placement of the four clamping brackets on the two mounting platforms of the transducer: two at the front and two at the back of the transducer.

All eigenmodes in tables 7.1 and 7.2 are symmetric eigenmodes. Asymmetric eigenmodes cannot be calculated due to the symmetric boundary condition in the ANSYS model. However, the first asymmetric eigenmode has a larger eigenfrequency than the first symmetric eigenmode. Therefore, the asymmetric eigenmodes will not pose a problem.

The order of the different mode shapes is very similar in both cases. Eigenmode one is in both cases the first bending mode of the transducer, mode two is the first torsion mode. Mode three is the first bending mode of the connection arms of the mounting platforms. Mode four is a combination of bending and torsion of the transducer. Mode five is different for both designs. For the transducer in compression, mode five is again a bending mode of the transducer. For the transducer in tension mode five is a combination of bending and torsion.

From the dynamic analysis in this section, it can be concluded that the eigenfrequencies of the designs are sufficiently high. Therefore, no dynamic behaviour of the transducer is included in the topology optimization.

7. NUMERICAL VALIDATION

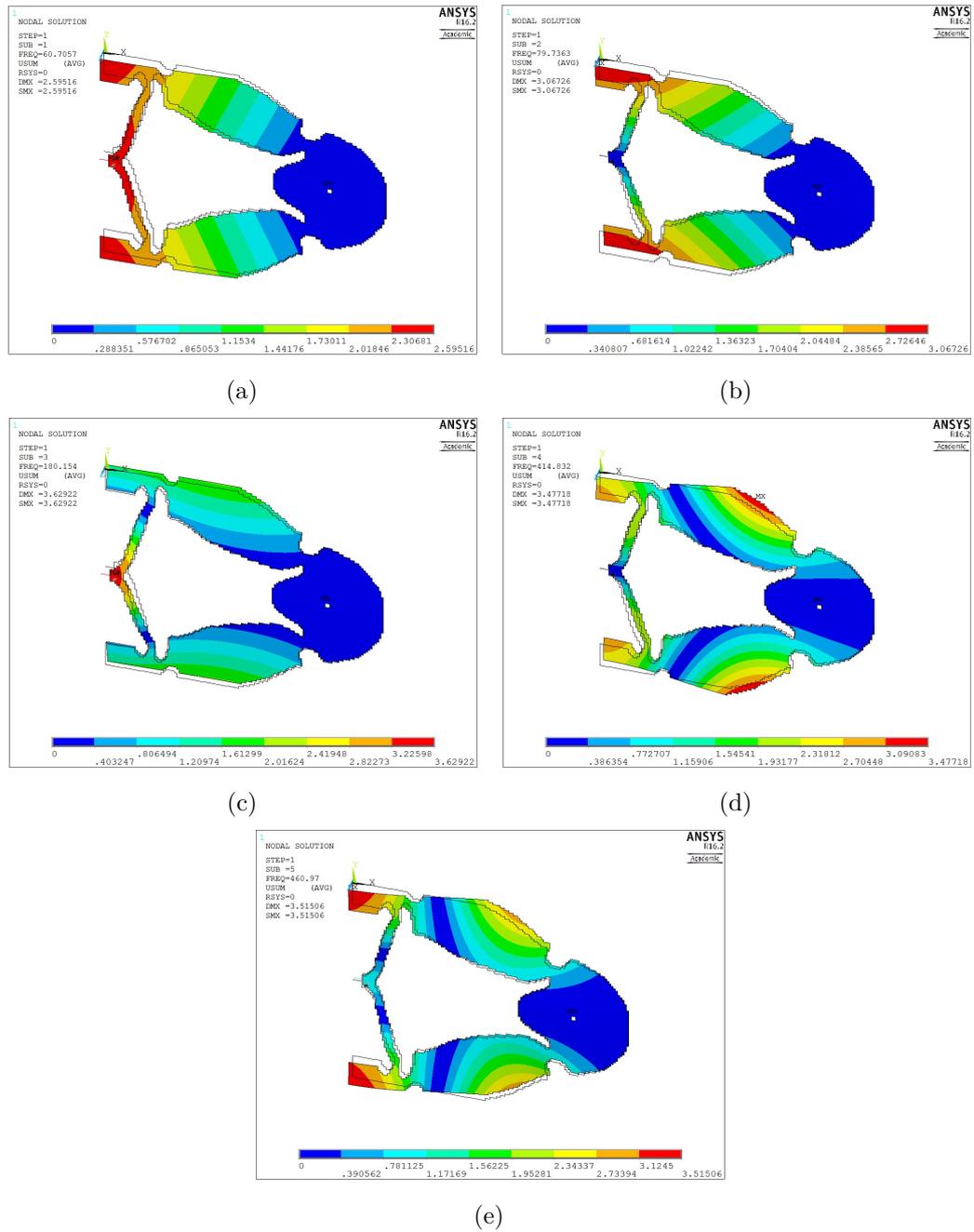


Figure 7.10: Mode shapes of eigenmodes (a) one (60.7 Hz), (b) two (79.7 Hz), (c) three (180.2 Hz), (d) four (414.8 Hz) and (e) five (461.0 Hz) for the transducer design in compression.

7.2. Dynamic analysis

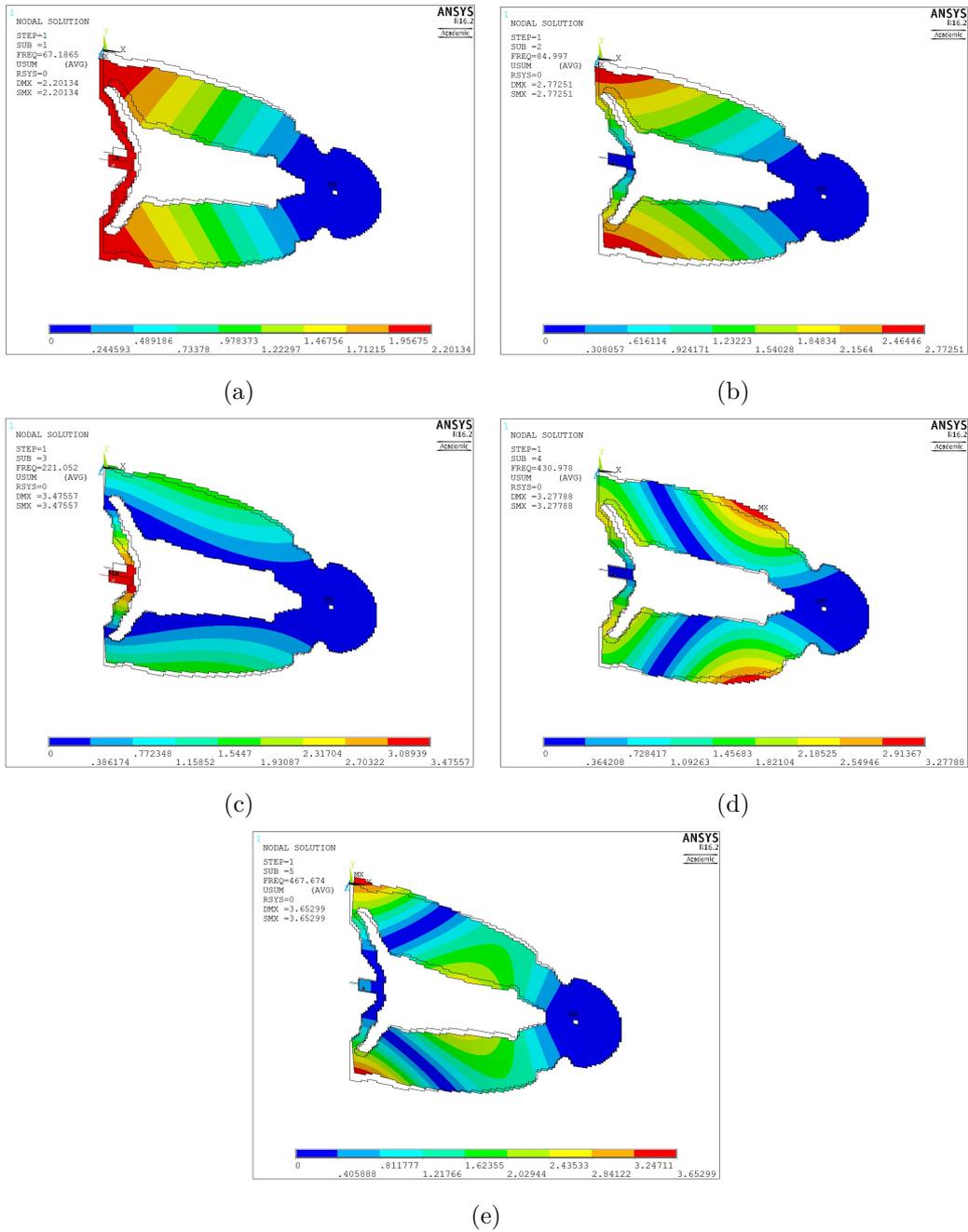


Figure 7.11: Mode shapes of eigenmodes (a) one (67.2 Hz), (b) two (85.0 Hz), (c) three (221.1 Hz), (d) four (431.0 Hz) and (e) five (467.7 Hz) for the transducer design in tension.

7.3 Buckling analysis

It is of importance that the transducer will not buckle when a load is applied. To ensure this does not happen, an eigenvalue or linear buckling analysis is performed in ANSYS. The calculation will result in a load multiplier which indicates how much margin is left before the transducer starts to buckle. A load multiplier of 10 for example indicates that the applied load can be multiplied by a factor ten before buckling occurs. Note that the linear buckling analysis calculates the theoretical buckling strength of an ideal linear elastic structure. No imperfections, plastic behaviour or large displacements are taken into account. This will typically result in an overestimation of the buckling strength. To incorporate all these effects, a more complex non-linear buckling analysis is recommended. However, the linear buckling analysis provides a good first estimation of the buckling loads.

As mentioned before, it is advisable not to load the optical fibre with more than 2000 μS . Therefore, a buckling analysis of both transducers will be performed with the strain in the fibre equal to this maximal value of 2000 μS . The first five load multipliers are listed in tables 7.3 and 7.4 for the design in compression and tension respectively. Note that some load multipliers are negative. This indicates that the applied force should be inverted in order for the buckling to occur. The analysis has been performed for the designs obtained using topology optimization and for the post-processed designs. The load multipliers listed in tables 7.3 and 7.4 show that post-processing has some influence on the buckling load. This influence is however small and will not pose a problem.

The design optimized in compression is loaded in compression under normal operational conditions. One would therefore expect this design to be sensitive to buckling. Table 7.3 shows however a load multiplier of 75.0 as the smallest (absolute) value. This buckling mode is the out-of-plane buckling mode of the transducer, which one would indeed expect to be the lowest buckling mode. This mode is displayed in figure 7.12b. Since the load multiplier has a very high value, it is clear that buckling will not pose a problem for the design optimized in compression. Note that one load multiplier has a negative value. This is the buckling mode of the connection arms of the mounting platform, visible in figure 7.12a. Since this negative value indicates that the design has to be loaded in tension for this buckling mode to occur, one can conclude that this buckling mode will never occur. In conclusion, buckling of this transducer design will not occur under normal operational conditions.

The transducer optimized in tension is loaded in tension under normal operational conditions. For this reason the design will be less sensitive to buckling than the other design. This suspicion is confirmed by the load multipliers listed in table 7.13. Four of the five first load multipliers have a negative value, meaning they will not occur under normal operational conditions. The only positive load multiplier obtained is equal to 188.0. This is the buckling mode of the connection arms of the mounting platform, as can be seen in figure 7.13e. This is the only part of the transducer which

will be sensitive to buckling. The load multiplier is however very high, meaning buckling will pose no problem.

All buckling modes in tables 7.3 and 7.4 are symmetric modes. Asymmetric buckling modes cannot be calculated due to the symmetric boundary condition in the ANSYS model. The first asymmetric buckling mode has however a larger load multiplier than the first symmetric buckling mode. Therefore, the asymmetric buckling modes will not pose a problem.

Table 7.3: Load multipliers for buckling of the design in compression.

Shape number	Load multipliers for buckling	
	Topology optimization	Post-processed
1	-287.9	-283.8
2	75.5	75.0
3	203.6	193.5
4	471.4	464.7
5	520.4	508.5

Table 7.4: Load multipliers for buckling of the design in tension.

Shape number	Load multipliers for buckling	
	Topology optimization	Post-processed
1	-225.4	-222.5
2	-212.3	-212.6
3	-124.9	-115.1
4	-41.7	-42.1
5	181.4	188.0

From the analysis above, it can be concluded that buckling does not pose a problem for either designs of the transducer. Therefore, no buckling of the transducer is included in the topology optimization.

7. NUMERICAL VALIDATION

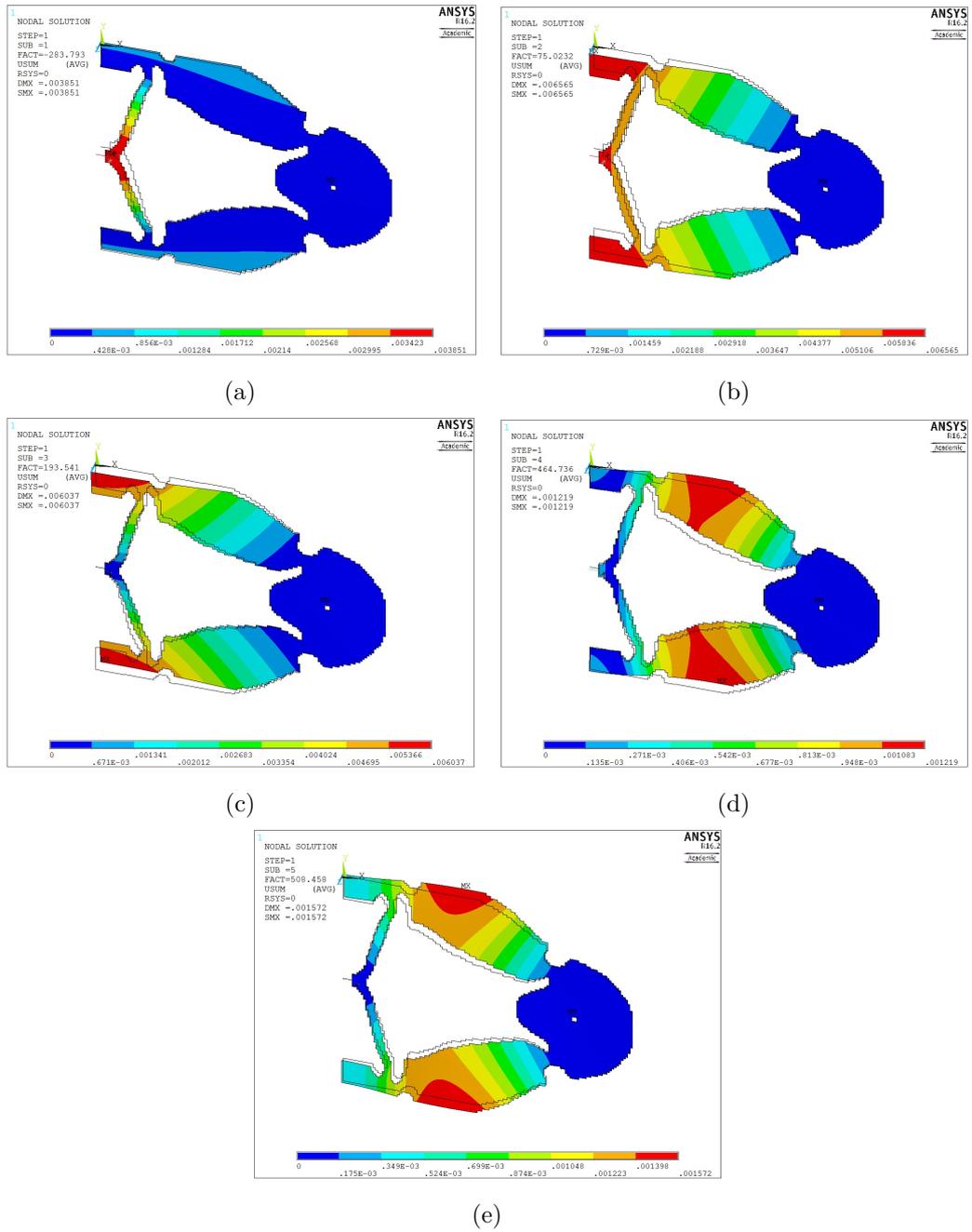


Figure 7.12: Buckling shapes (a) one (-283.8), (b) two (75.0), (c) three (193.5), (d) four (464.7) and (e) five (508.5) for the transducer design in compression.

7.3. Buckling analysis

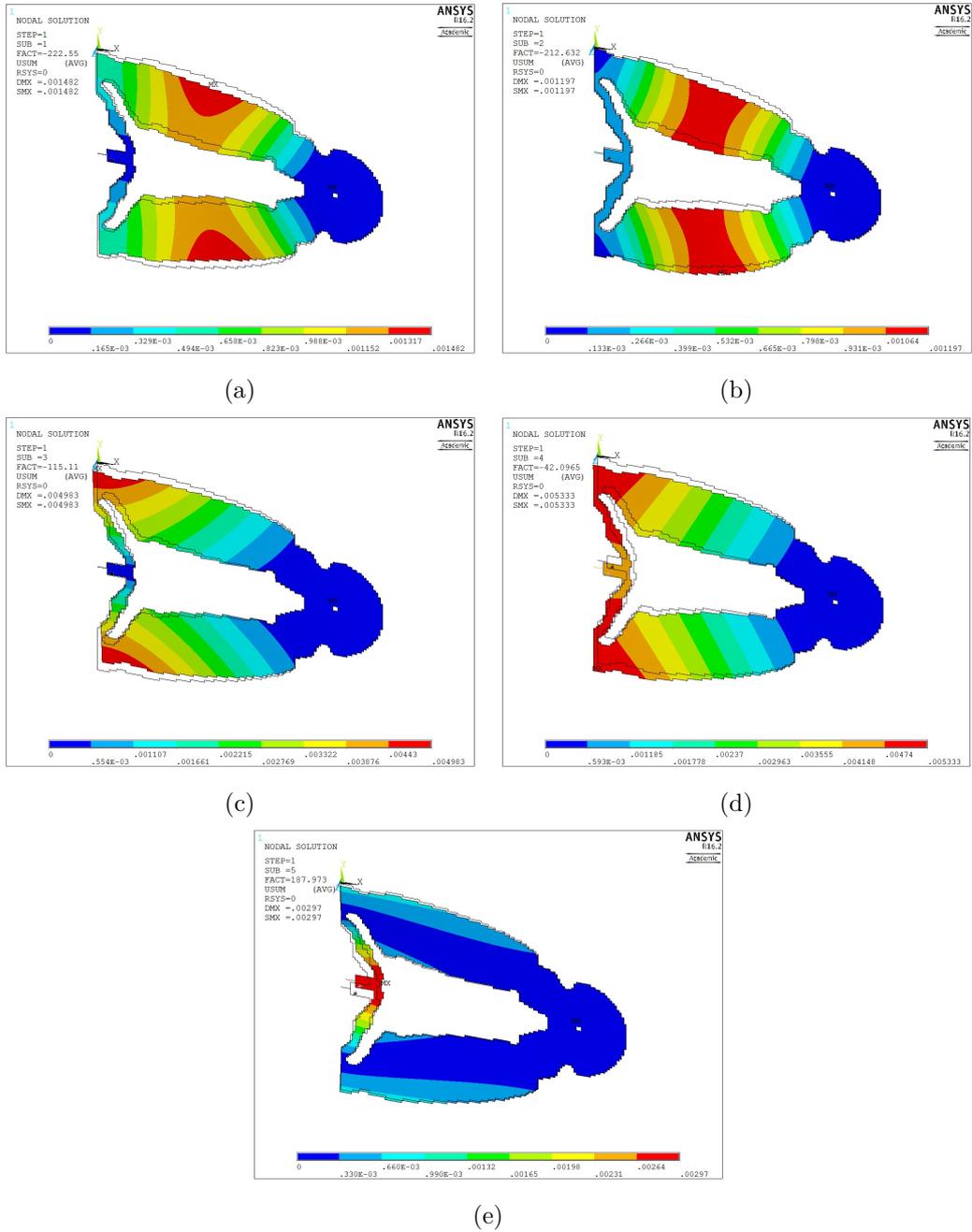


Figure 7.13: Buckling shapes (a) one (-222.5), (b) two (-212.6), (c) three (-115.1), (d) four (-42.1) and (e) five (188.0) for the transducer design in tension.

7.4 Conclusions

During this chapter a validation of the designs was performed. First of all the results from the topology optimization were checked. The upscaling factors obtained in ANSYS are higher than the ones previously calculated in MATLAB. However, this was explained by the increase of complexity of the model in ANSYS. Next the displacements and stresses in the transducers were checked under normal operational conditions. Both did not present a problem. Thirdly, the dynamic behaviour of the transducers was checked. The values of the eigenfrequencies did not pose a problem, which justifies the decision not to include them in the topology optimization. The same can be said for the decision not to include buckling in the topology optimization, as the transducers proved not to be sensitive to buckling.

In conclusion it is clear that the performance of the transducer meets all imposed requirements.

Chapter 8

Experimental validation

In a final stage of this thesis, the designs of the transducers will be tested in the structural mechanics laboratory. It will be checked if the transducer performs as predicted. First the preparation of the strain upscaling sensor package will be discussed. Next, static and dynamic experiments will be performed. The results of these experiments will be discussed, in order to draw some conclusions. Lastly, future experiments in the development of the strain upscaling sensor package will be proposed.

8.1 Introduction to fibre optic sensing using fibre Bragg gratings

Measuring strains using optical fibres has increased in popularity. During this thesis the fibre Bragg grating (FBG) system will be used. This system uses an optical fibre which, at selected places, contains Bragg gratings. These Bragg gratings will reflect one specific wavelength spectrum, while other wavelengths will be transmitted. The fibre optic sensing method utilizes the reflective properties of the FBG. A light spectrum between 1528 and 1568 nm [10] will be passed through the fibre. When this light spectrum reaches an FBG, the wavelength of the FBG will be reflected. The rest of the spectrum will be transmitted and will exit the optical fibre at the end. This is also visualised in figure 8.1. The reflected spectrum will then be analysed, as it will contain valuable information about the strain or temperature change in the fibre. When the optical fibre strains or changes temperature at the location of the FBG, the wavelength which is reflected by this FBG changes. By tracking the wavelength of the peak of reflected light, the strain or temperature change of the FBG can be determined. When the FBG is subjected to a length change of $1 \mu\text{S}$, the peak of the reflected spectrum will shift 1.2 pm. The fibre also reacts to temperature changes: a change of $1 \text{ }^\circ\text{C}$ is equal to a strain of $0.5 \mu\text{S}$ or a shift in wavelength of 0.6 pm of the reflected peak.[11]

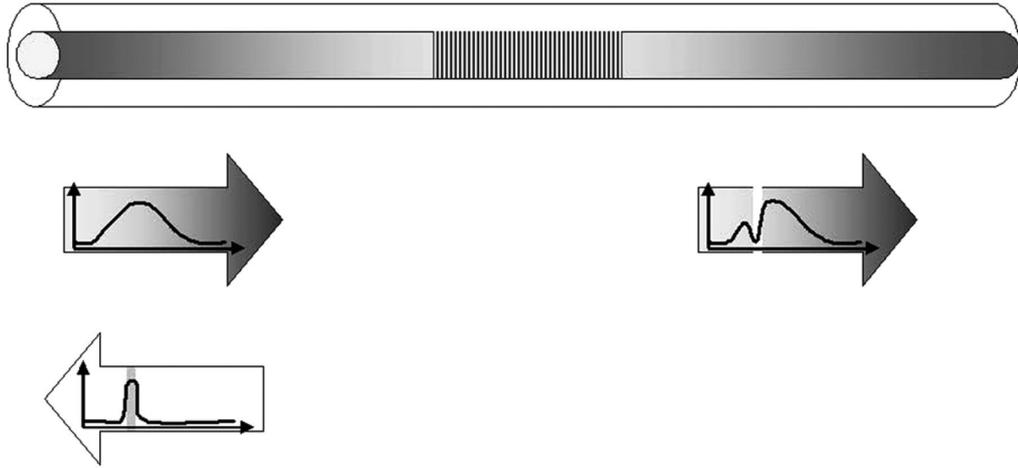


Figure 8.1: Working principle of fibre Bragg gratings [13].

Note that the transmitted spectrum can still be used by other FBGs further in the optical fibre, as long as these FBGs reflect different wavelengths of light. This makes it possible to incorporate multiple FBGs and therefore to perform multiple measurements at different locations with just one optical fibre. However, it is important that the peaks of light reflected by the FBGs do not overlap, as this would make measurements impossible. An example of a reflected light spectrum for a fibre with four different FBGs is displayed in figure 8.2.

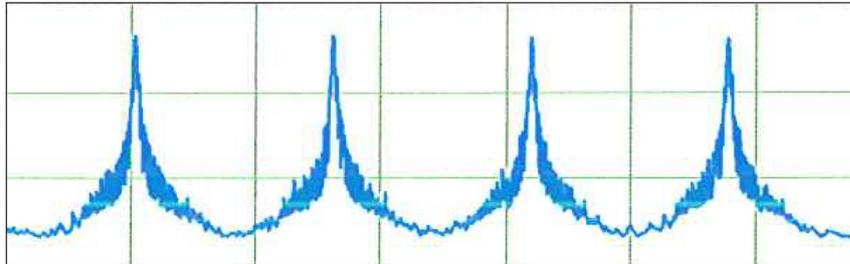


Figure 8.2: Example of a reflected light spectrum of a fibre containing four FBGs.

8.2 Preparation

Throughout this thesis, the design of the transducer itself was discussed. The total strain upscaling sensor package consists of other components as well. Apart from the transducer, the package uses clamping brackets, an optical fibre and an interrogator. The different components are presented in figure 8.3.

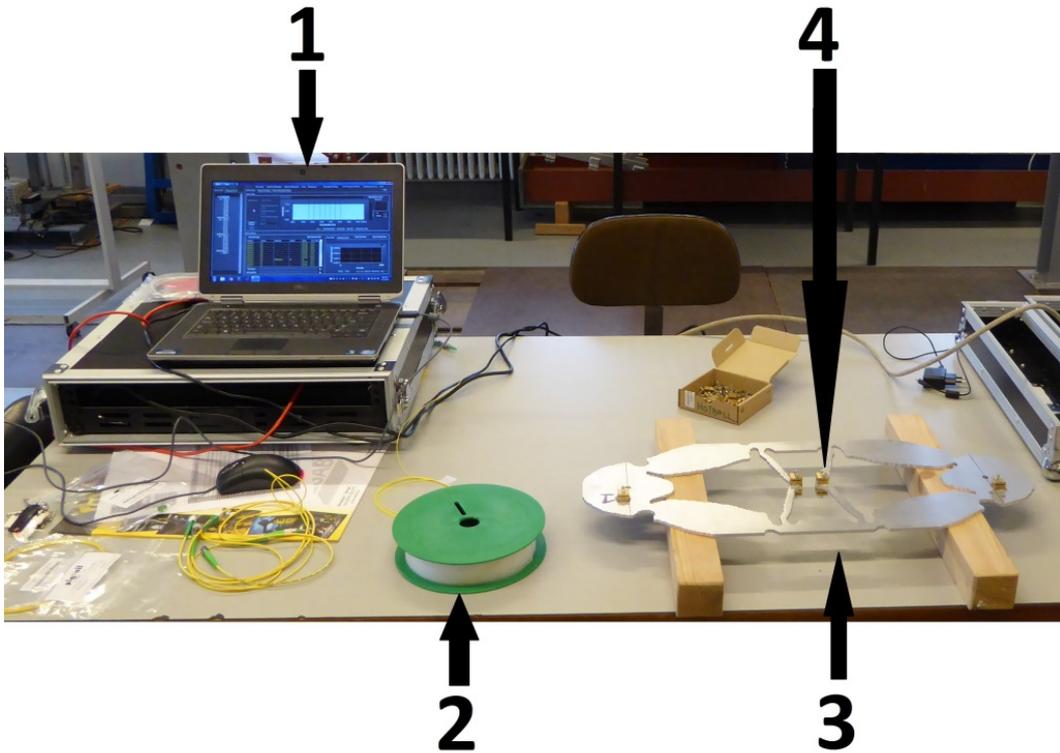


Figure 8.3: Different components of the strain upscaling sensor package: (1) interrogator, (2) optical fibre, (3) transducer and (4) clamping brackets.

The experiments are carried out with an optical fibre with four fibre Bragg gratings on each transducer. This means four different properties can be measured by one fibre: one FBG for reference strain, one FBG for temperature measurements and two FBGs to measure the upscaled strains. Two such optical fibres were used during the experiments. The wavelength and function of each FBG is listed in tables 8.1 and 8.2. More detailed information on the properties of each fibre can be found in appendix F.

Table 8.1: Properties of fibre one.

FBG	Function	Initial wavelength [nm]
1	Reference strain	1535.16
2	Temperature	1543.07
3	Upscaled strain	1551.05
4	Upscaled strain	1558.91

8. EXPERIMENTAL VALIDATION

Table 8.2: Properties of fibre two.

FBG	Function	Initial wavelength [nm]
1	Reference strain	1535.09
2	Temperature	1543.14
3	Upscaled strain	1551.08
4	Upscaled strain	1559.01

In order to install the fibre, several clamping brackets are glued to the transducer. The three FBGs used for measuring strain, will be installed using these clamping brackets. The layout of the FBGs and their corresponding clamping brackets is visible in figure 8.4.

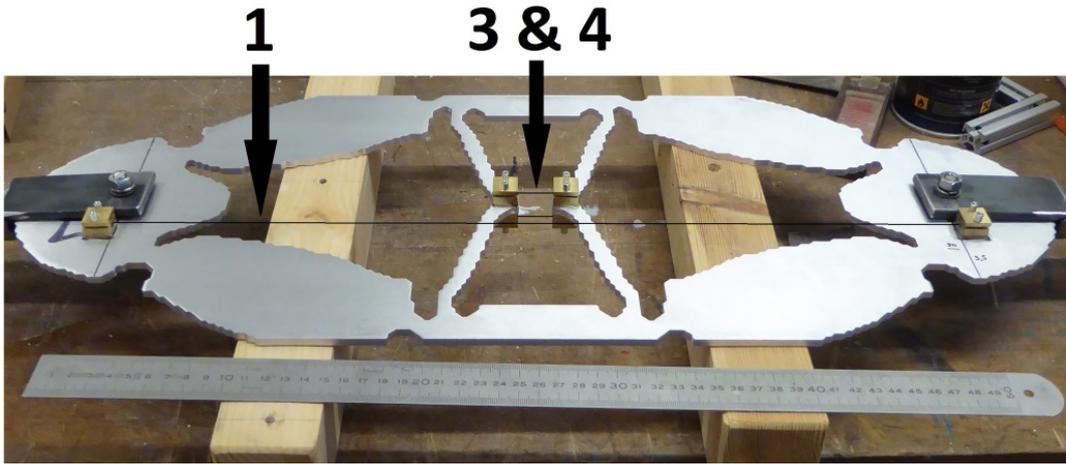


Figure 8.4: Configuration of the FBGs: (1) reference strain, (3) upscaled strain and (4) upscaled strain.

In this configuration of FBGs, a total of six clamping brackets are needed: four for installing the FBGs measuring the upscaled strains and two for installing the FBG measuring the reference strain.

The reference FBG is used to measure the total strains between the two bolt connections. In order to do this, the FBG will be placed eccentric from the centreline of the transducer, with a distance of 500 mm between the two clamping brackets. The eccentricity of the clamping brackets is visible in figure 8.5. Note that the two clamping brackets installed for the FBG measuring the reference strain were not modelled while calculating the eigenfrequencies of the transducer in chapter 7. The reason for this is that they will only be installed while developing the sensor package in the laboratory. Once the strain upscaling sensor package is fully developed and

ready for application in the field, the reference FBG will be redundant and the clamping brackets will not be installed any more.

Two FBGs will be used to measure the upscaled strains. One FBG is installed at the front of the transducer, the other at the back. By using this configuration, it is possible to check if any out-of-plane bending occurs. This could be an indication of eccentric loads and the presence of bending moments at the bolt connections. The out-of-plane bending can be recognized by different strain values of the two FBGs measuring upscaled strains.

The FBG measuring temperature changes will be taped onto the transducer directly. In order to ensure this FBG only measures temperature changes and no strains, the FBG will be compressed during installation. The installation of the temperature FBG is visible in figure 8.6.

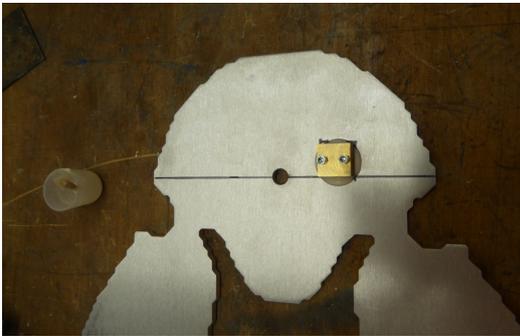


Figure 8.5: Eccentricity of the reference FBG.

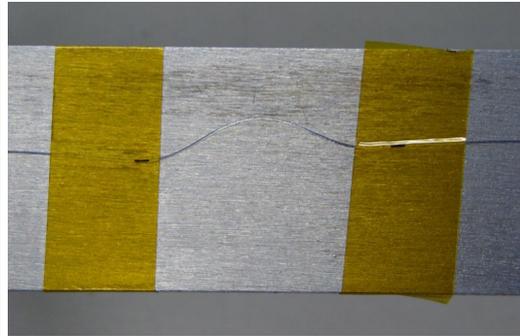


Figure 8.6: Temperature FBG.

The configuration as described above was used during the first experiments. However, the eccentricity of the FBG measuring the reference strains (see figure 8.4) caused the measurements of this FBG to be very noisy, as can be seen from figure 8.9a. This high noise made it impossible to interpret the results. To solve this problem, the reference FBG was moved to a new location, removing the eccentricity. This is visible in figure 8.7. The placement of the clamping bracket is displayed in figure 8.8. The distance between the two clamping brackets decreases from 500 mm to 405 mm for the transducer optimized in compression and 423 mm for the transducer optimized in tension, due to the relocation. To compensate this decrease in distance, a scaling factor is applied for the reference strain.

In order to be able to install the reference FBG at that location, one of the two FBGs measuring the upscaled strains had to be removed. This means only three of the four FBGs present in the fibre are used. The last FBG with the highest wavelength will not be used in this new configuration.

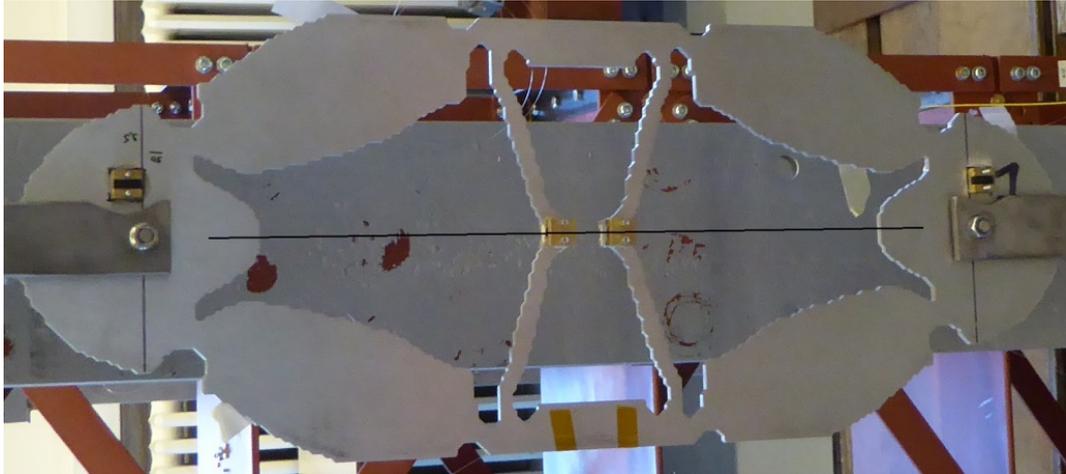


Figure 8.7: New location of the reference FBG.

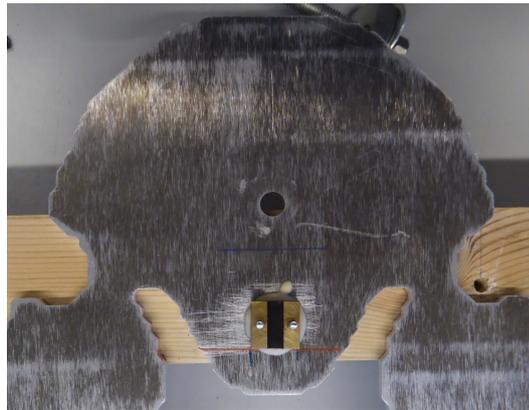


Figure 8.8: New location of the clamping brackets of the reference FBG.

It is clear from figure 8.9b that centering the reference FBG does significantly reduce the noise on the FBG measurements. However, the relocation of the reference FBG and consequently, the removal of one of the two FBGs measuring upscaled strain, makes it impossible to check if out-of-plane bending of the transducer occurs. Initial results of measurements with two FBGs measuring upscaled strain do indicate this out-of-plane bending does not occur, justifying the decision to remove one of those FBGs.

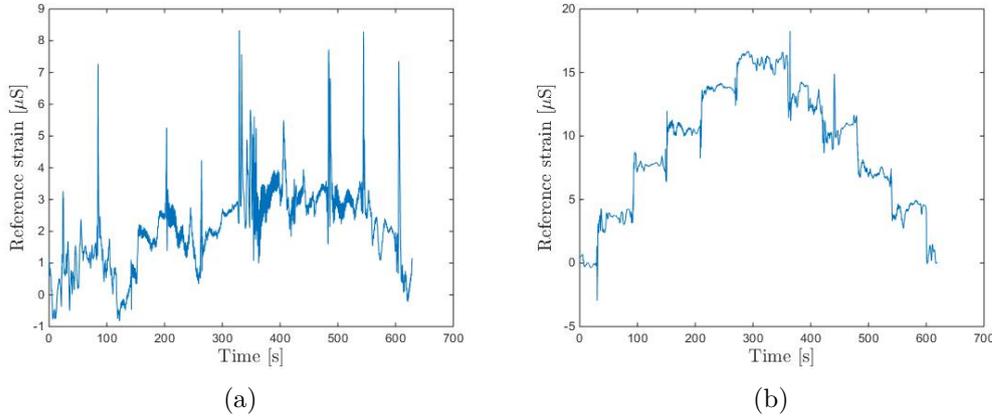


Figure 8.9: Graph with strains in the reference FBC in (a) eccentric and (b) centered configuration.

8.3 Discussion of the fibre mounting system

During topology optimization, as well as during the numerical verification, the assumption was made the fibre could be connected rigidly to the mounting platforms. This is an important simplification. In practice a certain amount of strain will always be lost in the connection between the fibre and the transducer.

A first possible way to attach the fibre to the transducer is to use glue. The gluing of a fibre has already been researched in the past. This research has proven that a portion of the strain is lost by the glued connection. According to Her and Huang [14] this portion is between 1 and 9 percent depending on the gluing length and the type of adhesive. A disadvantage of a glued connection is that the connection is permanent. Once the fibre is glued to the transducer, it cannot be removed.

At the structural mechanics section of the KU Leuven, a different system was developed to install the optical fibre. This system is displayed in figure 8.10. The system clamps the fibre instead of gluing it. Once the experiment is completed, the fibre can be removed. It is also possible to alter the pre-strain in the fibre using this clamping system. This clamping system has already been used in multiple experiments and has proven to work. In those experiments, a clamping bracket was placed every 50 cm over a length of 3 to 7 m, depending on the experiment. A schematic of the distribution of the clamping brackets in those experiments is displayed in figure 8.11. Given the fact that these clamping brackets have proven to work, and they allow the fibre to be removed and reused afterwards, the decision has been made to use these clamping brackets on the transducer.

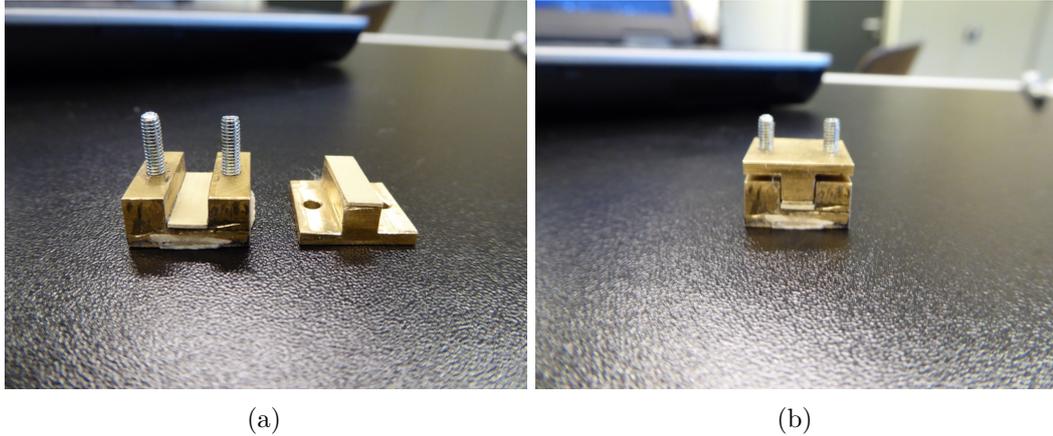


Figure 8.10: Clamping brackets used to install the fibre on the transducer.

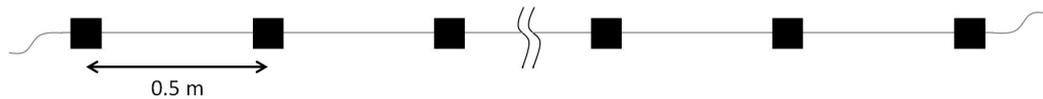


Figure 8.11: Fibre clamping system in the configuration it was previously used in other experiments.

The clamping brackets contain a rubber layer between the brass bracket and the optical fibre. This rubber layer will deform when subjected to shear forces. The deformation is also depicted in figure 8.12. The deformation of the rubber will result in some strain loss. It is important to note that the difference in spatial distribution of the clamping brackets in, on the one hand, previous experiments at the structural mechanics section and, on the other hand, the transducer, will have an effect on the strain loss. During the experiments of the structural mechanics section of the KU Leuven, the brackets were placed 50 cm apart. In the transducer, a space of only 2 cm is present between the brackets. As a consequence the loss in strain in the fibre will be 25 times larger in the transducer, when compared to previous experiments for the same deformation of the rubber. An other important difference is the fact that, during the experiments at the structural mechanics section, multiple brackets were placed on the same fibre. This means that the pre-strain on each side of the bracket will be similar, except for the two brackets at the ends of the fibre. The only difference in strain will be caused by the deformation of the structure it is attached to. Therefore, the shear forces on the rubber will be small, resulting in small deformations of the rubber. When using the clamping brackets on the transducer, the difference in strain will be much larger. Between the clamping brackets, where the FBG is located, a strain of up to $2000 \mu\text{S}$ is present. On the other side of the brackets no pre-strain is present at all. This is also visualised in figure 8.12.

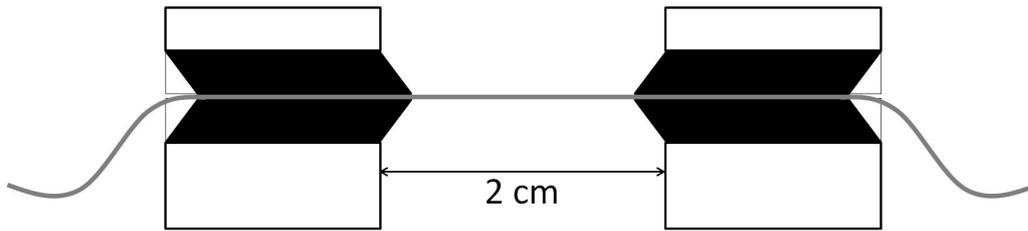


Figure 8.12: Deformation of the rubber while subjected to shear forces.

The combination of the small fibre length between the brackets and the large difference in strain on both sides of the brackets, makes the fibre mounting system susceptible to strain loss when applied to the transducer. Initial experiments confirmed this concern. Therefore the strain loss in the rubber was investigated.

A second rubber was also tested which should deform less under shear loading. The original rubber is quite flexible and has a thickness of 0.85 mm. The new rubber is much stiffer and has a thickness of just 0.4 mm. The new rubber is a composite material. It consists of a reinforcement mesh surrounded by a polymer matrix. The polymer matrix ensures the rubber has enough friction to clamp the fibre. The combination of a stiffer and thinner material, should result in smaller deformations of the rubber and therefore less strain loss.

In order to compare the strain loss of both rubbers, an experiment was executed. An optical fibre, containing two FBGs, was pre-strained between two weights to a strain of about $2500 \mu\text{S}$. Next, both FBGs were fixed between two clamping brackets. One FBG was fixed between clamping brackets equipped with the old rubber, the other was clamped between clamping brackets equipped with the new rubber. Lastly, the pre-strain of the weights was released. From the strain loss in both FBGs over time, a comparison of the rubbers can be made. The set-up of the experiment is also visualised in figure 8.13. By pre-straining the optical fibre using weights, one can be sure both FBGs experience the same pre-strain. Any difference in strain loss can therefore not be caused by a different pre-strain. The results of this experiment are displayed in figure 8.14. It is instantly clear that the new rubber performs much better than the old rubber. The pre-strain of the weights was released at time $t = 60 \text{ s}$. At this time a loss in pre-strain between the clamping brackets is visible for both rubbers. However, the loss with the old rubber is much larger than the loss with the new rubber. The old rubber slowly loses about $1600 \mu\text{S}$ of pre-strain over a period of 100 s. After that period the strain loss stabilises. At the end of the experiment a total of $1634 \mu\text{S}$ has been lost. The new rubber deforms faster than the old rubber. Within a period of 8 s, the maximal strain loss of $539 \mu\text{S}$ is reached. After the initial strain loss, the strain increases again. At the end of the experiment, the strain loss has been reduced to $281 \mu\text{S}$. No explanation has been found for the increase in pre-strain after initial losses.

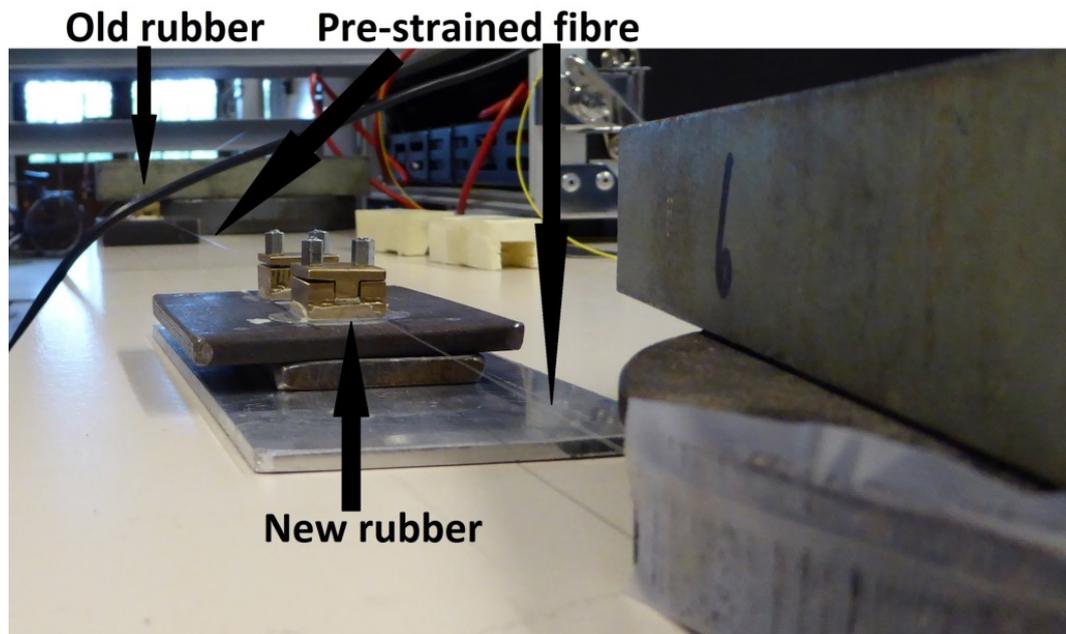


Figure 8.13: Experiment set-up to compare the performance of both rubbers.

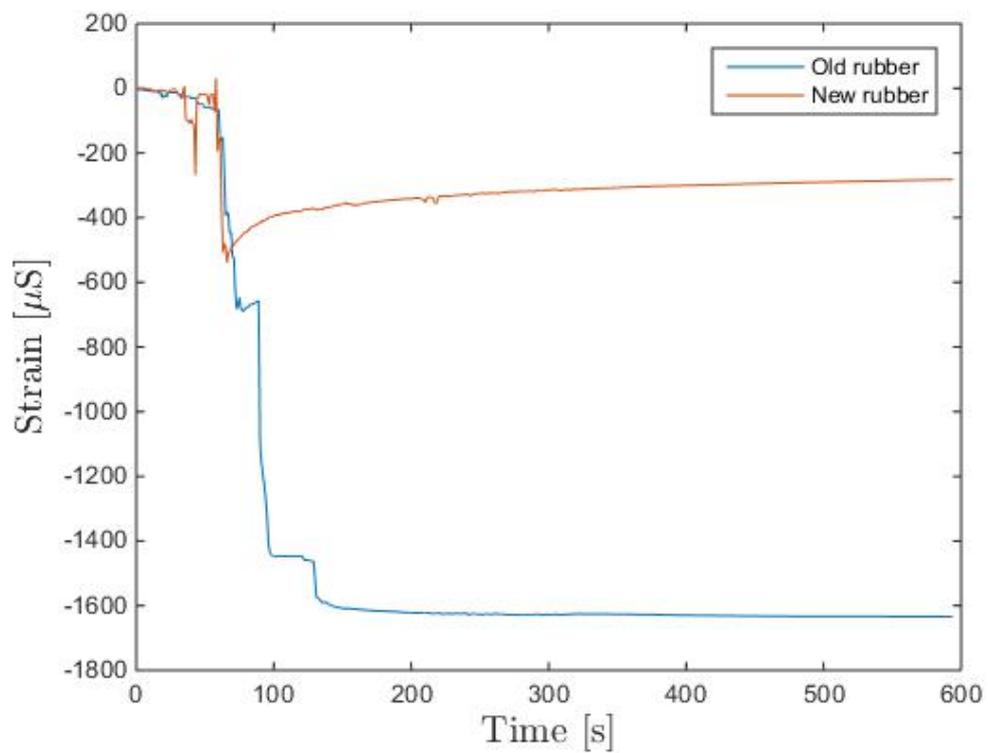


Figure 8.14: Comparison of the strain loss of the old and new rubber.

A strain loss of 1634 μS is of course not acceptable to perform accurate measurements. However, the displacement in the rubber corresponding to the strain loss is still pretty small. The displacement is calculated as follows:

$$\begin{aligned}\epsilon &= \frac{\Delta}{l} \\ \Rightarrow \Delta &= \epsilon \cdot l \\ &= 1634 \cdot 10^{-6} \cdot 20 \text{ mm} \\ &= 0.01634 \text{ mm}\end{aligned}\tag{8.1}$$

Where ϵ is the strain in the fibre, Δ is the decrease in length of the fibre and l is the length of the fibre. The decrease in length of the fibre is equally distributed over the rubbers at either side of the FBG. From this decrease in length, the angle θ of the rubber with a thickness $t = 0.85 \text{ mm}$ can be calculated:

$$\begin{aligned}\theta &= \arctan\left(\frac{\Delta/2}{t}\right) \\ &= 1.10^\circ\end{aligned}\tag{8.2}$$

The angle of deformation of the rubber is small.

The same calculation can be repeated for the new rubber. The obtained angle is even smaller:

$$\begin{aligned}\Delta &= \epsilon \cdot l \\ &= 539 \cdot 10^{-6} \cdot 20 \text{ mm} \\ &= 0.00539 \text{ mm} \\ \Rightarrow \theta &= \arctan\left(\frac{\Delta/2}{t}\right) \\ &= 0.39^\circ\end{aligned}\tag{8.3}$$

8.4 Static experiments

In this section, the results of the static experiments will be discussed. During the experimental phase of this thesis, many experiments were performed in order to gain experience and to fine-tune the procedure. The analysis in this report will only discuss three final experiments on both designs.

While executing the static experiment, the transducer will be loaded by weights. Every minute, 1 kg will be added, increasing the load on the transducer, until a total load of 5 kg is applied. Afterwards the weights will be removed again, one by one, with a one minute interval between the removals of each weight. This procedure will be repeated three times for each design. A view of the set-up of the static experiment is given in figure 8.15.

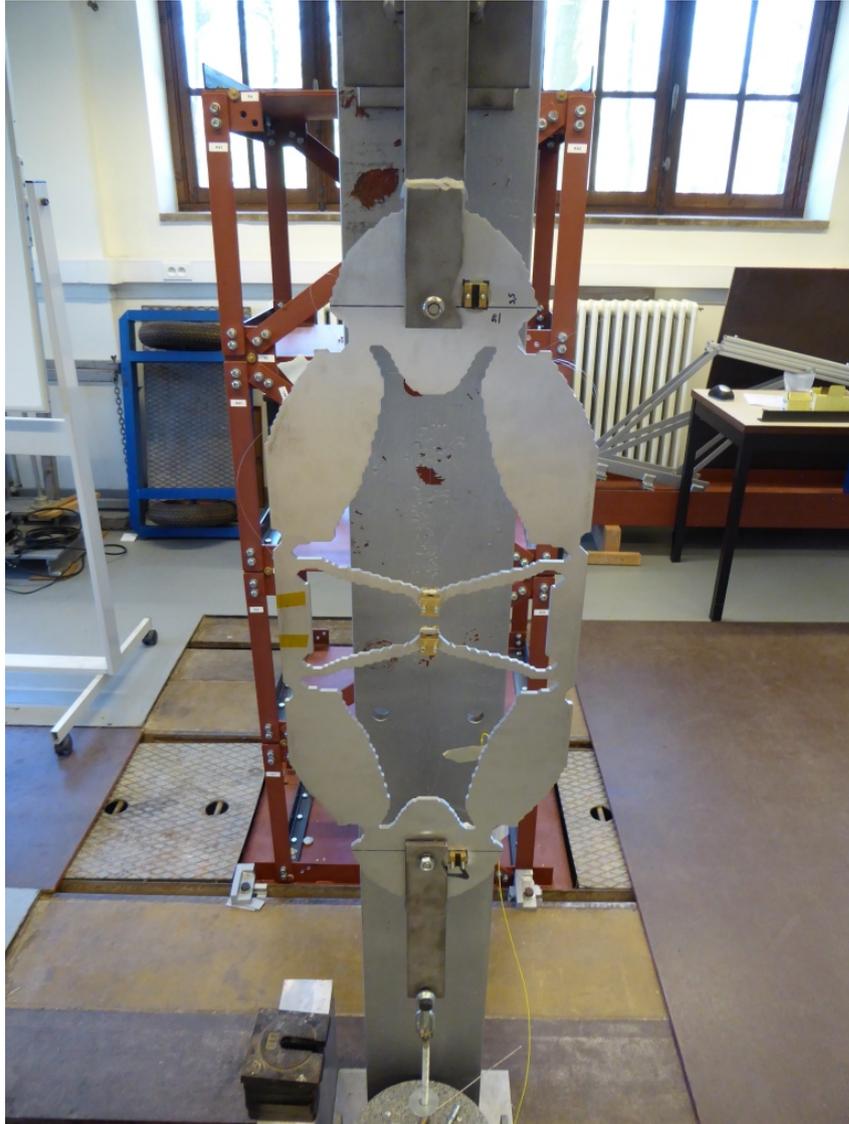


Figure 8.15: Set-up of the static experiments.

During the static experiments, the transducer will be loaded in tension. In order to avoid bending moments from being applied to the transducer by eccentric placement of the weights, a chain link is included between the weights and the transducer. This way, no in- nor out-of-plane bending moments are applied to the transducer. A detailed view of this chain link connection is visible in figure [8.16](#).

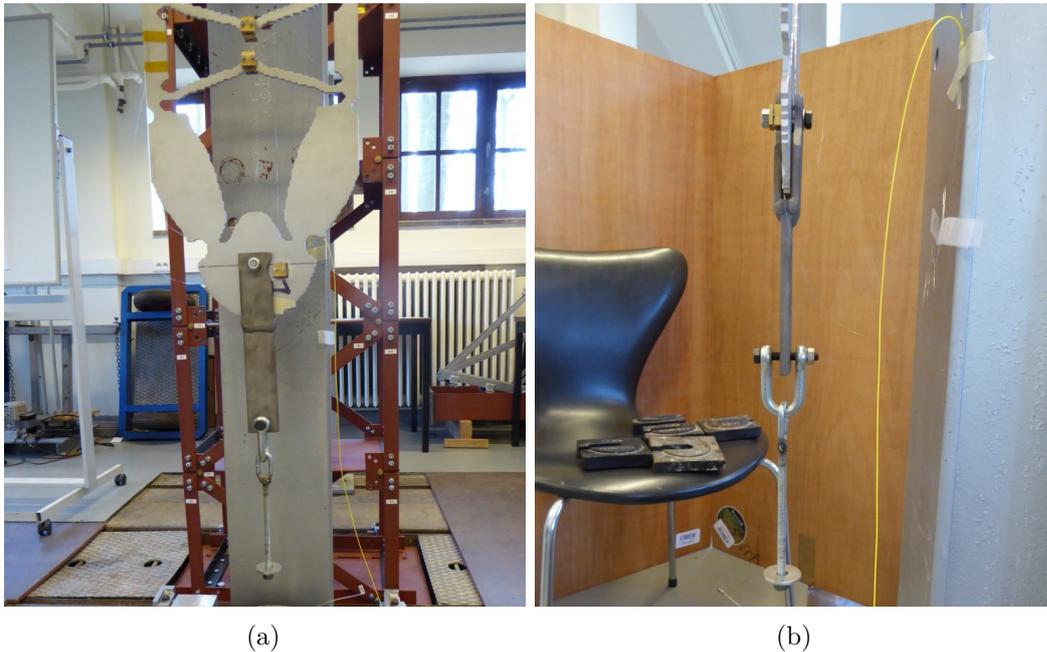


Figure 8.16: Detailed view of the chain connection of the weights to avoid eccentric loading and bending of the transducer.

8.4.1 Design optimized in compression

The transducer which is optimized in compression will be tested according to the previously described procedure. Note that, even though the transducer is designed to work under compressive loading, a tensile load will be exerted on the transducer during the static experiments. However, since the transducer behaves linearly elastically, this will have no influence on the performance of the transducer. To compensate for the tensile loading, one must make sure the optical fibre has been sufficiently pre-strained, as the strain in the FBG measuring upscaled strains will decrease when the transducer is loaded in tension. This FBG must be pre-strained at all times, even when the highest loads are applied to the transducer. Otherwise, the obtained results will not be correct.

The strains measured during a first static experiment are displayed in figure 8.17. In figure 8.17a reference and upscaled strains are displayed on the same figure. Due to the small value of the reference strains, they are not that clear. A more detailed view of the reference strain is given in figure 8.17b. A remark has to be made about the results of the reference strains. From the numerical model in ANSYS it can be found that the strain should increase with a value of $2.256 \mu\text{S}$ for each kg of load which is added. This is a very small increase, which approximates the accuracy of the fibre optical measuring technique. Glisic and Inaudi [13] mention an accuracy of $1 \mu\text{S}$. This explains the large proportion of noise on the measurements in figure 8.17b.

8. EXPERIMENTAL VALIDATION

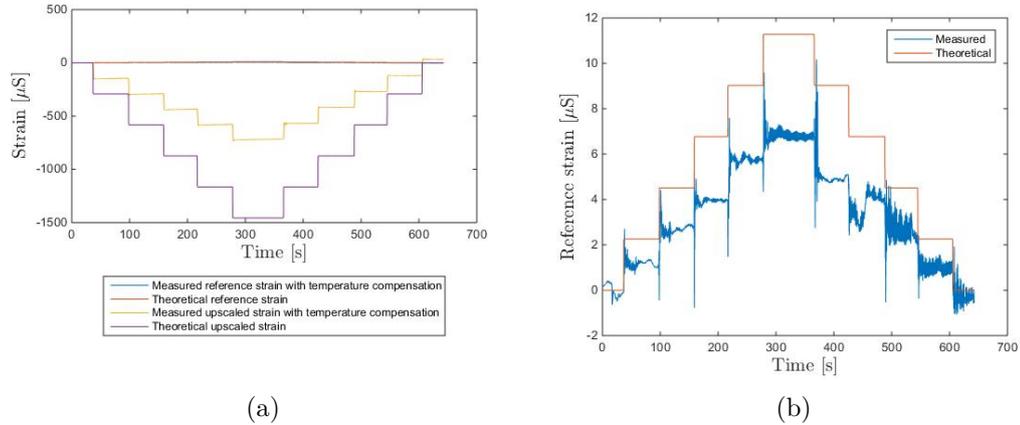


Figure 8.17: Measured strains during a first static test on the design optimized in compression: (a) all strains and (b) detail of the reference strains.

Another important remark which should be made is that the temperature can change during the experiment. It was previously mentioned that a change in temperature of $1\text{ }^{\circ}\text{C}$ results in a strain of $0.5\text{ }\mu\text{S}$. Figure 8.18 shows the fluctuations during the experiments. The variations are not negligible and therefore compensation is needed while processing the results.

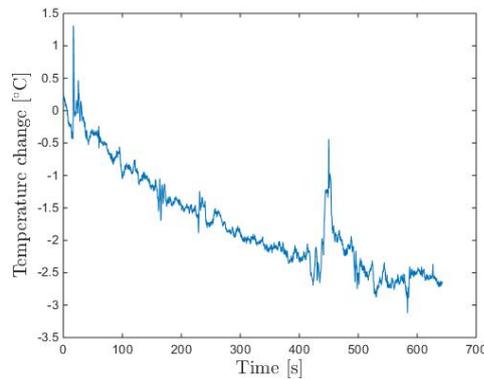


Figure 8.18: Temperature change during the first experiment.

An analysis of the measured upscaled strains, shows the strain decreases with a value of $144\text{ }\mu\text{S}$ with the application of each kg. A maximum absolute strain value of $720\text{ }\mu\text{S}$ is reached when all 5 kg are applied. According to the numerical model in ANSYS each load step should result in a decrease of $291.39\text{ }\mu\text{S}$ meaning that at a total load of 5 kg a strain of $-1457\text{ }\mu\text{S}$ should be reached. These values are not reached during the first static experiment. An explanation of this difference can be found in the fixation of the optical fibre. As previously mentioned, the deformation of the rubber in the clamping brackets will mean not all strain is transferred from the transducer to the optical fibre. The measured strains amount to about 49.4% of

the theoretically predicted values. An upscaling factor of 64 is reached, instead of the theoretical 129.

The relation between the reference strain and the upscaled strain is also displayed in figure 8.19. The slope of the regression line of the upscaled strain as a function of reference strain is equal to the measured upscaling factor. In figure 8.19a the measured upscaled strain is displayed as a function of the measured reference strain. However, as mentioned before, the measurements of the reference strain contain a large proportion of noise. This noise results in an incorrect upscaling factor of 102. In order to exclude the noisy measurements of the reference strain, the relation between the theoretical reference strain and the measured upscaled strain is displayed in figure 8.19b. The upscaling factor which is displayed in this figure is equal to 65. This is the same value as previously mentioned when analysing figure 8.17a. Note that some outliers are present in this graph. The outliers can be explained by a dynamic response while applying the weights.

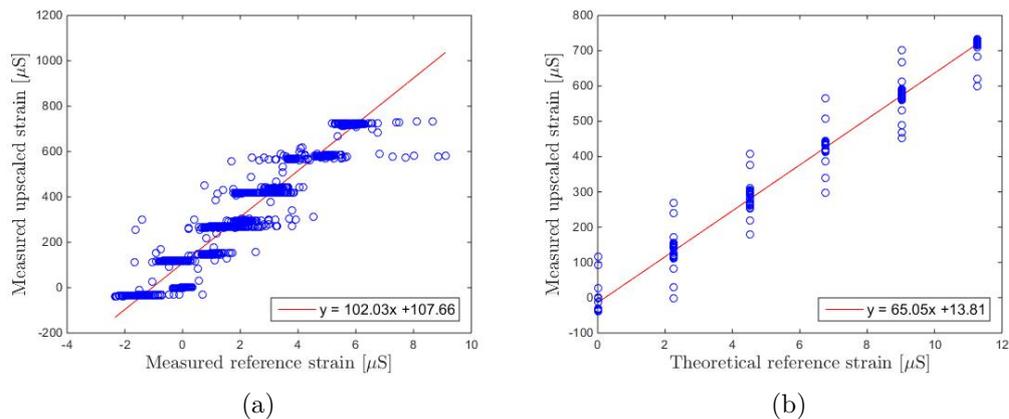


Figure 8.19: Relation between the (a) measured and (b) theoretical reference strain and the upscaled strains.

Next, the second and third experiment will be discussed. Between the first experiments and the experiments which will be analysed next, the optical fibre was removed from the transducer and reinstalled. This way, it is possible to check whether or not the installation of the fibre has an influence on the upscaling factor. The fibre was not removed and reinstalled between experiments two and three.

The results for experiment two are displayed in figure 8.20. From these graphs, one can see that the upscaled strain decreases with $164 \mu\text{S}$ per kg with a maximum absolute value of $820 \mu\text{S}$. This is significantly more than during the first experiment. However, it is still only 56.3% of the theoretical value. From this data it can be concluded that the upscaling factor during experiment two is 73, which is 12% more than during the first experiment. This is also confirmed by the regression line displayed in figure 8.20.

8. EXPERIMENTAL VALIDATION

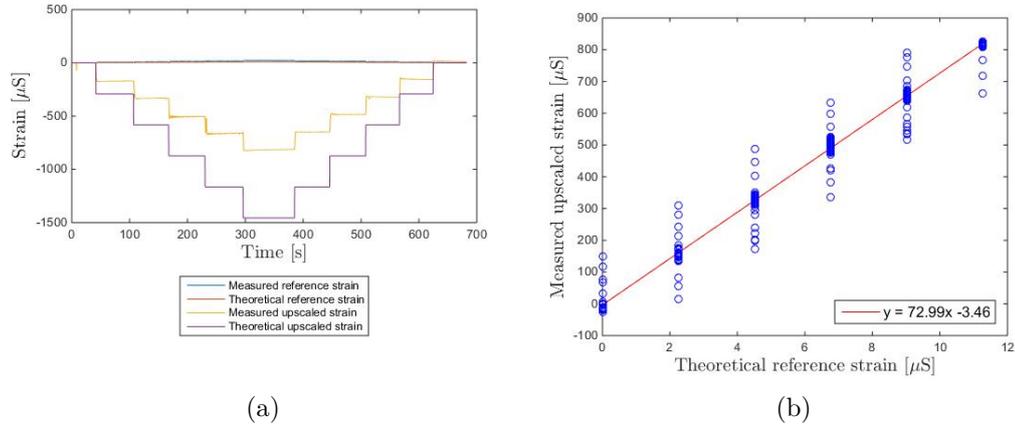


Figure 8.20: Results of a second static test on the design optimized in compression: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.

A third and final static experiment for the design optimized in compression yields exactly the same results as the second experiment. An upscaled strain of 164 μS per kg is found resulting in a total strain of 820 μS . The upscaling factor is again equal to 73, which is confirmed by figure 8.21.

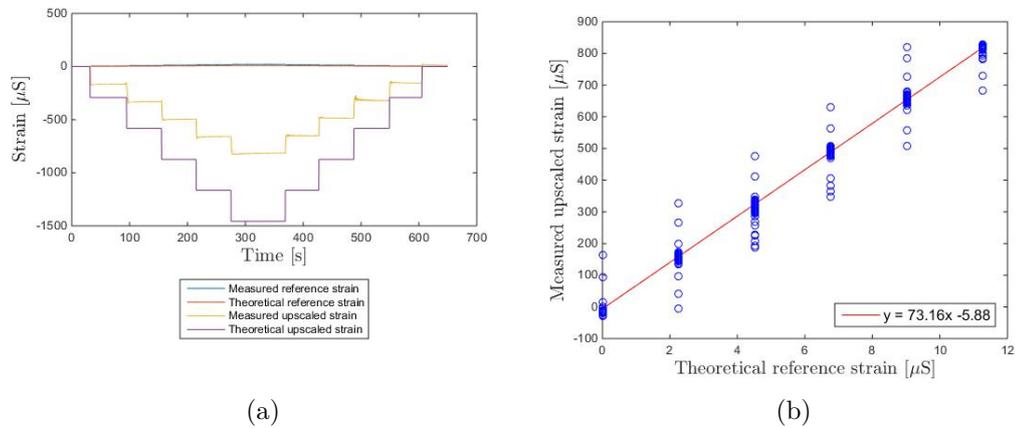


Figure 8.21: Results of a third static test on the design optimized in compression: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.

The fact that experiments two and three yield exactly the same results, indicates that once the fibre is mounted, the upscaling factor is very consistent. However, between experiment one and two, a difference in upscaling factor of 12% was found. All experiments were performed under exactly the same conditions. The only difference between experiment one and two is the fact that the fibre was removed and reinstalled. All other parameters remained the same. The increase in upscaling

Table 8.3: Summary of the static experiments on the transducer optimized in compression.

Number	Upscaling factor	
	Theoretical	Measured
1		65
2	129	73
3		73

factor suggests that the installation of the fibre has a significant influence on the performance of the sensor package. A calibration of the strain upscaling sensor package is therefore recommended every time the fibre is installed.

A summary of the results of the static experiments on the transducer optimized in compression is given in table 8.3.

8.4.2 Design optimized in tension

The design optimized in tension will be tested using the same procedure as the static experiments on the transducer optimized in compression. This design is stiffer than the design optimized in compression. As a result, a weight of 1 kg will result in a theoretical reference strain of $1 \mu\text{S}$ and a theoretical upscaled strain of $94.06 \mu\text{S}$. According to the calculation in ANSYS, a maximum upscaled strain of $470 \mu\text{S}$ should be reached when a load of 5 kg is applied. Figures 8.22, 8.23 and 8.24 display the results obtained for the first, second and third static experiment respectively.

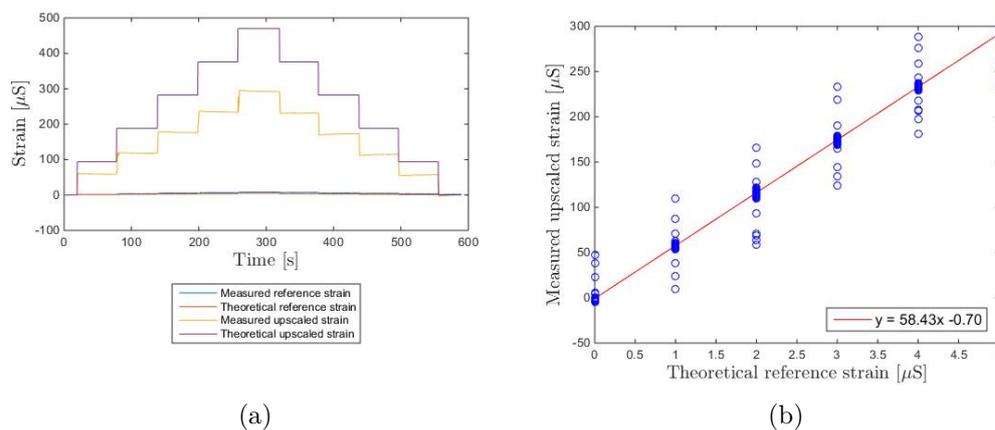


Figure 8.22: Results of a first static test on the design optimized in tension: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.

During the first experiment, a maximal upscaled strain of $294 \mu\text{S}$ was obtained.

8. EXPERIMENTAL VALIDATION

That is about 62.6% of the numerically predicted value. Therefore, an upscaling factor of about 59 is found instead of the numerically predicted value of 94. An upscaling factor of 58.4 is indicated by the regression line, confirming the calculation above. The second and third experiments give very similar results. During both experiments the obtained upscaling factor is equal to 59. A summary of the results of all static experiments on the transducer optimized in tension is given in table 8.4.

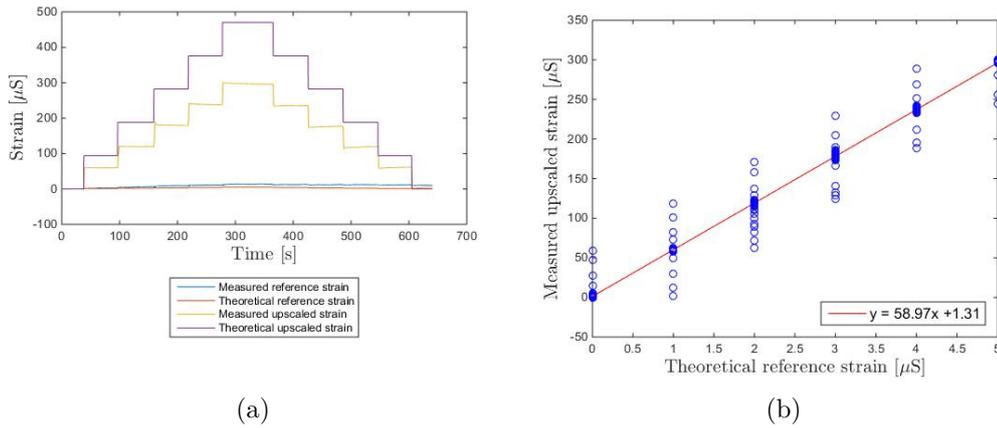


Figure 8.23: Results of a second static test on the design optimized in tension: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.

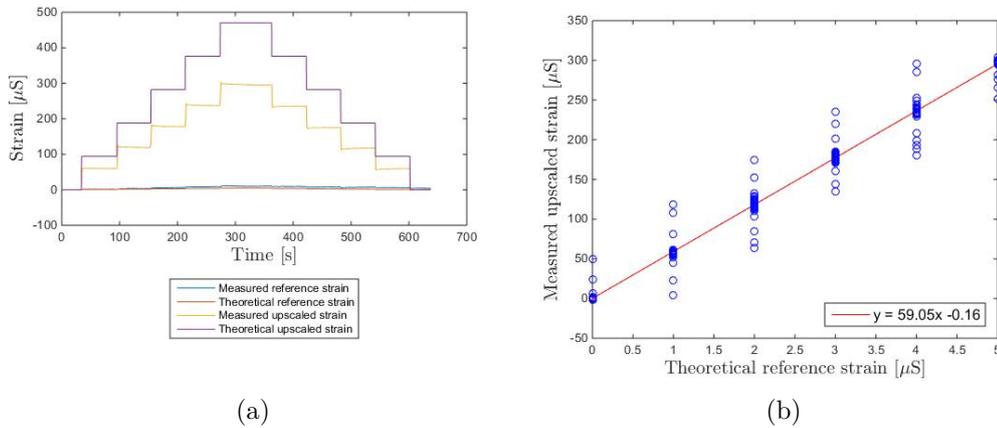


Figure 8.24: Results of a third static test on the design optimized in tension: (a) all measured strains and (b) relation between the theoretical reference strain and the upscaled strains.

The experiments on the transducer optimized in tension confirm the results from the design optimized in compression. In both designs, the obtained upscaling factor is significantly lower than the values predicted with ANSYS. This is due to the fact that not all strains are transferred from the transducer to the optical fibre using the

clamping brackets. From the three different experiments on the design optimized in tension, one can conclude the obtained upscaling factor is pretty consistent over all experiments.

Table 8.4: Summary of the static experiments on the transducer optimized in tension.

Number	Upscaling factor	
	Theoretical	Measured
1		58
2	94	59
3		59

8.5 Dynamic experiments

In the previous section, the transducers were loaded statically. However, the strain upscaling sensor packages are also designed to measure dynamic strains. To test the behaviour of the transducers during dynamic loading, they will be connected to an electromechanical shaker. This is done by replacing the chain link of the static test with a spring connection to the shaker. In order to know the force which is applied to the transducer, a load cell is also added. This is necessary in order to calculate the upscaling factor. The changes to the set-up are visible in figure 8.25.

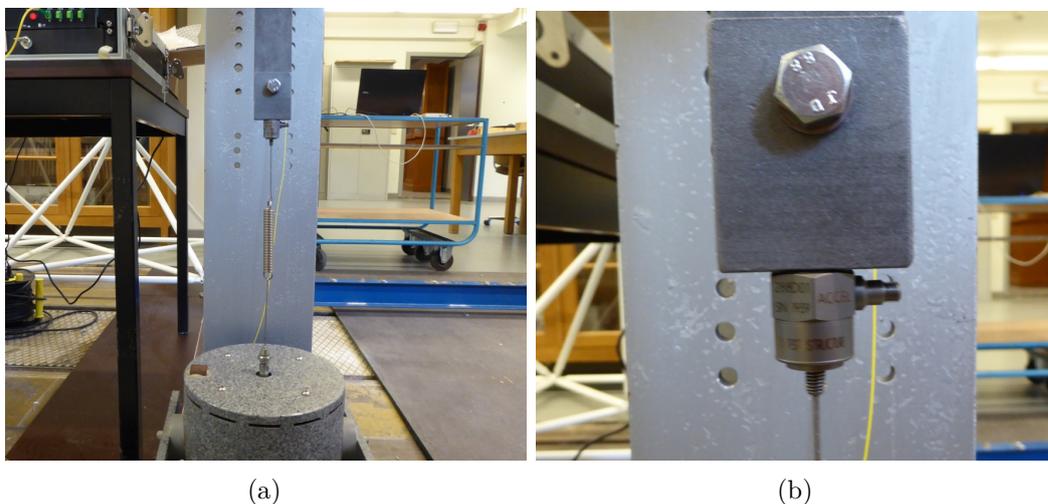


Figure 8.25: (a) Connection to the electromechanical shaker and (b) detail of the load cell.

Using the electromechanical shaker, a sinusoidal load will be applied to the transducer. Two sine functions will be tested, one with a frequency of 2 Hz and

one with a frequency of 4 Hz. It is important to remark that the bottom bolt connection of the transducer is essentially free to move out-of-plane. During the dynamic experiments, it became clear that this movement would occur if the load force became too large. To avoid this situation, the applied load was chosen low enough for the out-of-plane displacements not to occur.

8.5.1 Design optimized in compression

Before analysing the results of the dynamic experiments on the design optimized in compression, it should be noted that these experiments were executed directly after the first static test on the transducer optimized in compression. The fibre was not removed from the transducer in between the static and dynamic experiment. Consequently, the obtained upscaling factor during these dynamic experiments should be compared with the upscaling factor calculated from the first static experiment.

First, the design optimized in compression is subjected to a sinusoidal load with a frequency of 2 Hz and an amplitude of 7.5 N. The sinusoidal signal is also recognisable in the measured upscaled strains. Both the applied load and the upscaled strains are displayed in figure 8.26.

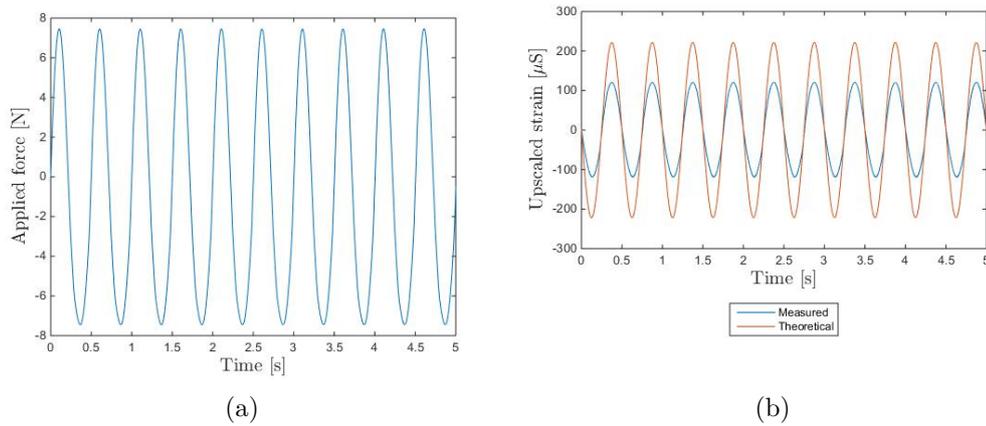


Figure 8.26: (a) Applied force and (b) measured and theoretical upscaled strains for the dynamic test with a loading frequency of 2 Hz on the design optimized in compression.

It is clear that the measured upscaled strains are again smaller than predicted using ANSYS. An amplitude of 122 μS is measured, while a value of 222 μS was predicted. The measured value is 55.0% of the theoretical value. The upscaling factor is therefore equal to 70.9. This result is confirmed by the regression line in figure 8.27, where an upscaling factor of 70.4 is displayed. In figure 8.27, one can also see some hysteresis between loading and unloading. This is due to the deformations of the rubber in the clamping brackets.

The obtained upscaling factor of 70 is about 8% larger than the value obtained during the first static experiment, where an upscaling factor of 65 was calculated. The difference might be explained by the fact that the rubber in the clamping bracket might deform less during the dynamic experiment, as there is less time for the deformation to occur.

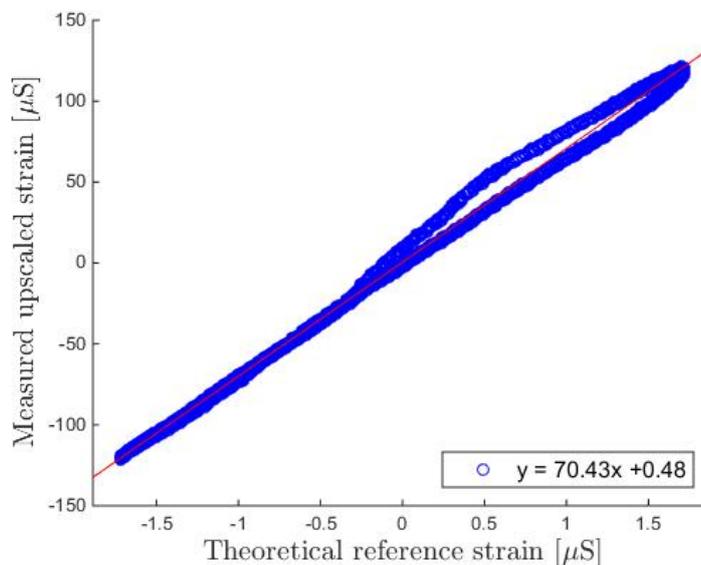


Figure 8.27: Relation between the theoretical reference strains and the measured upscaled strains for the dynamic test with a loading frequency of 2 Hz on the design optimized in compression.

A second dynamic experiment on the design optimized in compression is conducted using a sinusoidal load with a frequency of 4 Hz and an amplitude of 8.8 N. The applied load, as well as the measured and theoretical upscaled displacements are displayed in figure 8.28. An upscaled strain of 141 μS is measured which is 53.8% of the predicted theoretical upscaled strain of 262 μS . The experiment results therefore in an upscaling factor of 69. The regression line in figure 8.29 suggests a slightly lower upscaling factor of 67. In figure 8.29 hysteresis is again clearly present, even more than during the dynamic experiment with an excitation frequency of 2 Hz. The obtained upscaling factor is again slightly larger than the value obtained during the first static experiment, yet smaller than the upscaling factor obtained during the dynamic experiment with an excitation frequency of 2 Hz.

8. EXPERIMENTAL VALIDATION

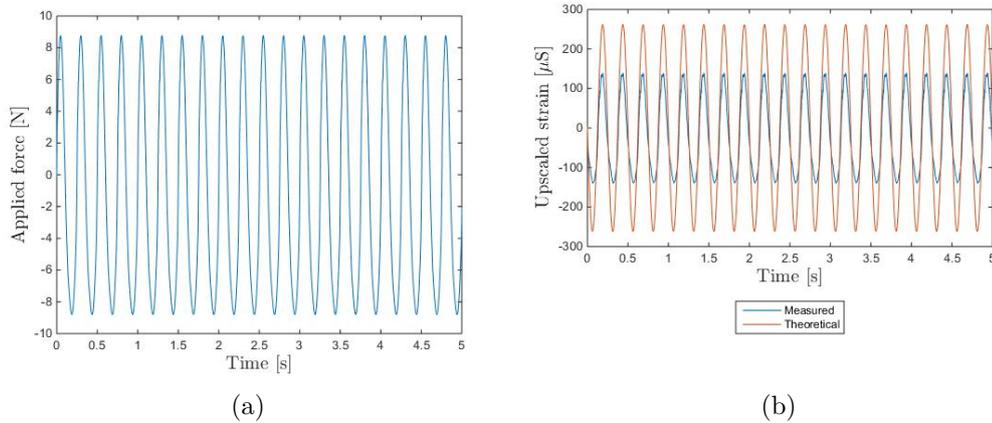


Figure 8.28: (a) Applied force and (b) measured and theoretical upscaled strains for the dynamic test with a loading frequency of 4 Hz on the design optimized in compression.

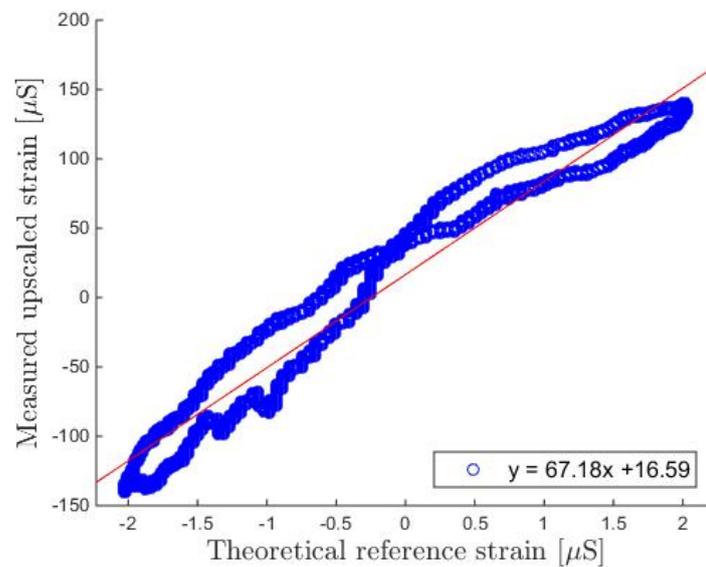


Figure 8.29: Relation between the theoretical reference strains and the measured upscaled strains for the dynamic test with a loading frequency of 4 Hz on the design optimized in compression.

A summary of the results obtained during the dynamic experiments on the transducer optimized in compression is given in table 8.5.

Table 8.5: Summary of the dynamic experiments on the transducer optimized in compression.

Number	Loading frequency	Upscaling factor	
		Theoretical	Measured
1	2 Hz	129	70
2	4 Hz		67

8.5.2 Design optimized in tension

The first dynamic experiment on the transducer optimized in tension is conducted by applying a sinusoidal load with a frequency of 2 Hz and an amplitude of 6.9 N. The applied load and the measured and theoretical upscaled strains are displayed in figure 8.30.

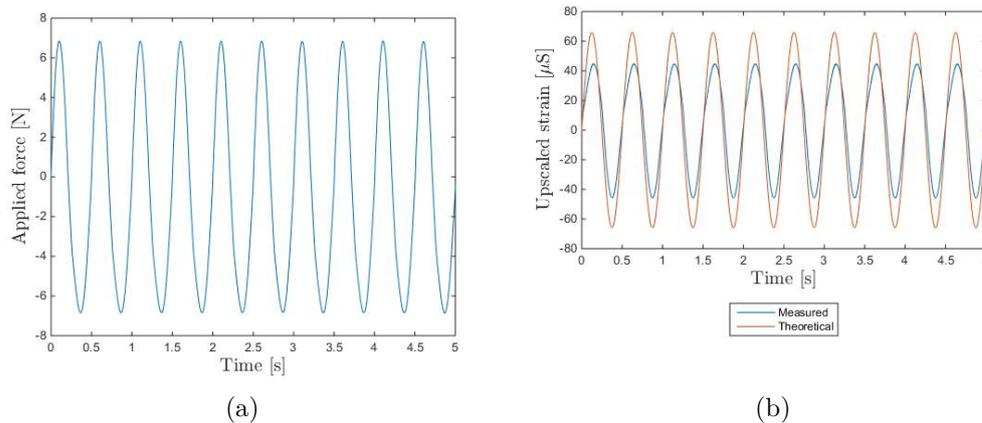


Figure 8.30: (a) Applied force and (b) measured and theoretical upscaled strains for the dynamic test with a loading frequency of 2 Hz on the design optimized in tension.

A strain amplitude of 46 μS is found, which is 70% of the theoretical 66 μS . The strain upscaling factor is therefore 70% of the theoretical 94, which is equal to 66. The regression line in figure 8.31 has a slightly smaller slope or upscaling factor of 63. The difference between the two values can be explained by the hysteresis present in figure 8.31. The upscaling factor obtained during the dynamic experiment is again larger than the value obtained during the static experiment.

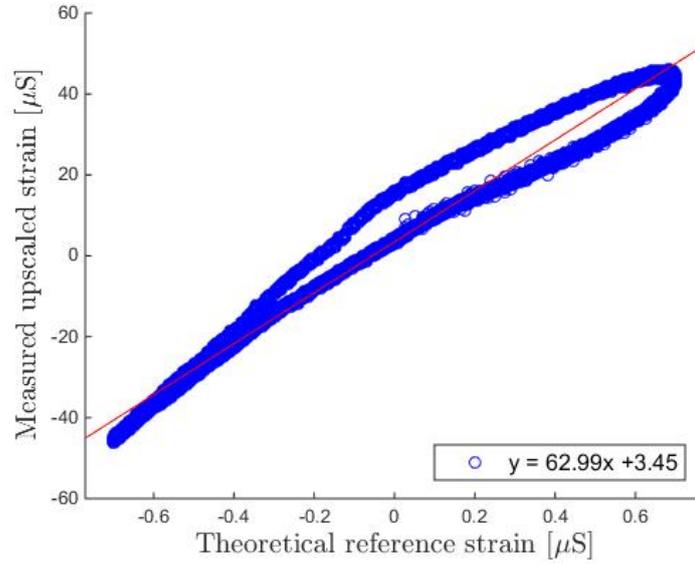


Figure 8.31: Relation between the theoretical reference strains and the measured upscaled strains for the dynamic test with a loading frequency of 2 Hz on the design optimized in tension.

The last dynamic experiment which is conducted exerts a sinusoidal load with a frequency of 4 Hz and an amplitude of 7.0 N on the transducer optimized in tension. The applied load and the resulting measured and theoretical upscaled strains are displayed in figure 8.32.

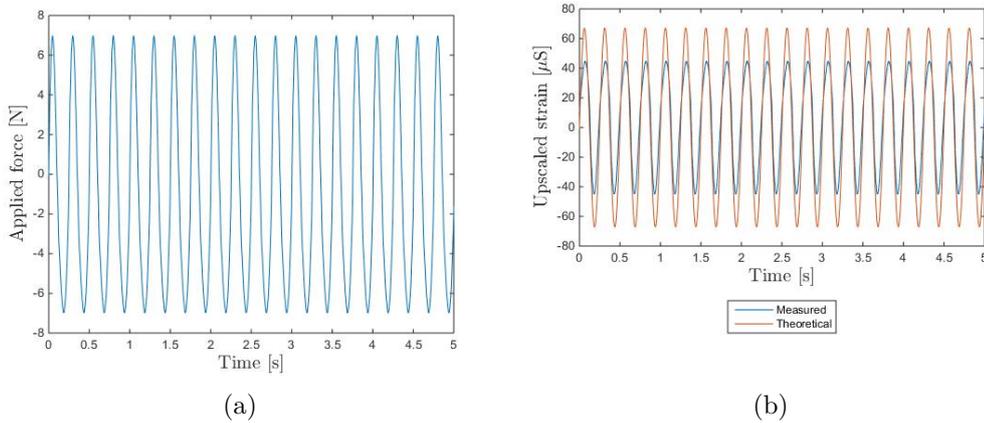


Figure 8.32: (a) Applied force and (b) measured and theoretical upscaled strains for the dynamic test with a loading frequency of 4 Hz on the design optimized in tension.

During the dynamic experiment with a load frequency of 4 Hz, an upscaled strain amplitude of 45 μS is measured. The theoretical value is equal to 67 μS , which

Table 8.6: Summary of the dynamic experiments on the transducer optimized in tension.

Number	Loading frequency	Upscaling factor	
		Theoretical	Measured
1	2 Hz	94	63
2	4 Hz		62

means the measured upscaled strain is 67% of the theoretical upscaled strain. The upscaling factor should therefore be equal to 67% of 94, which is equal to a measured upscaling factor of 63. The result obtained using the regression line in figure 8.33, suggests a similar upscaling factor of 62. One can again observe some hysteresis in figure 8.33. The obtained upscaling factor is again larger than the value calculated from the static experiment.

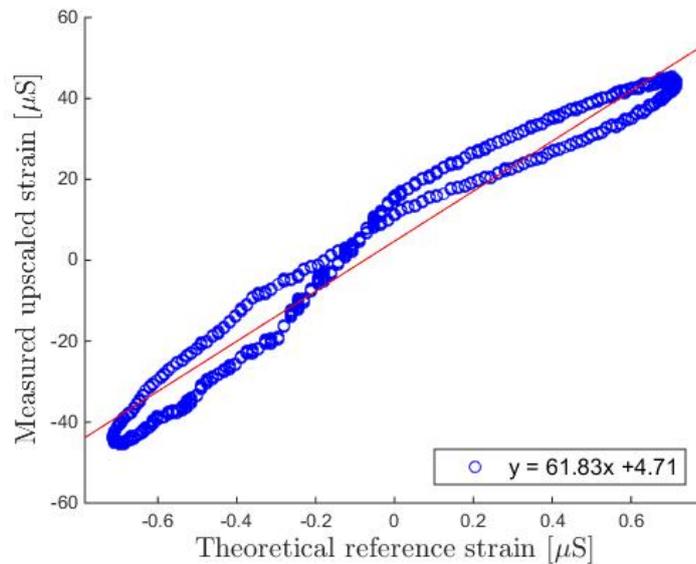


Figure 8.33: Relation between the theoretical reference strains and the measured upscaled strains for the dynamic test with a loading frequency of 4 Hz on the design optimized in tension.

A summary of the results obtained during the dynamic experiments on the transducer optimized in tension is given in table 8.6.

8.6 Further research

The previously described experiments give a first indication of the performance of the transducers. Beside the static and dynamic experiments on the transducer itself,

a number of different experiments could also be performed on the strain upscaling sensor package. These experiments were not yet performed due to the time constraint of this thesis. However, they would be very useful for the further development of the sensor package.

8.6.1 Research on the fibre mounting system

During the experiments, it was concluded not all strain was transferred from the transducer to the optical fibre. Depending on the design and the executed test, 50 to 70% of the predicted strain was measured. Furthermore, a difference in upscaling factor of 12% was found between the first and second static experiment on the transducer optimized in compression. The difference between both experiments was attributed to the removal and reinstallation of the optical fibre. Clearly the mounting system has a large influence on the strain upscaling. It is therefore important to investigate how the strain transfer from the transducer to the fibre can be increased. The lack of consistency in the results of identical experiments should be investigated and resolved.

Three different suggestions for further research can be proposed. First of all, one can investigate if a better suited rubber (or other material) exists. At the beginning of this chapter it was mentioned that the deformations of the rubber have a big influence on the strain transfer. A new stiffer rubber was already selected to increase the transfer. However, it would be of interest to search for even better materials.

Secondly, an alteration to the fibre mounting system can be proposed. In the current system, the FBG measuring upscaled strains is mounted by two clamping brackets, one on each side of the FBG. Two additional brackets could be added, meaning two brackets are placed on either side of the FBG. The additional brackets should decrease the shear forces on the original brackets. By tensioning a dummy fibre between the additional and original brackets, the pre-strain at both sides of the original brackets should be similar and the rubber will deform less. The working principle of this alteration is displayed in figure 8.34. The dummy fibre will not contain an FBG and will therefore not measure any strains. The addition of a second clamping bracket next to the original one, does however require an alteration of the design of the transducer, as more room has to be provided to place the clamping brackets.

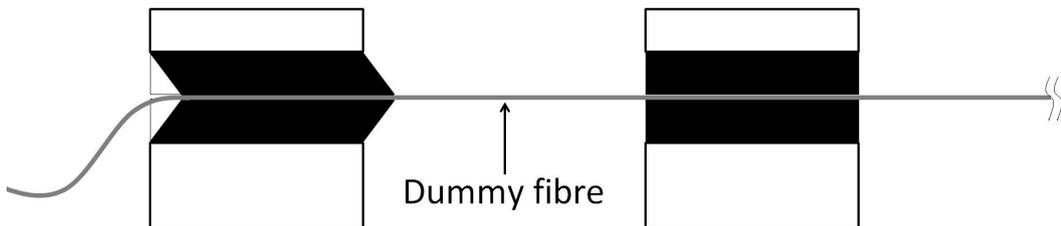


Figure 8.34: Altered fibre mounting mechanism: the addition of a second clamping bracket with a dummy fibre between the two brackets.

A third and final alteration of the fibre mounting which can be executed is to glue the fibre to the transducer. By gluing the fibre to the transducer directly, less strain might be lost. Glisic and Inaudi mention a strain loss of 1 to 9% depending on the type of glue and the gluing length.[13] This loss is significantly lower than the loss measured when using the clamping system. To implement this different fibre mounting technique, the design of the transducer should again be altered in order to ensure a sufficiently large gluing length. The previously mentioned paper suggests a gluing length of at least 5 cm.[13] By gluing the fibre, the connection is permanent and the fibre can no longer be removed from the transducer.

8.6.2 Static and dynamic measurements on a beam

During the experiments which were already performed, the behaviour of the transducer itself was always tested. However, the transducers have not yet been tested while being attached to a structure. A next step would therefore be to attach multiple transducers to a beam. This beam would then be statically and dynamically loaded in the laboratory.

A possible static experiment could be to perform a three or four point bending test on a concrete beam. The deformation of the beam under a certain load can easily be calculated. The theoretical deformations can then be compared to the values measured by the transducers, in order to verify the performance of the transducer.

To test the dynamic performance of the transducers, two possible experiments can be proposed. A first experiment would be to dynamically excite the beam using an impact hammer. If the transducers work correctly, they should be able to identify the eigenfrequencies of the beam. A second experiment would be to connect an electromechanical shaker to the beam. The shaker will excite the beam. The transducer should be able to identify the frequency and load applied by the electromechanical shaker.

8.6.3 Long-term monitoring on a beam

Another interesting experiment would be to monitor a beam over a longer period of time. This experiment could span several days, weeks or even months. By performing this experiment, one could check if the performance of the transducer would change in the long-term. An example of change in long-term performance is the loss of pre-strain. When pre-strain is lost, incorrect strains would be measured. The incorrect data would inhibit structural health monitoring, which is the ultimate objective of the transducers.

8.6.4 Monitoring of a real-life structure

After extensive testing in the laboratory has proven the performance of the transducers, a final experiment can be performed. The final experiment would be to test

the strain upscaling sensor package on a real-life structure. During this experiment, the dynamic response of a bridge to ambient loading will be checked. By identifying the eigenmodes and locating possible damage to the bridge, a structural health assessment can be made.

8.7 Conclusions

In this final chapter, an experimental validation of the strain upscaling sensor package was performed. After a short introduction to the fibre optic sensing method using FBGs, the different components of the sensor package were discussed. In previous chapters, solely the design of the transducer was analysed. However, there are other components as well: the optical fibre, the clamping brackets and the interrogator. Especially the performance of the clamping brackets proved to have a significant influence on the upscaling of the strains. The clamping brackets transfer the displacements in the transducer to the optical fibre. A certain amount of strain is lost during this transfer, reducing the performance of the strain upscaling sensor package. In order to decrease the strain loss, the deformations of the rubber component in the clamping brackets were investigated. A new, stiffer material was introduced which resulted in a higher strain transfer.

In a next step, the transducers were tested statically as well as dynamically. The measured upscaling factors were 50 to 70% of the numerically predicted values, depending on the design and the experiment. The difference between the numerically predicted and the measured upscaling factors were attributed to the strain loss in the clamping brackets. The transducer optimized in compression had a measured upscaling factor of 65 to 73. The transducer optimized in tension resulted in measured upscaling factors of 58 to 63. A difference in upscaling factor between static and dynamic experiments was observed. This difference was attributed to the fact that the rubbers in the clamping brackets had less time to deform during the dynamic experiments, therefore reducing the strain loss in the rubbers. Another observation which was made during the static experiments of the transducer optimized in compression was the fact that the removal and reinstallation of the optical fibre to the transducer had a significant influence on the measured upscaling factor. A difference in upscaling factor of 12% was observed before and after the reinstallation of the fibre. This suggests a calibration of the strain upscaling sensor package is required every time the fibre is removed. When the fibre was not removed between experiments, the results proved to be very consistent.

In the last part of this chapter, future research in the development of the strain upscaling sensor package was proposed. During the experiments, the installation of the fibre proved to have a large influence on the performance of the sensor package. Therefore, the first step is to reduce the strain loss and to increase the repeatability. Three different solutions were suggested: reduction of the deformations in the rubber by searching an even better suited material, doubling the amount of clamping

brackets to install the FBG and gluing the fibre to the transducer. A second step is to attach the sensor package to a beam and to test if the package can accurately measure the deformations in the beam. During these experiments, the beam would be loaded statically and dynamically. A third step of the further development is to check if the sensor package measures accurate long-term results. A final step would be to test the strain upscaling sensor package on a real-life structure, for example a bridge, in order to perform a structural health assessment.

The future research has not yet been conducted due to the time constraint of the MSc thesis. However, the performed experiments show some promising results, and give a clear direction for further development.

Chapter 9

Conclusions

Throughout this thesis, the development of the strain upscaling sensor package was discussed, starting from a simplistic design in chapter 2 until the experimental validation in chapter 8.

In a first part of this thesis, the design was made using topology optimization. The 88 lines of code algorithm of Andreassen et al. [2] was used as a starting point for the topology optimization. In chapter 2 a first design was obtained using the first topology optimization algorithm. However, two problems were observed in the obtained design. The first problem was the occurrence of single node hinges, the second problem was the fact that bending of the fibre could occur. To solve both problems, more robust topology optimization algorithms were developed. In chapter 3 a solution to the single node hinges was implemented, based on the work of Wang et al. [39]. By applying the more robust optimization algorithm, length scale control was introduced and single node hinges were eliminated. The problem of the bending of the fibre was solved in chapter 4. By introducing a min-max formulation the horizontal displacement at the top and the bottom of the mounting platform became identical, and no bending of the fibre could occur. In chapter 5, an attempt was made to combine the models of chapter 3 and 4. This combination was however difficult, as this algorithm could result in design where bending of the fibre was possible. The designs obtained by this last, most complex algorithm have to be analysed carefully to make sure bending of the fibre does not occur. The technique of topology optimization proved to be very powerful for the design of the transducer. Using the previously mentioned topology optimization algorithms, two designs were selected to be produced. The first design was optimized to work in compression, the second design was optimized in tension.

The selected designs were numerically validated in a second part of this thesis in chapter 7. First, a verification of the upscaling factors calculated during topology optimization was performed. The calculations in ANSYS resulted in larger upscaling factors than the values obtained during topology optimization in MATLAB. During the numerical validation, upscaling factors of 129 and 94 were obtained for the designs

in compression and tension respectively. Next, properties of the transducers which were not included in the topology optimization were investigated. The stresses were calculated and proved not to be a problem. The eigenfrequencies of the transducer were also checked. With all eigenfrequencies above 60 and 67 Hz for the design in compression and tension respectively, the requirement for the eigenfrequencies to be larger than 50 Hz was met. As a last part of the numerical validation, the buckling loads of the transducers were calculated. Both designs proved to be insensitive to buckling under normal operational loading conditions.

In a last part of this thesis, in chapter 8, an experimental validation of the sensors was performed. By loading the transducers statically and dynamically, the strain upscaling factors of both designs were determined experimentally. The measured upscaling factors were significantly lower than the numerically predicted values. Upscaling factors of 50 to 70% of the predicted values were measured. The difference between both values was attributed to the strain loss in the connection between the transducer and the optical fibre, which contains a rubber layer. The experimentally determined upscaling factor of the design optimized in compression ranged between 65 and 73. For the transducer optimized in tension, values between 58 and 63 were measured. Also the specific mounting condition of the fibre proved to have an influence on the measured upscaling factor. A difference of 12% was found after removing and reinstalling the fibre. To conclude this thesis, some future research was proposed in order to further develop the sensor package. The strain loss in the connection between the transducer and the optical fibre should be researched. More extensive experiments should also be performed with the transducer attached to a beam. Next, long-term accuracy of the sensor package should be investigated. A final experiment which is suggested is to measure the displacements of a real-life structure, and to perform a structural health assessment using the strain upscaling sensor package.

Appendices

Appendix A

MATLAB code: Standard Topology Optimization

A.1 Input Parameters

```
1 %%%
2 % All units in kN and mm
3 %%%
4 clear;
5 close all;
6 %% DIMENSIONS SENSOR PACKAGE
7 symmetricLength = 250;
8 width = symmetricLength;
9 rBolt = 2.5;
10 mountingPlatform = 15;
11 gridsize = 2.5;
12 %% POSITION MEASURING GAUGE
13 gaugeLength = 10;
14 relativeWidth = 1/2;
15 %% GRID PROPERTIES
16 nelx = 3/2*symmetricLength/gridsize;
17 nely = width/gridsize;
18 nelBolt = rBolt/gridsize;
19
20 nGaugeX = gaugeLength/gridsize;
21 nGaugeY = ceil(nely*relativeWidth);
22
23 nMount = mountingPlatform/gridsize;
24 springCstIn = 10800/gridsize;
25 springCstOut = 0.0908/gridsize;
26 %% OPTIMIZATION PARAMETERS
27 volfrac = 0.5;
28 penal = 3.0;
29 rmin = 0.03*nelx;
30 threshold = 1/2;
31 %% OUTPUT CONTROL
32 plotFiguresDuring = 'yes';
```

A. MATLAB CODE: STANDARD TOPOLOGY OPTIMIZATION

```
33 plotFiguresEnd = 'yes';
34 saveFigures = 'yes';
35 savingDirectory = ''; % Defined a directory to save the results
36 %% TOPOLOGY OPTIMIZATION
37 [xPhys,c,upf,U,inDof,outDof] = topStandard(nelx,nely,nelBolt, ...
38     nGaugeX,nGaugeY,nMount,springCstIn,springCstOut, ...
39     volfrac,penal,rmin,threshold, ...
40     plotFiguresDuring,plotFiguresEnd,saveFigures,savingDirectory);
41 %% SAVE OUTPUT DATA
42 xPhysData = reshape(xPhys,nely*nelx,1);
43 save(strcat(savingDirectory,'xPhys.txt'),'xPhysData','-ascii');
44 save(strcat(savingDirectory,'inDof.txt'),'inDof','-ascii');
45 save(strcat(savingDirectory,'outDof.txt'),'outDof','-ascii');
46 inputData = [nelx;nely;gridsize;length(inDof);length(outDof);c;upf];
47 save(strcat(savingDirectory,'input.txt'),'inputData','-ascii');
```

A.2 Optimization Code

```
1 function [xPhys,c,upf,U,inDof,outDof] = topStandard(nelx,nely, ...
2     nelBolt,nGaugeX,nGaugeY,nMount,springCstIn,springCstOut, ...
3     volfrac,penal,rmin,eta, ...
4     plotFiguresDuring,plotFiguresEnd,saveFigures,savingDirectory)
5 %% MATERIAL PROPERTIES
6 E0 = 69;
7 Emin = 1e-9;
8 nu = 0.33;
9
10 %% INITIALIZE MMA
11 x = repmat(volfrac,nely,nelx);
12 nel = nely*nelx;
13
14 m = 1;
15 n = nely*nelx;
16 xmin = zeros(nel,1);
17 xmax = ones(nel,1);
18 xval = reshape(x,nel,1);
19 xold1 = xval;
20 xold2 = xval;
21 u = xval;
22 a0 = 1;
23 a = [ones(m-1,1);0];
24 c_constant = 1000*ones(m,1);
25 d_constant = zeros(m,1);
26 low = xmin;
27 upp = xmax;
28 %% PREPARE FINITE ELEMENT ANALYSIS
29 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
30 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
31 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
32 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
33 KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
```

```

34 nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
35 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
36 edofMat = repmat(edofVec,1,8) ...
37     +repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
38 iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
39 jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
40 %% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM) AND ...
41 % PASSIVE AND ACTIVE ELEMENTS
42 fixeddofs = [1:2:2*(nely+1)];
43 passact = sparse(nely,nelx);
44 inDof = [];
45 outDof = [edofMat(nGaugeX*nely+nGaugeY,1)];
46 for i = 1:nelx
47     for j = 1:nely
48         if i <= nGaugeX+nMount && ...
49             j > nGaugeY-nMount/2 && j <= nGaugeY+nMount/2
50             if i > nGaugeX
51                 passact(j,i) = 2;
52             else
53                 passact(j,i) = 1;
54             end
55         end
56         if sqrt((j-nely/2-0.5)^2+(i-nelx*2/3-0.5)^2) < nelBolt
57             passact(j,i) = 1;
58         elseif sqrt((j-nely/2-0.5)^2+(i-nelx*2/3-0.5)^2) < nelBolt+2
59             passact(j,i) = 2;
60             inDof = union(inDof,edofMat(nely*i-(nely-j),[1:2:7]));
61             fixeddofs = ...
62                 union(fixeddofs,[edofMat(nely*i-(nely-j),[2:2:8])]);
63         end
64     end
65 end
66 KSpringIn = sparse(inDof,inDof, springCstIn/length(inDof), ...
67     2*(nely+1)*(nelx+1),2*(nely+1)*(nelx+1));
68 KSpringOut = sparse(outDof,outDof, springCstOut/length(outDof), ...
69     2*(nely+1)*(nelx+1),2*(nely+1)*(nelx+1));
70 F = sparse(inDof,1,270/length(inDof),2*(nely+1)*(nelx+1),1);
71 U = zeros(2*(nely+1)*(nelx+1),1);
72 lMat = sparse(outDof,1,1,2*(nely+1)*(nelx+1),1);
73 lambda = sparse(2*(nely+1)*(nelx+1),1);
74 alldofs = [1:2*(nely+1)*(nelx+1)];
75 freedofs = setdiff(alldofs,fixeddofs);
76 %% PREPARE FILTER
77 iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
78 jH = ones(size(iH));
79 sH = zeros(size(iH));
80 k = 0;
81 for i1 = 1:nelx
82     for j1 = 1:nely
83         e1 = (i1-1)*nely+j1;
84         for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
85             for j2 = max(j1-(ceil(rmin)-1),1): ...
86                 min(j1+(ceil(rmin)-1),nely)
87                 e2 = (i2-1)*nely+j2;
88                 k = k+1;

```

A. MATLAB CODE: STANDARD TOPOLOGY OPTIMIZATION

```

89         iH(k) = e1;
90         jH(k) = e2;
91         sH(k) = max(0, rmin-sqrt((i1-i2)^2+(j1-j2)^2));
92     end
93 end
94 end
95 end
96 H = sparse(iH, jH, sH);
97 Hs = sum(H, 2);
98 %% INITIALIZE ITERATION
99 beta = 1;
100 xTilde = x;
101 loop = 0;
102 change = 1;
103 if strcmpi(plotFiguresDuring, 'yes') || strcmpi(plotFiguresEnd, 'yes')
104     designPlot = figure;
105     designPlot.IntegerHandle = 'off'; designPlot.Name = 'Design';
106 end
107 %% START ITERATION
108 while change > 0.05
109     loop = loop+1;
110     %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND ...
111     %% PHYSICAL DENSITIES USING MMA
112     if loop ~= 1
113         xval = reshape(x, nely, 1);
114         fval = [sum(sum(xPhys))/(volfrac*nely*nelyx)-1];
115         dfdx = [ones(1, nely)/(volfrac*nely*nelyx)];
116         f0val = c;
117         df0dx = reshape(dc, nely, 1);
118         df0dx2 = 0*df0dx;
119         dfdx2 = 0*dfdx;
120         [x, ~, ~, ~, ~, ~, ~, ~, ~, ~, low, upp, ~, ~] = ...
121             mmasub(m, n, loop, xval, xmin, xmax, xold1, xold2, f0val, ...
122                 df0dx, df0dx2, fval, dfdx, dfdx2, low, upp, ...
123                 a0, a, c_constant, d_constant);
124         xold2 = xold1;
125         xold1 = u;
126         u = x;
127         x = reshape(x, nely, nelyx);
128         x(passact==1) = 0;
129         x(passact==2) = 1;
130         xTilde(:) = (H*x(:))./Hs;
131     end
132     xPhys = ((tanh(beta*eta)+tanh(beta*(xTilde-eta)))/ ...
133             (tanh(beta*eta)+tanh(beta*(1-eta))));
134     xPhys(passact==1) = 0;
135     xPhys(passact==2) = 1;
136     %% FE-ANALYSIS
137     sK = reshape(KE(:)*(Emin+xPhys(:)).^penal*(E0-Emin), ...
138                 64*nely*nelyx, 1);
139     K = sparse(iK, jK, sK) + KSpringIn + KSpringOut; K = (K+K')/2;
140     U(freedofs) = K(freedofs, freedofs)\F(freedofs);
141     %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
142     lambda(freedofs, 1) = - K(freedofs, freedofs)\lMat(freedofs);
143     ce = reshape(sum((lambda(edofMat)*KE).*U(edofMat), 2), nely, nelyx);

```

```

144     c = lMat(freedofs)'*U(freedofs)+100;
145     dc = penal*(E0-Emin)*xPhys.^(penal-1).*ce;
146     dv = ones(nely,nelx);
147     %% FILTERING/MODIFICATION OF SENSITIVITIES
148     dx = (beta*(1-(tanh(beta*(xTilde-eta))).^2)/ ...
149           (tanh(beta*eta)+tanh(beta*(1-eta))));
150     dc(:) = H*(dc(:).*dx(:)./Hs);
151     dv(:) = H*(dv(:).*dx(:)./Hs);
152     %% CALCULATE OUTPUT
153     change = max(max(abs(x-reshape(xold1,nely,nelx))));
154     upf = -(U(outDof(1))/nGaugeX)/(mean(U(inDof))/(2/3*(nelx)));
155     Mnd = sum(4*reshape(xPhys(:),nel,1).* ...
156             (1-reshape(xPhys(:),nel,1)))/nel*100;
157     %% PRINT RESULTS
158     fprintf(' It.:%5i      Obj.:%7.4f      Upf.:%7.3f      Vol.:%7.3f ...
159             Mnd.:%7.1f%%      Ch.:%7.3f\n', ...
160             loop,c,upf,mean(xPhys(:)),Mnd,change);
161     %% PLOT DENSITIES
162     if strcmpi(plotFiguresDuring,'yes')
163         figure(designPlot);
164         colormap(gray); imagesc(1-xPhys); caxis([0 1]); ...
165         axis equal; axis off; drawnow;
166     end
167     %% UPDATE HEAVISIDE REGULARIZATION PARAMETER
168     if beta < 25
169         beta = 1.01*beta;
170         change = 1;
171         fprintf('Parameter beta increased to %g.\n',beta);
172     end
173     %% PLOT DEFORMATIONS OF THE DESIGNS
174     if strcmpi(plotFiguresEnd,'yes')
175         if strcmpi(plotFiguresDuring,'no')
176             figure(designPlot);
177             colormap(gray); imagesc(1-xPhys); caxis([0 1]); ...
178             axis equal; axis off; drawnow;
179         end
180         designDeform = figure;
181         designDeform.IntegerHandle = 'off';
182         designDeform.Name = 'Deformation design';
183         colormap(gray); axis equal; axis off;
184         for ely = 1:nely
185             for elx = 1:nelx
186                 n1 = (nely+1)*(elx-1)+ely;
187                 n2 = (nely+1)* elx +ely;
188                 Ue = 1*U([2*n1-1;2*n1; 2*n2-1;2*n2; ...
189                         2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
190                 ly = ely-1; lx = elx-1;
191                 xx = [Ue(1,1)+lx Ue(3,1)+lx+1 Ue(5,1)+lx+1 Ue(7,1)+lx ]';
192                 yy = [Ue(2,1)-ly Ue(4,1)-ly Ue(6,1)-ly-1 Ue(8,1)-ly-1]';
193                 patch(xx,yy,-xPhys(ely,elx),'EdgeColor','none')
194             end
195         end
196         drawnow;
197         if strcmpi(saveFigures,'yes')

```

A. MATLAB CODE: STANDARD TOPOLOGY OPTIMIZATION

```
198         saveas(designPlot, strcat(savingDirectory, 'designPlot'), 'jpeg');
199         saveas(designDeform, ...
200             strcat(savingDirectory, 'designDeform'), 'jpeg');
201     end
202 end
```

Appendix B

MATLAB code: Length scale control

B.1 Input Parameters

```
1 %%%
2 % All units in kN and mm
3 %%%
4 clear;
5 close all;
6 %% DIMENSIONS SENSOR PACKAGE
7 symmetricLength = 250;
8 width = symmetricLength;
9 rBolt = 2.5;
10 mountingPlatform = 15;
11 gridsize = 2.5;
12 %% POSITION MEASURING GAUGE
13 gaugeLength = 10;
14 relativeWidth = 1/2;
15 %% GRID PROPERTIES
16 nelx = 3/2*symmetricLength/gridsize;
17 nely = width/gridsize;
18 nelBolt = rBolt/gridsize;
19
20 nGaugeX = gaugeLength/gridsize;
21 nGaugeY = ceil(nely*relativeWidth);
22
23 nMount = mountingPlatform/gridsize;
24 springCstIn = 10800/gridsize;
25 springCstOut = 0.0908/gridsize;
26 %% OPTIMIZATION PARAMETERS
27 volfrac = 0.5;
28 penal = 3.0;
29 rmin = 0.03*nelx;
30 threshold = 0.31;
31 %% OUTPUT CONTROL
32 plotFiguresDuring = 'yes';
```

B. MATLAB CODE: LENGTH SCALE CONTROL

```
33 plotFiguresEnd = 'yes';
34 saveFigures = 'yes';
35 savingDirectory = ''; % Defined a directory to save the results
36 %% TOPOLOGY OPTIMIZATION
37 [xPhys,c,upf,U,inDof,outDof] = topEID(nelx,nely,nelBolt, ...
38     nGaugeX,nGaugeY,nMount,springCstIn,springCstOut, ...
39     volfrac,penal,rmin,threshold, ...
40     plotFiguresDuring,plotFiguresEnd,saveFigures,savingDirectory);
41 %% SAVE OUTPUT DATA
42 xPhysData = reshape(xPhys,nely*nelx,1);
43 save(strcat(savingDirectory,'xPhys.txt'),'xPhysData','-ascii');
44 save(strcat(savingDirectory,'inDof.txt'),'inDof','-ascii');
45 save(strcat(savingDirectory,'outDof.txt'),'outDof','-ascii');
46 inputData = [nelx;nely;gridsize;length(inDof);length(outDof);c;upf];
47 save(strcat(savingDirectory,'input.txt'),'inputData','-ascii');
```

B.2 Optimization Code

```
1 function [xPhys,ci,upfi,Ui,inDof,outDof] = topEID(nelx,nely, ...
2     nelBolt,nGaugeX,nGaugeY,nMount,springCstIn,springCstOut, ...
3     volfrac,penal,rmin,eta, ...
4     plotFiguresDuring,plotFiguresEnd,saveFigures,savingDirectory)
5 %% MATERIAL PROPERTIES
6 E0 = 69;
7 Emin = 1e-9;
8 nu = 0.33;
9
10 %% INITIALIZE MMA
11 z = repmat(volfrac,nely,nelx);
12 nel = nely*nelx;
13
14 m = 4;
15 n = nely*nelx;
16 zmin = zeros(nel,1);
17 zmax = ones(nel,1);
18 zval = reshape(z,nel,1);
19 zold1 = zval;
20 zold2 = zval;
21 u = zval;
22 a0 = 1;
23 a = [ones(m-1,1);0];
24 c_constant = 1000*ones(m,1);
25 d_constant = zeros(m,1);
26 low = zmin;
27 upp = zmax;
28 %% PREPARE FINITE ELEMENT ANALYSIS
29 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
30 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
31 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
32 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
33 KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
```

```

34 nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
35 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
36 edofMat = repmat(edofVec,1,8) ...
37     +repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
38 iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
39 jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
40 %% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM) AND ...
41 % PASSIVE AND ACTIVE ELEMENTS
42 fixeddofs = [1:2:2*(nely+1)];
43 passact = sparse(nely,nelx);
44 inDof = [];
45 outDof = [edofMat(nGaugeX*nely+nGaugeY,1)];
46 for i = 1:nelx
47     for j = 1:nely
48         if i <= nGaugeX+nMount && ...
49             j > nGaugeY-nMount/2 && j <= nGaugeY+nMount/2
50             if i > nGaugeX
51                 passact(j,i) = 2;
52             else
53                 passact(j,i) = 1;
54             end
55         end
56         if sqrt((j-nely/2-0.5)^2+(i-nelx*2/3-0.5)^2) < nelBolt
57             passact(j,i) = 1;
58         elseif sqrt((j-nely/2-0.5)^2+(i-nelx*2/3-0.5)^2) < nelBolt+2
59             passact(j,i) = 2;
60             inDof = union(inDof,edofMat(nely*i-(nely-j),[1:2:7]));
61             fixeddofs = ...
62                 union(fixeddofs,[edofMat(nely*i-(nely-j),[2:2:8])]);
63         end
64     end
65 end
66 KSpringIn = sparse(inDof,inDof, springCstIn/length(inDof), ...
67     2*(nely+1)*(nelx+1),2*(nely+1)*(nelx+1));
68 KSpringOut = sparse(outDof,outDof, springCstOut/length(outDof), ...
69     2*(nely+1)*(nelx+1),2*(nely+1)*(nelx+1));
70 F = sparse(inDof,1,270/length(inDof),2*(nely+1)*(nelx+1),1);
71 Ue = zeros(2*(nely+1)*(nelx+1),1);
72 Ui = zeros(2*(nely+1)*(nelx+1),1);
73 Ud = zeros(2*(nely+1)*(nelx+1),1);
74 lMat = sparse(outDof,1,1,2*(nely+1)*(nelx+1),1);
75 lambdae=sparse(2*(nely+1)*(nelx+1),1); lambdai=lambdae; lambda d=lambdae;
76 alldofs = [1:2*(nely+1)*(nelx+1)];
77 freedofs = setdiff(alldofs,fixeddofs);
78 %% PREPARE FILTER
79 iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
80 jH = ones(size(iH));
81 sH = zeros(size(iH));
82 k = 0;
83 for il = 1:nelx
84     for jl = 1:nely
85         e1 = (il-1)*nely+jl;
86         for i2 = max(il-(ceil(rmin)-1),1):min(il+(ceil(rmin)-1),nelx)
87             for j2 = max(jl-(ceil(rmin)-1),1): ...
88                 min(jl+(ceil(rmin)-1),nely)

```

B. MATLAB CODE: LENGTH SCALE CONTROL

```

89             e2 = (i2-1)*nely+j2;
90             k = k+1;
91             iH(k) = e1;
92             jH(k) = e2;
93             sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
94         end
95     end
96 end
97 end
98 H = sparse(iH,jH,sH);
99 Hs = sum(H,2);
100 %% INITIALIZE ITERATION
101 VMax = volfrac*nely*nely;
102 VdMax = volfrac*nely*nely;
103 beta = 1;
104 zTilde = z;
105 loop = 0;
106 change = 1;
107 if strcmpi(plotFiguresDuring,'yes')||strcmpi(plotFiguresEnd,'yes')
108     erodedPlot = figure;
109     erodedPlot.IntegerHandle = 'off';
110     erodedPlot.Name = 'Eroded design';
111     intermediatePlot = figure;
112     intermediatePlot.IntegerHandle = 'off';
113     intermediatePlot.Name = 'Intermediate design';
114     delatedPlot = figure;
115     delatedPlot.IntegerHandle = 'off';
116     delatedPlot.Name = 'Delated design';
117 end
118 cd = 0;
119 %% START ITERATION
120 while change > 0.05
121     loop = loop+1;
122     %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES ...
123     %AND PHYSICAL DENSITIES USING MMA
124     if loop ~= 1
125         zval = reshape(z,nel,1);
126         fval = [ce;
127             ci;
128             cd;
129             sum(sum(xPhysMean))/VdMax-1];
130         dfdx = [reshape(dce,nel,1)';
131             reshape(dci,nel,1)';
132             reshape(dcd,nel,1)';
133             ones(1,nel)/VdMax];
134         f0val = 0;
135         df0dx = 0*reshape(dci,nel,1);
136         df0dx2 = 0*df0dx;
137         dfdx2 = 0*dfdx;
138         [z,~,~,~,~,~,~,~,~,low,upp,~,~] = ...
139             mmasub(m,n,loop,zval,zmin,zmax,zold1,zold2,f0val, ...
140                 df0dx,df0dx2,fval,dfdx,dfdx2,low,upp, ...
141                 a0,a,c_constant,d_constant);
142         zold2 = zold1;
143         zold1 = u;

```

```

144     u = z;
145     z = reshape(z,nely,nelx);
146     z(passact==1) = 0;
147     z(passact==2) = 1;
148     zTilde(:) = (H*z(:))./Hs;
149 end
150 xPhyse = ((tanh(beta*(1-eta))+tanh(beta*(zTilde-(1-eta))))/ ...
151     (tanh(beta*(1-eta))+tanh(beta*(1-(1-eta)))));
152 xPhysi = ((tanh(beta*1/2)+tanh(beta*(zTilde-1/2)))/ ...
153     (tanh(beta*1/2)+tanh(beta*(1-1/2))));
154 xPhysd = ((tanh(beta*eta)+tanh(beta*(zTilde-eta)))/ ...
155     (tanh(beta*eta)+tanh(beta*(1-eta))));
156 xPhyse(passact==1) = 0;
157 xPhyse(passact==2) = 1;
158 xPhysi(passact==1) = 0;
159 xPhysi(passact==2) = 1;
160 xPhysd(passact==1) = 0;
161 xPhysd(passact==2) = 1;
162 %% FE-ANALYSIS
163 sKe = reshape(KE(:)*(Emin+xPhyse(:)'.^penal*(E0-Emin)), ...
164     64*nelx*nely,1);
165 Ke = sparse(iK,jK,sKe) + KSpringIn + KSpringOut; Ke = (Ke+Ke')/2;
166 Ue(freedofs) = Ke(freedofs,freedofs)\F(freedofs);
167 sKi = reshape(KE(:)*(Emin+xPhysi(:)'.^penal*(E0-Emin)), ...
168     64*nelx*nely,1);
169 Ki = sparse(iK,jK,sKi) + KSpringIn + KSpringOut; Ki = (Ki+Ki')/2;
170 Ui(freedofs) = Ki(freedofs,freedofs)\F(freedofs);
171 sKd = reshape(KE(:)*(Emin+xPhysd(:)'.^penal*(E0-Emin)), ...
172     64*nelx*nely,1);
173 Kd = sparse(iK,jK,sKd) + KSpringIn + KSpringOut; Kd = (Kd+Kd')/2;
174 Ud(freedofs) = Kd(freedofs,freedofs)\F(freedofs);
175 %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
176 lambdae(freedofs,1) = - Ke(freedofs,freedofs)\lMat(freedofs);
177 cee = ...
178     reshape(sum((lambdae(edofMat)*KE).*Ue(edofMat),2),nely,nelx);
179 ce = lMat(freedofs)'*Ue(freedofs)+100;
180 dce = penal*(E0-Emin)*xPhyse.^(penal-1).*cee;
181 dve = ones(nely,nelx);
182 lambdai(freedofs,1) = - Ki(freedofs,freedofs)\lMat(freedofs);
183 cei = ...
184     reshape(sum((lambdai(edofMat)*KE).*Ui(edofMat),2),nely,nelx);
185 ci = lMat(freedofs)'*Ui(freedofs)+100;
186 dci = penal*(E0-Emin)*xPhysi.^(penal-1).*cei;
187 dvi = ones(nely,nelx);
188 lambda d(freedofs,1) = - Kd(freedofs,freedofs)\lMat(freedofs);
189 ced = ...
190     reshape(sum((lambda d(edofMat)*KE).*Ud(edofMat),2),nely,nelx);
191 cd = lMat(freedofs)'*Ud(freedofs)+100;
192 dcd = penal*(E0-Emin)*xPhysd.^(penal-1).*ced;
193 dvd = ones(nely,nelx);
194 %% FILTERING/MODIFICATION OF SENSITIVITIES
195 dx = (beta*(1-(tanh(beta*(zTilde-(1-eta))))).^2)/ ...
196     (tanh(beta*(1-eta))+tanh(beta*(1-(1-eta)))));
197 dce(:) = H*(dce(:).*dx(:))./Hs;
198 dve(:) = H*(dve(:).*dx(:))./Hs;

```

B. MATLAB CODE: LENGTH SCALE CONTROL

```

199     dxi = (beta*(1-(tanh(beta*(zTilde-1/2))).^2)/ ...
200         (tanh(beta*1/2)+tanh(beta*(1-1/2))));
201     dci(:) = H*(dci(:).*dxi(:)./Hs);
202     dvi(:) = H*(dvi(:).*dxi(:)./Hs);
203     dxd = (beta*(1-(tanh(beta*(zTilde-eta))).^2)/ ...
204         (tanh(beta*eta)+tanh(beta*(1-eta))));
205     dcd(:) = H*(dcd(:).*dxd(:)./Hs);
206     dvd(:) = H*(dvd(:).*dxd(:)./Hs);
207     %% CALCULATE OUTPUT
208     change = max(max(abs(z-reshape(zold1,nely,nelx))));
209     upfe = -(Ue(outDof(1))/nGaugeX)/(mean(Ue(inDof))/(2/3*(nelx)));
210     upfi = -(Ui(outDof(1))/nGaugeX)/(mean(Ui(inDof))/(2/3*(nelx)));
211     upfd = -(Ud(outDof(1))/nGaugeX)/(mean(Ud(inDof))/(2/3*(nelx)));
212     Mnde = sum(4*reshape(xPhyse(:),nel,1).* ...
213         (1-reshape(xPhyse(:),nel,1)))/nel*100;
214     Mndi = sum(4*reshape(xPhysi(:),nel,1).* ...
215         (1-reshape(xPhysi(:),nel,1)))/nel*100;
216     Mndd = sum(4*reshape(xPhysd(:),nel,1).* ...
217         (1-reshape(xPhysd(:),nel,1)))/nel*100;
218     %% PRINT RESULTS
219     fprintf(' It.:%5i      Obj.:%7.4f      Upf.:%7.3f      Vol.:%7.3f ...
220             Mnd.:%7.1f%%      Ch.:%7.3f\n', ...
221         loop,ce,upfe,mean(xPhyse(:)),Mnde,change);
222     fprintf('%27.4f %16.3f %16.3f %16.1f%%\n', ...
223         ci,upfi,mean(xPhysi(:)),Mndi);
224     fprintf('%27.4f %16.3f %16.3f %16.1f%%\n', ...
225         cd,upfd,mean(xPhysd(:)),Mndd);
226     %% PLOT DENSITIES
227     if strcmpi(plotFiguresDuring,'yes')
228         figure(erodedPlot);
229         colormap(gray); imagesc(1-xPhyse); caxis([0 1]); ...
230         axis equal; axis off; drawnow;
231         figure(intermediatePlot);
232         colormap(gray); imagesc(1-xPhysi); caxis([0 1]); ...
233         axis equal; axis off; drawnow;
234         figure(delatedPlot);
235         colormap(gray); imagesc(1-xPhysd); caxis([0 1]); ...
236         axis equal; axis off; drawnow;
237     end
238     %% UPDATE VOLUME FRACTION
239     xPhysTotal(:,:,1) = xPhyse;
240     xPhysTotal(:,:,2) = xPhysi;
241     xPhysTotal(:,:,3) = xPhysd;
242     xPhysMean = mean(xPhysTotal,3);
243     %% UPDATE HEAVISIDE REGULARIZATION PARAMETER
244     if beta < 25
245         beta = 1.01*beta;
246         change = 1;
247         fprintf('Parameter beta increased to %g.\n',beta);
248     end
249     %% PLOT DEFORMATIONS OF THE DESIGNS
250     if strcmpi(plotFiguresEnd,'yes')
251         if strcmpi(plotFiguresDuring,'no')
252             figure(erodedPlot);

```

```

253     colormap(gray); imagesc(1-xPhyse); caxis([0 1]); ...
254     axis equal; axis off; drawnow;
255     figure(intermediatePlot);
256     colormap(gray); imagesc(1-xPhysi); caxis([0 1]); ...
257     axis equal; axis off; drawnow;
258     figure(delatedPlot);
259     colormap(gray); imagesc(1-xPhysd); caxis([0 1]); ...
260     axis equal; axis off; drawnow;
261 end
262 erodedDeform = figure;
263 erodedDeform.IntegerHandle = 'off';
264 erodedDeform.Name = 'Deformation eroded design';
265 colormap(gray); axis equal; axis off;
266 for ely = 1:nely
267     for elx = 1:nelx
268         n1 = (nely+1)*(elx-1)+ely;
269         n2 = (nely+1)* elx +ely;
270         Uee = 1*Ue([2*n1-1;2*n1; 2*n2-1;2*n2; ...
271                 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
272         ly = ely-1; lx = elx-1;
273         xx = [Uee(1,1)+lx Uee(3,1)+lx+1 ...
274               Uee(5,1)+lx+1 Uee(7,1)+lx ]';
275         yy = [Uee(2,1)-ly Uee(4,1)-ly ...
276               Uee(6,1)-ly-1 Uee(8,1)-ly-1]';
277         patch(xx,yy,-xPhyse(ely,elx), 'EdgeColor', 'none')
278     end
279 end
280 drawnow;
281 intermediateDeform = figure;
282 intermediateDeform.IntegerHandle = 'off';
283 intermediateDeform.Name = 'Deformation intermediate design';
284 colormap(gray); axis equal; axis off;
285 for ely = 1:nely
286     for elx = 1:nelx
287         n1 = (nely+1)*(elx-1)+ely;
288         n2 = (nely+1)* elx +ely;
289         Uei = 1*Ui([2*n1-1;2*n1; 2*n2-1;2*n2; ...
290                  2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
291         ly = ely-1; lx = elx-1;
292         xx = [Uei(1,1)+lx Uei(3,1)+lx+1 ...
293               Uei(5,1)+lx+1 Uei(7,1)+lx ]';
294         yy = [Uei(2,1)-ly Uei(4,1)-ly ...
295               Uei(6,1)-ly-1 Uei(8,1)-ly-1]';
296         patch(xx,yy,-xPhysi(ely,elx), 'EdgeColor', 'none')
297     end
298 end
299 drawnow;
300 delatedDeform = figure;
301 delatedDeform.IntegerHandle = 'off';
302 delatedDeform.Name = 'Deformation delated design';
303 colormap(gray); axis equal; axis off;
304 for ely = 1:nely
305     for elx = 1:nelx
306         n1 = (nely+1)*(elx-1)+ely;
307         n2 = (nely+1)* elx +ely;

```

B. MATLAB CODE: LENGTH SCALE CONTROL

```
308         Ued = 1*Ud([2*n1-1;2*n1; 2*n2-1;2*n2; ...
309                 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
310         ly = ely-1; lx = elx-1;
311         xx = [Ued(1,1)+lx Ued(3,1)+lx+1 ...
312               Ued(5,1)+lx+1 Ued(7,1)+lx ]';
313         yy = [Ued(2,1)-ly Ued(4,1)-ly ...
314               Ued(6,1)-ly-1 Ued(8,1)-ly-1]';
315         patch(xx,yy,-xPhysd(ely,elx), 'EdgeColor','none')
316     end
317 end
318 drawnow;
319 if strcmpi(saveFigures,'yes')
320     saveas(erodedPlot, strcat(savingDirectory,'erodedPlot'),'jpeg');
321     saveas(intermediatePlot, ...
322           strcat(savingDirectory,'intermediatePlot'),'jpeg');
323     saveas(delatedPlot, ...
324           strcat(savingDirectory,'delatedPlot'),'jpeg');
325     saveas(erodedDeform, ...
326           strcat(savingDirectory,'erodedDeform'),'jpeg');
327     saveas(intermediateDeform, ...
328           strcat(savingDirectory,'intermediateDeform'),'jpeg');
329     saveas(delatedDeform, ...
330           strcat(savingDirectory,'delatedDeform'),'jpeg');
331 end
332 end
```

Appendix C

MATLAB code: Fibre in pure tension

C.1 Input Parameters

```
1 %%%
2 % All units in kN and mm
3 %%%
4 clear;
5 close all;
6 %% DIMENSIONS SENSOR PACKAGE
7 symmetricLength = 250;
8 width = symmetricLength;
9 rBolt = 2.5;
10 mountingPlatform = 15;
11 gridsize = 2.5;
12 %% POSITION MEASURING GAUGE
13 gaugeLength = 10;
14 relativeWidth = 1/2;
15 %% GRID PROPERTIES
16 nelx = 3/2*symmetricLength/gridsize;
17 nely = width/gridsize;
18 nelBolt = rBolt/gridsize;
19
20 nGaugeX = gaugeLength/gridsize;
21 nGaugeY = ceil(nely*relativeWidth);
22
23 nMount = mountingPlatform/gridsize;
24 springCstIn = 10800/gridsize;
25 springCstOut = 0.0908/gridsize;
26 %% OPTIMIZATION PARAMETERS
27 volfrac = 0.5;
28 penal = 3.0;
29 rmin = 0.03*nelx;
30 threshold = 1/2;
31 %% OUTPUT CONTROL
32 plotFiguresDuring = 'yes';
```

C. MATLAB CODE: FIBRE IN PURE TENSION

```
33 plotFiguresEnd = 'yes';
34 saveFigures = 'yes';
35 savingDirectory = ''; % Defined a directory to save the results
36 %% TOPOLOGY OPTIMIZATION
37 [xPhys,c,upf,U,inDof,outDof] = topUniform(nelx,nely,nelBolt, ...
38     nGaugeX,nGaugeY,nMount,springCstIn,springCstOut, ...
39     volfrac,penal,rmin,threshold, ...
40     plotFiguresDuring,plotFiguresEnd,saveFigures,savingDirectory);
41 %% SAVE OUTPUT DATA
42 xPhysData = reshape(xPhys,nely*nelx,1);
43 save(strcat(savingDirectory,'xPhys.txt'),'xPhysData','-ascii');
44 save(strcat(savingDirectory,'inDof.txt'),'inDof','-ascii');
45 save(strcat(savingDirectory,'outDof.txt'),'outDof','-ascii');
46 inputData = [nelx;nely;gridsize;length(inDof);length(outDof);c;upf];
47 save(strcat(savingDirectory,'input.txt'),'inputData','-ascii');
```

C.2 Optimization Code

```
1 function [xPhys,c1,upf1,U,inDof,outDof] = topUniform(nelx,nely, ...
2     nelBolt,nGaugeX,nGaugeY,nMount,springCstIn,springCstOut, ...
3     volfrac,penal,rmin,eta, ...
4     plotFiguresDuring,plotFiguresEnd,saveFigures,savingDirectory)
5 %% MATERIAL PROPERTIES
6 E0 = 69;
7 Emin = 1e-9;
8 nu = 0.33;
9
10 %% INITIALIZE MMA
11 z = repmat(volfrac,nely,nelx);
12 nel = nely*nelx;
13
14 m = 3;
15 n = nely*nelx;
16 zmin = zeros(nel,1);
17 zmax = ones(nel,1);
18 zval = reshape(z,nel,1);
19 zold1 = zval;
20 zold2 = zval;
21 u = zval;
22 a0 = 1;
23 a = [ones(m-1,1);0];
24 c_constant = 1000*ones(m,1);
25 d_constant = zeros(m,1);
26 low = zmin;
27 upp = ones(nel,1);
28 %% PREPARE FINITE ELEMENT ANALYSIS
29 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
30 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
31 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
32 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
33 KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
```

```

34 nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
35 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
36 edofMat = repmat(edofVec,1,8) ...
37     +repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
38 iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
39 jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
40 %% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM) AND ...
41 % PASSIVE AND ACTIVE ELEMENTS
42 fixeddofs = [1:2:2*(nely+1)];
43 passact = sparse(nely,nelx);
44 inDof = [];
45 outDof = [edofMat(nGaugeX*nely+nGaugeY-3,1) ...
46     edofMat(nGaugeX*nely+nGaugeY+3,1)];
47 for i = 1:nelx
48     for j = 1:nely
49         if i <= nGaugeX+nMount && ...
50             j > nGaugeY-nMount/2 && j <= nGaugeY+nMount/2
51             if i > nGaugeX
52                 passact(j,i) = 2;
53             else
54                 passact(j,i) = 1;
55             end
56         end
57         if sqrt((j-nely/2-0.5)^2+(i-nelx*2/3-0.5)^2) < nelBolt
58             passact(j,i) = 1;
59         elseif sqrt((j-nely/2-0.5)^2+(i-nelx*2/3-0.5)^2) < nelBolt+2
60             passact(j,i) = 2;
61             inDof = union(inDof,edofMat(nely*i-(nely-j),[1:2:7]));
62             fixeddofs = ...
63                 union(fixeddofs,[edofMat(nely*i-(nely-j),[2:2:8])]);
64         end
65     end
66 end
67 KSpringIn = sparse(inDof,inDof, springCstIn/length(inDof), ...
68     2*(nely+1)*(nelx+1),2*(nely+1)*(nelx+1));
69 KSpringOut = sparse(outDof,outDof, springCstOut/length(outDof), ...
70     2*(nely+1)*(nelx+1),2*(nely+1)*(nelx+1));
71 F = sparse(inDof,1,270/length(inDof),2*(nely+1)*(nelx+1),1);
72 U = zeros(2*(nely+1)*(nelx+1),1);
73 lMat1 = sparse(outDof,1,[1 0],2*(nely+1)*(nelx+1),1);
74 lMat2 = sparse(outDof,1,[0 1],2*(nely+1)*(nelx+1),1);
75 lambda1 = sparse(2*(nely+1)*(nelx+1),1);
76 lambda2 = sparse(2*(nely+1)*(nelx+1),1);
77 alldofs = [1:2*(nely+1)*(nelx+1)];
78 freedofs = setdiff(alldofs, fixeddofs);
79 %% PREPARE FILTER
80 iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
81 jH = ones(size(iH));
82 sH = zeros(size(iH));
83 k = 0;
84 for i1 = 1:nelx
85     for j1 = 1:nely
86         e1 = (i1-1)*nely+j1;
87         for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)

```

C. MATLAB CODE: FIBRE IN PURE TENSION

```

88         for j2 = ...
89             max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
90             e2 = (i2-1)*nely+j2;
91             k = k+1;
92             iH(k) = e1;
93             jH(k) = e2;
94             sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
95         end
96     end
97 end
98 H = sparse(iH, jH, sH);
99 Hs = sum(H, 2);
100 %% INITIALIZE ITERATION
101 beta = 1;
102 zTilde = z;
103 loop = 0;
104 change = 1;
105 if strcmpi(plotFiguresDuring, 'yes') || strcmpi(plotFiguresEnd, 'yes')
106     designPlot = figure;
107     designPlot.IntegerHandle = 'off'; designPlot.Name = 'Design';
108 end
109 %% START ITERATION
110 while change > 0.05
111     loop = loop+1;
112     %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND ...
113     % PHYSICAL DENSITIES USING MMA
114     if loop ~= 1
115         zval = reshape(z, nel, 1);
116         fval = [c1;
117             c2;
118             sum(sum(xPhys))/(volfrac*nely*nelyx)-1];
119         dfdx = [reshape(dc1, nel, 1)';
120             reshape(dc2, nel, 1)'];
121         ones(1, nel)/(volfrac*nely*nelyx)];
122         f0val = 0;
123         df0dx = 0*reshape(dc1, nel, 1);
124         df0dx2 = 0*df0dx;
125         dfdx2 = 0*dfdx;
126         [z,~,~,~,~,~,~,~,~,~,low,upp,~,~] = ...
127             mmasub(m,n,loop,zval,zmin,zmax,zold1,zold2,f0val, ...
128                 df0dx,df0dx2,fval,dfdx,dfdx2,low,upp, ...
129                 a0,a,c_constant,d_constant);
130         zold2 = zold1;
131         zold1 = u;
132         u=z;
133         z = reshape(z, nely, nelx);
134         z(passact==1) = 0;
135         z(passact==2) = 1;
136         zTilde(:) = (H*z(:))./Hs;
137     end
138     xPhys = ((tanh(beta*eta)+tanh(beta*(zTilde-eta)))/ ...
139         (tanh(beta*eta)+tanh(beta*(1-eta))));
140     xPhys(passact==1) = 0;
141     xPhys(passact==2) = 1;

```

```

142 %% FE-ANALYSIS
143 sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)), ...
144     64*nelx*nely,1);
145 K = sparse(iK,jK,sK) + KSpringIn + KSpringOut; K = (K+K')/2;
146 U(freedofs) = K(freedofs,freedofs)\F(freedofs);
147 %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
148 lambda1(freedofs,1) = - K(freedofs,freedofs)\lMat1(freedofs);
149 ce1 = reshape(sum((lambda1(edofMat)*KE).*U(edofMat),2),nely,nelx);
150 c1 = lMat1(freedofs)'*U(freedofs)+100;
151 dc1 = penal*(E0-Emin)*xPhys.^(penal-1).*ce1;
152 dv1 = ones(nely,nelx);
153 lambda2(freedofs,1) = - K(freedofs,freedofs)\lMat2(freedofs);
154 ce2 = reshape(sum((lambda2(edofMat)*KE).*U(edofMat),2),nely,nelx);
155 c2 = lMat2(freedofs)'*U(freedofs)+100;
156 dc2 = penal*(E0-Emin)*xPhys.^(penal-1).*ce2;
157 dv2 = ones(nely,nelx);
158 %% FILTERING/MODIFICATION OF SENSITIVITIES
159 dx = (beta*(1-(tanh(beta*(zTilde-eta))).^2)/ ...
160     (tanh(beta*eta)+tanh(beta*(1-eta))));
161 dc1(:) = H*(dc1(:).*dx(:)./Hs);
162 dc2(:) = H*(dc2(:).*dx(:)./Hs);
163 dv1(:) = H*(dv1(:).*dx(:)./Hs);
164 dv2(:) = H*(dv2(:).*dx(:)./Hs);
165 %% CALCULATE OUTPUT
166 change = max(max(abs(z-reshape(zold1,nely,nelx))));
167 upf1 = -(U(outDof(1))/nGaugeX)/(mean(U(inDof))/(2/3*(nelx)));
168 upf2 = -(U(outDof(2))/nGaugeX)/(mean(U(inDof))/(2/3*(nelx)));
169 Mnd = sum(4*reshape(xPhys(:),nel,1).* ...
170     (1-reshape(xPhys(:),nel,1))/nel*100;
171 %% PRINT RESULTS
172 fprintf(' It.:%5i      Obj.:%5.4f      upf.:%7.3f      Vol.:%7.3f ...
173         Mnd.:%7.1f%%      ch.:%7.3f\n', ...
174         loop,c1,upf1,mean(xPhys(:)),Mnd,change);
175 fprintf('%27.4f %16.3f\n',c2,upf2);
176 %% PLOT DENSITIES
177 if strcmpi(plotFiguresDuring,'yes')
178     figure(designPlot);
179     colormap(gray); imagesc(1-xPhys); caxis([0 1]); ...
180     axis equal; axis off; drawnow;
181 end
182 %% UPDATE HEAVISIDE REGULARIZATION PARAMETER
183 if beta < 25
184     beta = 1.01*beta;
185     change = 1;
186     fprintf('Parameter beta increased to %g.\n',beta);
187 end
188 %% PLOT DEFORMATIONS OF THE DESIGNS
189 if strcmpi(plotFiguresEnd,'yes')
190     if strcmpi(plotFiguresDuring,'no')
191         figure(designPlot);
192         colormap(gray); imagesc(1-xPhys); caxis([0 1]); ...
193         axis equal; axis off; drawnow;
194     end
195     designDeform = figure;

```

C. MATLAB CODE: FIBRE IN PURE TENSION

```
196     designDeform.IntegerHandle = 'off';
197     designDeform.Name = 'Deformation design';
198     colormap(gray); axis equal; axis off;
199     for ely = 1:nely
200         for elx = 1:nelx
201             n1 = (nely+1)*(elx-1)+ely;
202             n2 = (nely+1)* elx +ely;
203             Ue = 1*U([2*n1-1;2*n1; 2*n2-1;2*n2; ...
204                 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
205             ly = ely-1; lx = elx-1;
206             xx = [Ue(1,1)+lx Ue(3,1)+lx+1 Ue(5,1)+lx+1 Ue(7,1)+lx ]';
207             yy = [Ue(2,1)-ly Ue(4,1)-ly Ue(6,1)-ly-1 Ue(8,1)-ly-1]';
208             patch(xx,yy,-xPhys(ely,elx), 'EdgeColor', 'none')
209         end
210     end
211     drawnow;
212     if strcmpi(saveFigures, 'yes')
213         saveas(designPlot, strcat(savingDirectory, 'designPlot'), 'jpeg');
214         saveas(designDeform, ...
215             strcat(savingDirectory, 'designDeform'), 'jpeg');
216     end
217 end
```

Appendix D

MATLAB code: Combination of length scale control and fibre in pure tension

D.1 Input Parameters

```
1  %%%%%
2  % All units in kN and mm
3  %%%%%
4  clear;
5  close all;
6  %% DIMENSIONS SENSOR PACKAGE
7  symmetricLength = 250;
8  width = symmetricLength;
9  rBolt = 2.5;
10 mountingPlatform = 15;
11 gridsize = 2.5;
12 %% POSITION MEASURING GAUGE
13 gaugeLength = 10;
14 relativeWidth = 1/2;
15 %% GRID PROPERTIES
16 nelx = 3/2*symmetricLength/gridsize;
17 nely = width/gridsize;
18 nelBolt = rBolt/gridsize;
19
20 nGaugeX = gaugeLength/gridsize;
21 nGaugeY = ceil(nely*relativeWidth);
22
23 nMount = mountingPlatform/gridsize;
24 springCstIn = 10800/gridsize;
25 springCstOut = 0.0908/gridsize;
26 %% OPTIMIZATION PARAMETERS
27 volfrac = 0.5;
28 penal = 3.0;
29 rmin = 0.03*nelx;
```

D. MATLAB CODE: COMBINATION OF LENGTH SCALE CONTROL AND FIBRE IN PURE TENSION

```
30 threshold = 0.31;
31 %% OUTPUT CONTROL
32 plotFiguresDuring = 'yes';
33 plotFiguresEnd = 'yes';
34 saveFigures = 'yes';
35 savingDirectory = ''; % Defined a directory to save the results
36 %% TOPOLOGY OPTIMIZATION
37 [xPhys,c,upf,U,inDof,outDof] = topCombination(nelx,nely,nelBolt, ...
38     nGaugeX,nGaugeY,nMount,springCstIn,springCstOut, ...
39     volfrac,penal,rmin,threshold, ...
40     plotFiguresDuring,plotFiguresEnd,saveFigures,savingDirectory);
41 %% SAVE OUTPUT DATA
42 xPhysData = reshape(xPhys,nely*nelx,1);
43 save(strcat(savingDirectory,'xPhys.txt'),'xPhysData','-ascii');
44 save(strcat(savingDirectory,'inDof.txt'),'inDof','-ascii');
45 save(strcat(savingDirectory,'outDof.txt'),'outDof','-ascii');
46 inputData = [nelx;nely;gridsize:length(inDof);length(outDof);c;upf];
47 save(strcat(savingDirectory,'input.txt'),'inputData','-ascii');
```

D.2 Optimization Code

```
1 function [xPhysi,cil,upfil,Ui,inDof,outDof] = topCombination(nelx, ...
2     nely,nelBolt,nGaugeX,nGaugeY,nMount,springCstIn,springCstOut, ...
3     volfrac,penal,rmin,eta, ...
4     plotFiguresDuring,plotFiguresEnd,saveFigures,savingDirectory)
5 %% MATERIAL PROPERTIES
6 E0 = 69;
7 Emin = 1e-9;
8 nu = 0.33;
9
10 %% INITIALIZE MMA
11 z = repmat(volfrac,nely,nelx);
12 nel = nely*nelx;
13
14 m = 7;
15 n = nely*nelx;
16 zmin = zeros(nel,1);
17 zmax = ones(nel,1);
18 zval = reshape(z,nel,1);
19 zold1 = zval;
20 zold2 = zval;
21 u = zval;
22 a0 = 1;
23 a = [ones(m-1,1);0];
24 c_constant = 1000*ones(m,1);
25 d_constant = zeros(m,1);
26 low = zmin;
27 upp = zmax;
28 %% PREPARE FINITE ELEMENT ANALYSIS
29 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
30 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
```

```

31 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
32 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
33 KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
34 nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
35 edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
36 edofMat = repmat(edofVec,1,8)...
37     +repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
38 iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
39 jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
40 %% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM) AND ...
41 % PASSIVE AND ACTIVE ELEMENTS
42 fixeddofs = [1:2:2*(nely+1)];
43 passact = sparse(nely,nelx);
44 inDof = [];
45 outDof = [edofMat(nGaugeX*nely+nGaugeY-3,1) ...
46     edofMat(nGaugeX*nely+nGaugeY+3,1)];
47 for i = 1:nelx
48     for j = 1:nely
49         if i <= nGaugeX+nMount && ...
50             j > nGaugeY-nMount/2 && j <= nGaugeY+nMount/2
51             if i > nGaugeX
52                 passact(j,i) = 2;
53             else
54                 passact(j,i) = 1;
55             end
56         end
57         if sqrt((j-nely/2-0.5)^2+(i-nelx*2/3-0.5)^2) < nelBolt
58             passact(j,i) = 1;
59         elseif sqrt((j-nely/2-0.5)^2+(i-nelx*2/3-0.5)^2) < nelBolt+2
60             passact(j,i) = 2;
61             inDof = union(inDof,edofMat(nely*i-(nely-j),[1:2:7]));
62             fixeddofs = ...
63                 union(fixeddofs,[edofMat(nely*i-(nely-j),[2:2:8])]);
64         end
65     end
66 end
67 KSpringIn = sparse(inDof,inDof, springCstIn/length(inDof), ...
68     2*(nely+1)*(nelx+1),2*(nely+1)*(nelx+1));
69 KSpringOut = sparse(outDof,outDof, springCstOut/length(outDof), ...
70     2*(nely+1)*(nelx+1),2*(nely+1)*(nelx+1));
71 F = sparse(inDof,1,270/length(inDof),2*(nely+1)*(nelx+1),1);
72 Ue = zeros(2*(nely+1)*(nelx+1),1);
73 Ui = zeros(2*(nely+1)*(nelx+1),1);
74 Ud = zeros(2*(nely+1)*(nelx+1),1);
75 lMat1 = sparse(outDof(1),1,1,2*(nely+1)*(nelx+1),1);
76 lMat2 = sparse(outDof(2),1,1,2*(nely+1)*(nelx+1),1);
77 lambdae1 = sparse(2*(nely+1)*(nelx+1),1); lambdai1 = lambdae1;
78 lambdaad1 = lambdae1; lambdae2 = lambdae1; lambdai2 = lambdae1;
79 lambdaad2 = lambdae1;
80 alldofs = [1:2*(nely+1)*(nelx+1)];
81 freedofs = setdiff(alldofs, fixeddofs);
82 %% PREPARE FILTER
83 iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
84 jH = ones(size(iH));
85 sH = zeros(size(iH));

```

D. MATLAB CODE: COMBINATION OF LENGTH SCALE CONTROL AND FIBRE IN PURE TENSION

```

86 k = 0;
87 for i1 = 1:nelx
88     for j1 = 1:nely
89         e1 = (i1-1)*nely+j1;
90         for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
91             for j2 = max(j1-(ceil(rmin)-1),1): ...
92                 min(j1+(ceil(rmin)-1),nely)
93                 e2 = (i2-1)*nely+j2;
94                 k = k+1;
95                 iH(k) = e1;
96                 jH(k) = e2;
97                 sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
98             end
99         end
100     end
101 end
102 H = sparse(iH,jH,sH);
103 Hs = sum(H,2);
104 %% INITIALIZE ITERATION
105 VMax = volfrac*nely*nelx;
106 VdMax = volfrac*nely*nelx;
107 beta = 1;
108 zTilde = z;
109 loop = 0;
110 change = 1;
111 if strcmpi(plotFiguresDuring,'yes')||strcmpi(plotFiguresEnd,'yes')
112     erodedPlot = figure;
113     erodedPlot.IntegerHandle = 'off';
114     erodedPlot.Name = 'Eroded design';
115     intermediatePlot = figure;
116     intermediatePlot.IntegerHandle = 'off';
117     intermediatePlot.Name = 'Intermediate design';
118     delatedPlot = figure;
119     delatedPlot.IntegerHandle = 'off';
120     delatedPlot.Name = 'Delated design';
121 end
122 %% START ITERATION
123 while change > 0.05
124     loop = loop+1;
125     %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND ...
126     % PHYSICAL DENSITIES USING MMA
127     if loop ~= 1
128         zval = reshape(z,nel,1);
129         fval = [ce1;
130                 ce2;
131                 ci1;
132                 ci2;
133                 cd1;
134                 cd2;
135                 sum(sum(xPhysMean))/VdMax-1];
136         dfdx = [reshape(dce1,nel,1)';
137                 reshape(dce2,nel,1)';
138                 reshape(dci1,nel,1)';
139                 reshape(dci2,nel,1)';
140                 reshape(dcd1,nel,1)'];

```

```

141         reshape(dcd2,nel,1)';
142         ones(1,nel)/VdMax];
143     f0val = 0;
144     df0dx = 0*reshape(dci1,nel,1);
145     df0dx2 = 0*df0dx;
146     dfdx2 = 0*dfdx;
147     [z,~,~,~,~,~,~,~,~,low,upp,~,~] = ...
148         mmasub(m,n,loop,zval,zmin,zmax,zold1,zold2,f0val, ...
149             df0dx,df0dx2,fval,dfdx,dfdx2,low,upp, ...
150             a0,a,c_constant,d_constant);
151     zold2 = zold1;
152     zold1 = u;
153     u = z;
154     z = reshape(z,nely,nelx);
155     z(passact==1) = 0;
156     z(passact==2) = 1;
157     zTilde(:) = (H*z(:))./Hs;
158 end
159 xPhyse = ((tanh(beta*(1-eta))+tanh(beta*(zTilde-(1-eta))))/ ...
160     (tanh(beta*(1-eta))+tanh(beta*(1-(1-eta)))));
161 xPhysi = ((tanh(beta*1/2)+tanh(beta*(zTilde-1/2)))/ ...
162     (tanh(beta*1/2)+tanh(beta*(1-1/2))));
163 xPhysd = ((tanh(beta*eta)+tanh(beta*(zTilde-eta)))/ ...
164     (tanh(beta*eta)+tanh(beta*(1-eta))));
165 xPhyse(passact==1) = 0;
166 xPhyse(passact==2) = 1;
167 xPhysi(passact==1) = 0;
168 xPhysi(passact==2) = 1;
169 xPhysd(passact==1) = 0;
170 xPhysd(passact==2) = 1;
171 %% FE-ANALYSIS
172 sKe = reshape(KE(:)*(Emin+xPhyse(:)'.^penal*(E0-Emin)), ...
173     64*nelx*nely,1);
174 Ke = sparse(iK,jK,sKe) + KSpringIn + KSpringOut; Ke = (Ke+Ke')/2;
175 Ue(freedofs) = Ke(freedofs,freedofs)\F(freedofs);
176 sKi = reshape(KE(:)*(Emin+xPhysi(:)'.^penal*(E0-Emin)), ...
177     64*nelx*nely,1);
178 Ki = sparse(iK,jK,sKi) + KSpringIn + KSpringOut; Ki = (Ki+Ki')/2;
179 Ui(freedofs) = Ki(freedofs,freedofs)\F(freedofs);
180 sKd = reshape(KE(:)*(Emin+xPhysd(:)'.^penal*(E0-Emin)), ...
181     64*nelx*nely,1);
182 Kd = sparse(iK,jK,sKd) + KSpringIn + KSpringOut; Kd = (Kd+Kd')/2;
183 Ud(freedofs) = Kd(freedofs,freedofs)\F(freedofs);
184 %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
185 lambdae1(freedofs,1) = - Ke(freedofs,freedofs)\lMat1(freedofs);
186 ceel = ...
187     reshape(sum((lambdae1(edofMat)*KE).*Ue(edofMat),2),nely,nelx);
188 ce1 = lMat1(freedofs)'*Ue(freedofs)+100;
189 dce1 = penal*(E0-Emin)*xPhyse.^(penal-1).*ceel;
190 dve1 = ones(nely,nelx);
191 lambdae2(freedofs,1) = - Ke(freedofs,freedofs)\lMat2(freedofs);
192 ce2 = ...
193     reshape(sum((lambdae2(edofMat)*KE).*Ue(edofMat),2),nely,nelx);
194 ce2 = lMat2(freedofs)'*Ue(freedofs)+100;
195 dce2 = penal*(E0-Emin)*xPhyse.^(penal-1).*ce2;

```

D. MATLAB CODE: COMBINATION OF LENGTH SCALE CONTROL AND FIBRE IN PURE TENSION

```

196     dve2 = ones(nely,nelx);
197     lambdai1(freedofs,1) = - Ki(freedofs,freedofs)\lMat1(freedofs);
198     cei1 = ...
199         reshape(sum((lambdai1(edofMat)*KE).*Ui(edofMat),2),nely,nelx);
200     ci1 = lMat1(freedofs)'*Ui(freedofs)+100;
201     dci1 = penal*(E0-Emin)*xPhysi.^(penal-1).*cei1;
202     dvi1 = ones(nely,nelx);
203     lambdai2(freedofs,1) = - Ki(freedofs,freedofs)\lMat2(freedofs);
204     cei2 = ...
205         reshape(sum((lambdai2(edofMat)*KE).*Ui(edofMat),2),nely,nelx);
206     ci2 = lMat2(freedofs)'*Ui(freedofs)+100;
207     dci2 = penal*(E0-Emin)*xPhysi.^(penal-1).*cei2;
208     dvi2 = ones(nely,nelx);
209     lambdad1(freedofs,1) = - Kd(freedofs,freedofs)\lMat1(freedofs);
210     ced1 = ...
211         reshape(sum((lambdad1(edofMat)*KE).*Ud(edofMat),2),nely,nelx);
212     cd1 = lMat1(freedofs)'*Ud(freedofs)+100;
213     dcd1 = penal*(E0-Emin)*xPhysd.^(penal-1).*ced1;
214     dvd1 = ones(nely,nelx);
215     lambdad2(freedofs,1) = - Kd(freedofs,freedofs)\lMat2(freedofs);
216     ced2 = ...
217         reshape(sum((lambdad2(edofMat)*KE).*Ud(edofMat),2),nely,nelx);
218     cd2 = lMat2(freedofs)'*Ud(freedofs)+100;
219     dcd2 = penal*(E0-Emin)*xPhysd.^(penal-1).*ced2;
220     dvd2 = ones(nely,nelx);
221     %% FILTERING/MODIFICATION OF SENSITIVITIES
222     dx = (beta*(1-(tanh(beta*(zTilde-(1-eta))))).^2) / ...
223         (tanh(beta*(1-eta))+tanh(beta*(1-(1-eta))));
224     dce1(:) = H*(dce1(:).*dx(:)./Hs);
225     dve1(:) = H*(dve1(:).*dx(:)./Hs);
226     dce2(:) = H*(dce2(:).*dx(:)./Hs);
227     dve2(:) = H*(dve2(:).*dx(:)./Hs);
228     dxi = (beta*(1-(tanh(beta*(zTilde-1/2))))).^2) / ...
229         (tanh(beta*1/2)+tanh(beta*(1-1/2))));
230     dci1(:) = H*(dci1(:).*dxi(:)./Hs);
231     dvi1(:) = H*(dvi1(:).*dxi(:)./Hs);
232     dci2(:) = H*(dci2(:).*dxi(:)./Hs);
233     dvi2(:) = H*(dvi2(:).*dxi(:)./Hs);
234     dxd = (beta*(1-(tanh(beta*(zTilde-eta))))).^2) / ...
235         (tanh(beta*eta)+tanh(beta*(1-eta))));
236     dcd1(:) = H*(dcd1(:).*dxd(:)./Hs);
237     dvd1(:) = H*(dvd1(:).*dxd(:)./Hs);
238     dcd2(:) = H*(dcd2(:).*dxd(:)./Hs);
239     dvd2(:) = H*(dvd2(:).*dxd(:)./Hs);
240     %% CALCULATE OUTPUT
241     change = max(max(abs(z-reshape(zold1,nely,nelx))));
242     upfe1 = -(Ue(outDof(1))/nGaugeX)/(mean(Ue(inDof))/(2/3*(nelx)));
243     upfe2 = -(Ue(outDof(2))/nGaugeX)/(mean(Ue(inDof))/(2/3*(nelx)));
244     upfi1 = -(Ui(outDof(1))/nGaugeX)/(mean(Ui(inDof))/(2/3*(nelx)));
245     upfi2 = -(Ui(outDof(2))/nGaugeX)/(mean(Ui(inDof))/(2/3*(nelx)));
246     upfd1 = -(Ud(outDof(1))/nGaugeX)/(mean(Ud(inDof))/(2/3*(nelx)));
247     upfd2 = -(Ud(outDof(2))/nGaugeX)/(mean(Ud(inDof))/(2/3*(nelx)));
248     Mnde = sum(4*reshape(xPhyse(:),nel,1).* ...
249         (1-reshape(xPhyse(:),nel,1)))/nel*100;
250     Mndi = sum(4*reshape(xPhysi(:),nel,1).* ...

```

```

251     (1-reshape(xPhysi(:),nel,1))/nel*100;
252     Mndd = sum(4*reshape(xPhysd(:),nel,1).* ...
253     (1-reshape(xPhysd(:),nel,1))/nel*100;
254     %% PRINT RESULTS
255     fprintf(' It.:%5i      Obj.:%7.4f      Upf.:%7.3f      Vol.:%7.3f ...
256             Mnd.:%7.1f%      Ch.:%7.3f\n', ...
257             loop, cel, upfel, mean(xPhyse(:)), Mnde, change);
258     fprintf('%27.4f %16.3f\n', ce2, upfe2);
259     fprintf('%27.4f %16.3f %16.3f %16.1f%\n', ...
260             cil, upfil, mean(xPhysi(:)), Mndi);
261     fprintf('%27.4f %16.3f\n', ci2, upfi2);
262     fprintf('%27.4f %16.3f %16.3f %16.1f%\n', ...
263             cd1, upfd1, mean(xPhysd(:)), Mndd);
264     fprintf('%27.4f %16.3f\n', cd2, upfd2);
265     %% PLOT DENSITIES
266     if strcmpi(plotFiguresDuring, 'yes')
267         figure(erodedPlot);
268         colormap(gray); imagesc(1-xPhyse); caxis([0 1]); ...
269         axis equal; axis off; drawnow;
270         figure(intermediatePlot);
271         colormap(gray); imagesc(1-xPhysi); caxis([0 1]); ...
272         axis equal; axis off; drawnow;
273         figure(delatedPlot);
274         colormap(gray); imagesc(1-xPhysd); caxis([0 1]); ...
275         axis equal; axis off; drawnow;
276     end
277     %% UPDATE VOLUME FRACTION
278     xPhysTotal(:, :, 1) = xPhyse;
279     xPhysTotal(:, :, 2) = xPhysi;
280     xPhysTotal(:, :, 3) = xPhysd;
281     xPhysMean = mean(xPhysTotal, 3);
282     %% UPDATE HEAVISIDE REGULARIZATION PARAMETER
283     if beta < 25
284         beta = 1.01*beta;
285         change = 1;
286         fprintf('Parameter beta increased to %g.\n', beta);
287     end
288 end
289 %% PLOT DEFORMATIONS OF THE DESIGNS
290 if strcmpi(plotFiguresEnd, 'yes')
291     if strcmpi(plotFiguresDuring, 'no')
292         figure(erodedPlot);
293         colormap(gray); imagesc(1-xPhyse); caxis([0 1]); ...
294         axis equal; axis off; drawnow;
295         figure(intermediatePlot);
296         colormap(gray); imagesc(1-xPhysi); caxis([0 1]); ...
297         axis equal; axis off; drawnow;
298         figure(delatedPlot);
299         colormap(gray); imagesc(1-xPhysd); caxis([0 1]); ...
300         axis equal; axis off; drawnow;
301     end
302     erodedDeform = figure;
303     erodedDeform.IntegerHandle = 'off';
304     erodedDeform.Name = 'Deformation eroded design';
305     colormap(gray); axis equal; axis off;

```

D. MATLAB CODE: COMBINATION OF LENGTH SCALE CONTROL AND FIBRE IN PURE TENSION

```

305     for ely = 1:nely
306         for elx = 1:nelx
307             n1 = (nely+1)*(elx-1)+ely;
308             n2 = (nely+1)* elx +ely;
309             Uee = 1*Ue([2*n1-1;2*n1; 2*n2-1;2*n2; ...
310                 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
311             ly = ely-1; lx = elx-1;
312             xx = [Uee(1,1)+lx Uee(3,1)+lx+1 ...
313                 Uee(5,1)+lx+1 Uee(7,1)+lx ]';
314             yy = [Uee(2,1)-ly Uee(4,1)-ly ...
315                 Uee(6,1)-ly-1 Uee(8,1)-ly-1]';
316             patch(xx,yy,-xPhyse(ely,elx), 'EdgeColor', 'none')
317         end
318     end
319     drawnow;
320     intermediateDeform = figure;
321     intermediateDeform.IntegerHandle = 'off';
322     intermediateDeform.Name = 'Deformation intermediate design';
323     colormap(gray); axis equal; axis off;
324     for ely = 1:nely
325         for elx = 1:nelx
326             n1 = (nely+1)*(elx-1)+ely;
327             n2 = (nely+1)* elx +ely;
328             Uei = 1*Ui([2*n1-1;2*n1; 2*n2-1;2*n2; ...
329                 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
330             ly = ely-1; lx = elx-1;
331             xx = [Uei(1,1)+lx Uei(3,1)+lx+1 ...
332                 Uei(5,1)+lx+1 Uei(7,1)+lx ]';
333             yy = [Uei(2,1)-ly Uei(4,1)-ly ...
334                 Uei(6,1)-ly-1 Uei(8,1)-ly-1]';
335             patch(xx,yy,-xPhysi(ely,elx), 'EdgeColor', 'none')
336         end
337     end
338     drawnow;
339     delatedDeform = figure;
340     delatedDeform.IntegerHandle = 'off';
341     delatedDeform.Name = 'Deformation delated design';
342     colormap(gray); axis equal; axis off;
343     for ely = 1:nely
344         for elx = 1:nelx
345             n1 = (nely+1)*(elx-1)+ely;
346             n2 = (nely+1)* elx +ely;
347             Ued = 1*Ud([2*n1-1;2*n1; 2*n2-1;2*n2; ...
348                 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
349             ly = ely-1; lx = elx-1;
350             xx = [Ued(1,1)+lx Ued(3,1)+lx+1 ...
351                 Ued(5,1)+lx+1 Ued(7,1)+lx ]';
352             yy = [Ued(2,1)-ly Ued(4,1)-ly ...
353                 Ued(6,1)-ly-1 Ued(8,1)-ly-1]';
354             patch(xx,yy,-xPhysd(ely,elx), 'EdgeColor', 'none')
355         end
356     end
357     drawnow;
358     if strcmpi(saveFigures, 'yes')
359         saveas(erodedPlot, strcat(savingDirectory, 'erodedPlot'), 'jpeg');

```

```
360     saveas(intermediatePlot, ...
361           strcat(savingDirectory, 'intermediatePlot'), 'jpeg');
362     saveas(delatedPlot, ...
363           strcat(savingDirectory, 'delatedPlot'), 'jpeg');
364     saveas(erodedDeform, ...
365           strcat(savingDirectory, 'erodedDeform'), 'jpeg');
366     saveas(intermediateDeform, ...
367           strcat(savingDirectory, 'intermediateDeform'), 'jpeg');
368     saveas(delatedDeform, ...
369           strcat(savingDirectory, 'delatedDeform'), 'jpeg');
370     end
371 end
```


Appendix E

ANSYS input files

E.1 General input file

```
! SET DIRECTORY OF INPUT FILES
*DIM,directory,STRING,248
directory(1) = 'C:\Users\Tom\Ansys\Compression'

! READ INPUT PARAMETERS
*dim,input,array,7,1 !input (parameters)
*vread,input(1),input,txt,directory(1)
(E26.16)

! GRID PROPERTIES
*SET,nelx,input(1)
*SET,nely,input(2)
*SET,gridsize,input(3)/1000
*SET,t,0.005

! READ MATLAB DESIGN
*dim,xPhys,array,nely*nelx,1 !xPhys (densities)
! ASK USER TO LOAD MATLAB OR POST-PROCESSED DENSITIES
*ASK,Density,'Load MATLAB (M) or post-processed (P) densities?','P'
Density=UPCASE(Density)
*IF,Density,EQ,'M',or,Density,EQ,'MATLAB',THEN
*vread,xPhys(1),xPhys,txt,directory(1)
(E26.16)
*ELSEIF,Density,EQ,'P',or,Density,EQ,'POST-PROCESSED',THEN
*vread,xPhys(1),xPhysPost,txt,directory(1)
(E26.16)
*ENDIF
*dim,inDof,array,input(4) !inDof (input nodes for the displacements)
*vread,inDof(1),inDof,txt,directory(1)
```

E. ANSYS INPUT FILES

```
(E26.16)
*dim,outDof,array,input(5) !outDof (output nodes for the displacements)
*vread,outDof(1),inDof,txt,directory(1)
(E26.16)
```

```
! MATERIAL PROPERTIES
```

```
*SET,E,69e9
*SET,nu,0.33
*SET,rho,2700
*SET,k,0.0908
```

```
/prep7
```

```
! CREATE NODES
```

```
*DO,i,1,2*(nely+1)*(nelx+1),2
*SET,XX,(i-mod(i,2*(nely+1)))/(2*(nely+1))
*SET,YY,-(mod(i,2*(nely+1))-1)/2
N,i,gridSize*XX,gridSize*YY,0
*ENDDO
```

```
! DEFINE ELEMENT PROPERTIES
```

```
ET,1,SHELL63
ET,2,COMBIN14
R,1,t,t,t,t
R,2,k
MP,EX,1,E
MP,PRXY,1,nu
MP,DENS,1,rho
MP,DENS,2,0
```

```
! DEFINE ELEMENTS
```

```
elementnumber=0
*DO,i,1,2*(nely+1)*nelx,2
*IF,mod(i+1,2*(nely+1)),NE,0,THEN
elementnumber=elementnumber+1
*IF,xPhys(elementnumber),GE,0.5,THEN
NUMSTR,ELEM,elementnumber
E,i,i+2,i+2*(nely+1)+2,i+2*(nely+1)
*ENDIF
*ENDIF
*ENDDO
```

```
! DEFINE FIBRE
```

```
TYPE,2
REAL,2
MAT,2
```

```

E,(nely+1),9*(nely+1)
CE,1,0,9*(nely+1),UY,1,(nely+1),UY,-1
CE,2,0,9*(nely+1),UZ,1,(nely+1),UZ,-1

! DELETE UNUSED NODES
NDELE,ALL

! APPLY SYMMETRIC BOUNDARY CONDITIONS
NSEL,,LOC,x,0
DSYM,SYMM,X

! ASK USER IF MASS OF FIBRE MOUNTING BRACKETS IS REQUIRED
*ASK,Mounting,'Add mass of fibre mounting brackets? (yes/no)', 'Yes'
Mounting=UPCASE(Mounting)
*IF,Mounting,EQ,'Y',or,Mounting,EQ,'YES',THEN
ET,3,MASS21,,2
R,3,0.02
TYPE,3
REAL,3
MAT,2
N,2*(nely+1)*(nelx+1)+1,gridsize*7,-gridSize*(nely/2),0.0075
N,2*(nely+1)*(nelx+1)+3,gridsize*7,-gridSize*(nely/2),-0.0075
E,2*(nely+1)*(nelx+1)+1
E,2*(nely+1)*(nelx+1)+3
CE,3,0,2*(nely+1)*(nelx+1)+1,UX,1,15*(nely+1),UX,-1
CE,4,0,2*(nely+1)*(nelx+1)+1,UY,1,15*(nely+1),UY,-1
CE,5,0,2*(nely+1)*(nelx+1)+1,UZ,1,15*(nely+1),UZ,-1
CE,6,0,2*(nely+1)*(nelx+1)+3,UX,1,15*(nely+1),UX,-1
CE,7,0,2*(nely+1)*(nelx+1)+3,UY,1,15*(nely+1),UY,-1
CE,8,0,2*(nely+1)*(nelx+1)+3,UZ,1,15*(nely+1),UZ,-1
*ELSEIF,Mounting,EQ,'N',or,Mounting,EQ,'NO',THEN
*MSG,UI,'No mass from fibre','mounting bracket added.'
%C %C
*ENDIF

! ASK USER WHICH ANALYSIS TO PERFORM AND EXECUTE THAT LOGFILE
*ASK,Analysis,'S (Static); D (Dynamic) or B (Buckling)', 'Static'
Analysis=UPCASE(Analysis)
*IF,Analysis,EQ,'S',or,Analysis,EQ,'STATIC',THEN
/INPUT,logStatic,txt,directory(1)
*ELSEIF,Analysis,EQ,'D',or,Analysis,EQ,'DYNAMIC',THEN
/INPUT,logDynamic,txt,directory(1)
*ELSEIF,Analysis,EQ,'B',or,Analysis,EQ,'BUCKLING',THEN
/INPUT,logBuckling,txt,directory(1)
*ELSE

```

```
*MSG,UI,'The selected option was ','not valid.',\  
  'The execution of the ','logfile is terminated.'  
%C %C %/ %C %C  
*ENDIF
```

E.2 Input file static analysis

```
! SELECT INPUT DOFS AND APPLY DISPLACEMENTS  
NSEL,NONE  
*DO,i,1,input(4),1  
NSEL,A,NODE,,inDof(i)  
*ENDDO  
D,ALL,UX,-3.8667e-6  
D,ALL,UY,0,, ,UZ  
NSEL,ALL  
  
! STATIC ANALYSIS  
/SOLUTION  
  
ANTYPE,STATIC  
NLGEOM,ON  
NSUBST,10  
OUTRES,ERASE  
OUTRES,ALL,1  
AUTOTS,0  
TIME,1  
  
SOLVE  
FINISH  
  
! TIME HISTORY POST-PROCESSING  
/POST26  
FILE,'file','rst','.'  
/UI,COLL,1  
NUMVAR,200  
SOLU,191,NCMIT  
STORE,MERGE  
FILLDATA,191,, ,1,1  
REALVAR,191,191  
NSOL,2,909,U,X, UX_2,  
STORE,MERGE  
NSOL,3,20099,U,X, UX_3,  
STORE,MERGE  
FILLDATA,192,, ,0,0
```

```

FILLDATA,193,, ,1,0
FILLDATA,194,, , -1,0
FILLDATA,195,, ,1,1
VARNAME,195,NSET
FILLDATA,199,, , -10,0
REALVAR,199,199
QUOT,200,2,199
FILLDATA,198,, ,250,0
REALVAR,198,198
QUOT,199,3,198
QUOT,4,200,199,,Upscaling_factor
STORE,MERGE
FILLDATA,191,, ,1,1
REALVAR,191,191
FILLDATA,192,, ,0,0
FILLDATA,193,, ,1,0
FILLDATA,194,, , -1,0
FILLDATA,195,, ,1,1
VARNAME,195,NSET
FILLDATA,199,, ,3.8667e-6,0
REALVAR,199,199
PROD,5,199,1,,U_Input
STORE,MERGE

```

E.3 Input file dynamic analysis

```

! SELECT INPUT DOFS AND APPLY DISPLACEMENTS
NSEL,NONE
*DO,i,1,input(4),1
NSEL,A,NODE,,inDof(i)
*ENDDO
D,ALL,UX,-0e-3
D,ALL,UY,0,, ,UZ
NSEL,ALL

! DYNAMIC ANALYSIS
/SOLUTION
ANTYPE,modal
MODOPT,LANB,5
SOLVE

```

E.4 Input file buckling analysis

```
! SELECT INPUT DOFS AND APPLY DISPLACEMENTS
NSEL,NONE
*DO,i,1,input(4),1
NSEL,A,NODE,,inDof(i)
*ENDDO
D,ALL,UX,-3.8667e-6
D,ALL,UY,0,,,UZ
NSEL,ALL

! STATIC ANALYSIS
/SOLUTION

ANTYPE,STATIC
PSTRES,ON
NSUBST,10
OUTRES,ERASE
OUTRES,ALL,1
AUTOTS,0
TIME,1

SOLVE
FINISH

! BUCKLING ANALYSIS
/SOLUTION
ANTYPE,1
BUCOPT,LANB,5
SOLVE
```

Appendix F

Production data of the optical fibres



DTG® INFORMATION

1. ORDER

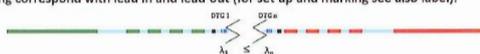
DOC201600 077

2. COATING

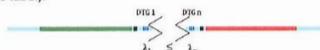
- ORMOCER®
- Acrylat
- Other:

3. WAVELENGTH INDICATION

- no extra indication for Single DTG®
- The fibre end next to the lowest Bragg-wavelength is indicated with a green ink mark, the highest Bragg-wavelength is indicated with a red ink mark. Before the DTG® with the lowest Bragg-wavelength there is a green dashed marking and after the DTG® with the highest Bragg-wavelength a red dashed marking. For chains with same wavelength DTG®s green and red marking correspond with lead in and lead out (for set up and marking see also label).



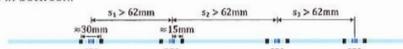
- Due to short fiber ends, before the DTG® with the lowest Bragg-wavelength there is a green marking and after the DTG® with the highest Bragg-wavelength a red marking. For chains with same wavelength DTG®s green and red marking correspond with lead in and lead out (for set up and marking see also label).



- Other:

4. DTG® MARKING

- The DTG®s are individually marked by 2 black markings with a spacing of approx. 30 mm and the DTG® is in between.



- Due to the spacing smaller than 62 mm between successive DTG®, the DTG®s are marked by a black marking approx. 15 mm before the centre of the first DTG® and a second black marking approx. 15 mm after the centre of the last DTG®.



2 black markings on the fibre to mark a group of DTG®s with tight spacings (before first and after last DTG® of that group)

- Other:

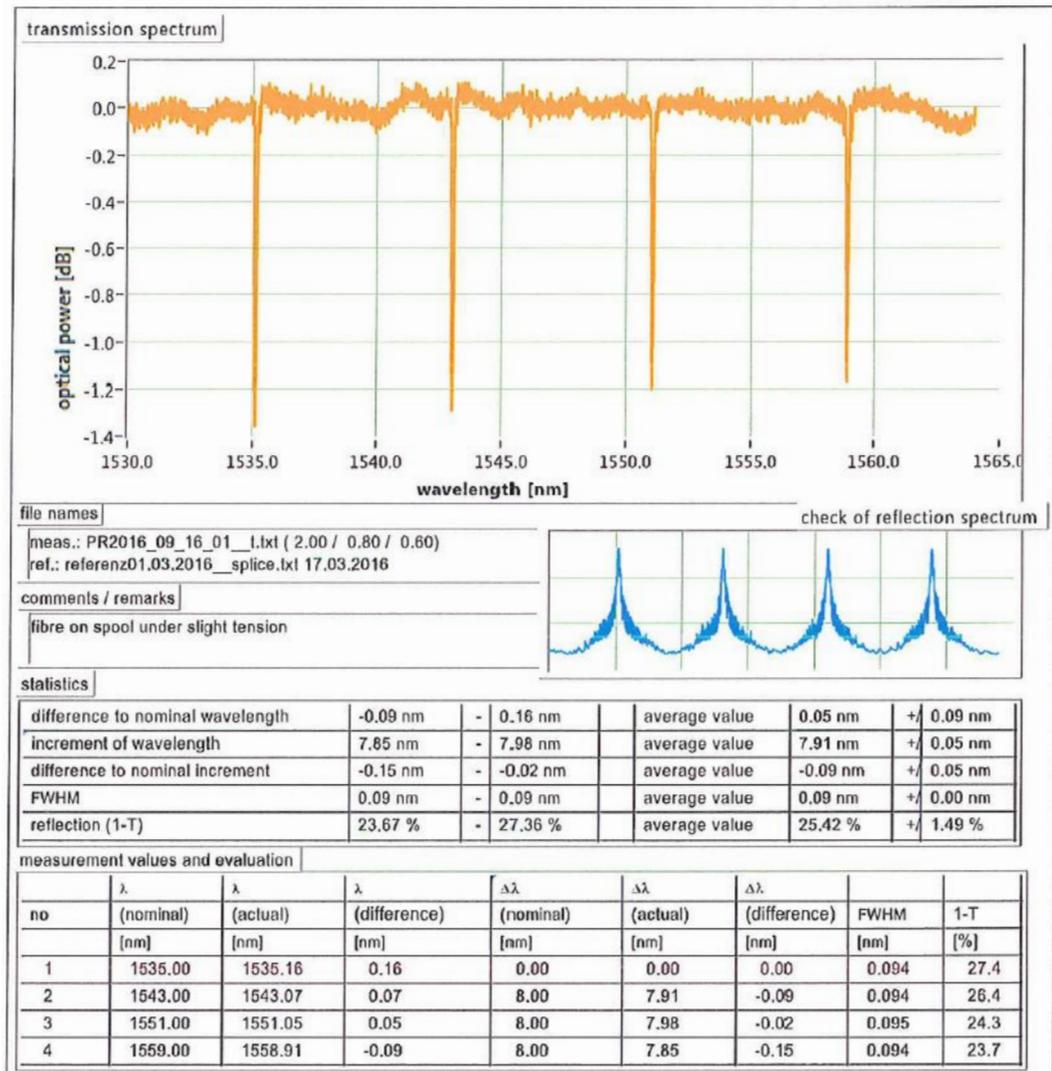
5. REMARKS

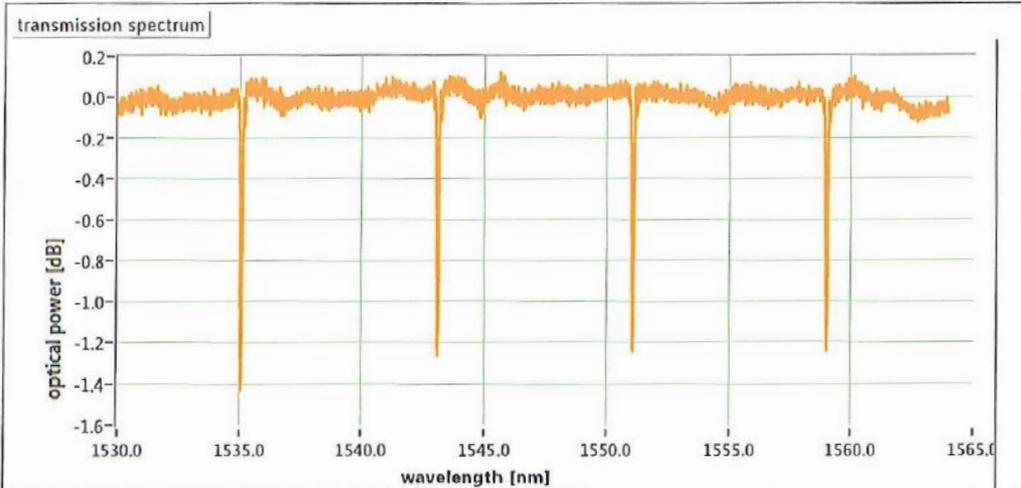
- The DTG® measurement and its evaluation are added
- All marking can be removed by wiping with ethanol or acetone

18.03.2016 *Schubmann*

Date, Signature

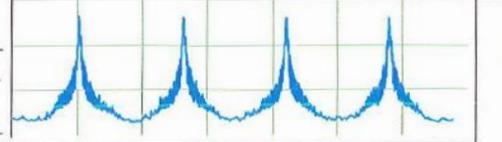
F. PRODUCTION DATA OF THE OPTICAL FIBRES





file names check of reflection spectrum

meas.: PR2016_09_16_02_1.txt (2.00 / 0.80 / 0.60)
 ref.: referenz01.03.2016__splice.txt 17.03.2016



comments / remarks
 fibre on spool under slight tension

statistics					
difference to nominal wavelength	0.01 nm	-	0.14 nm	average value	0.08 nm +/- 0.05 nm
increment of wavelength	7.93 nm	-	8.04 nm	average value	7.97 nm +/- 0.05 nm
difference to nominal increment	-0.07 nm	-	0.04 nm	average value	-0.03 nm +/- 0.05 nm
FWHM	0.09 nm	-	0.10 nm	average value	0.10 nm +/- 0.00 nm
reflection (1-T)	25.16 %	-	28.50 %	average value	26.31 % +/- 1.31 %

measurement values and evaluation

no	λ (nominal) [nm]	λ (actual) [nm]	λ (difference) [nm]	$\Delta\lambda$ (nominal) [nm]	$\Delta\lambda$ (actual) [nm]	$\Delta\lambda$ (difference) [nm]	FWHM [nm]	1-T [%]
1	1535.00	1535.09	0.09	0.00	0.00	0.00	0.094	28.5
2	1543.00	1543.14	0.14	8.00	8.04	0.04	0.095	26.1
3	1551.00	1551.08	0.08	8.00	7.94	-0.06	0.098	25.5
4	1559.00	1559.01	0.01	8.00	7.93	-0.07	0.098	25.2

Bibliography

- [1] M. M. Abdel Wahab and G. De Roeck. Damage Detection in Bridges Using Modal Curvatures: Application To a Real Damage Scenario. *Journal of Sound and Vibration*, 226(2):217–235, 1999.
- [2] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43:1–16, 2011.
- [3] M. P. Bendsøe and O. Sigmund. *Topology optimization: theory, methods, and applications*. Number 724. Springer, 2 edition, 2003.
- [4] J. R. Casas and P. J. S. Cruz. Fiber Optic Sensors for Bridge Monitoring. *Journal of Bridge Engineering*, 8(6):362–373, 2003.
- [5] F. Casciati, M. Domaneschi, D. Inaudi, A. Figini, B. Glišić, and A. Gupta. Long-Gauge Fiber-Optic Sensors: A New Approach to Dynamic System Identification. In *Proceedings of the Third European Conference on Structural Control, 3ECSC*, page 4, 2004.
- [6] T. H. T. Chan, L. Yu, H. Y. Tam, Y. Q. Ni, S. Y. Liu, W. H. Chung, and L. K. Cheng. Fiber Bragg grating sensors for structural health monitoring of Tsing Ma bridge: Background and experimental observation. *Engineering Structures*, 28(5):648–659, 2006.
- [7] B. J. A. Costa and J. A. Figueiras. Evaluation of a strain monitoring system for existing steel railway bridges. *Journal of Constructional Steel Research*, 72:179–191, 2012.
- [8] Engineering Toolbox. Modulus of Elasticity or Young’s Modulus - and Tensile Modulus for common Materials, 2015.
- [9] S.-E. Fang, R. Perera, and G. De Roeck. Damage identification of a reinforced concrete frame by finite element model updating using damage parameterization. *Journal of Sound and Vibration*, 313:544–559, 2008.
- [10] FAZtechnology. FAZT I4 Interrogator, 2016.
- [11] FBGS. How is the DTG wavelength changing with temperature and strain?, 2016.

- [12] Y. M. Gebremichael, W. Li, B. T. Meggitt, W. J. O. Boyle, K. T. V. Grattan, B. McKinley, L. F. Boswell, K. A. Aarnes, S. E. Aasen, B. Tynes, Y. Fonjallaz, and T. Triantafillou. A field deployable, multiplexed Bragg grating sensor system used in an extensive highway bridge monitoring evaluation tests. *IEEE Sensors Journal*, 5(3):510–519, 2005.
- [13] B. Glišić and D. Inaudi. *Fibre Optic Methods for Structural Health Monitoring*. Wiley, Chichester, 2007.
- [14] S.-c. Her and C.-y. Huang. The Effects of Adhesive and Bonding Length on the Strain Transfer of Optical Fiber Sensors. *Applied Sciences*, 6(13):1–9, 2016.
- [15] W. Hong, Z. Wu, C. Yang, C. Wan, and G. Wu. Investigation on the damage identification of bridges using distributed long-gauge dynamic macrostrain response under ambient excitation. *Journal of Intelligent Material Systems and Structures*, 23(1):85–103, 2012.
- [16] M. Jansen. *A robust approach to topology optimization accounting for geometric imperfections*. PhD thesis, KU Leuven, 2014.
- [17] A. Kerrouche, W. J. O. Boyle, T. Sun, and K. T. V. Grattan. Design and in-the-field performance evaluation of compact FBG sensor system for structural health monitoring applications. *Sensors and Actuators, A: Physical*, 151(2):107–112, 2009.
- [18] J. T. Kim, Y. S. Ryu, H. M. Cho, and N. Stubbs. *Damage identification in beam-type structures: Frequency-based method vs mode-shape-based method*, volume 25. 2003.
- [19] B. S. Lazarov, M. Schevenels, and O. Sigmund. Robust design of large-displacement compliant mechanisms. *Mechanical Sciences*, 2(2):175–182, 2011.
- [20] K. Liu and A. Tovar. An efficient 3D topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 50(6):1175–1196, 2014.
- [21] J. M. Lopez-Higuera, L. Rodriguez Cobo, A. Quintela Incera, and A. Cobo. Fiber Optic Sensors in Structural Health Monitoring. *Journal of Lightwave Technology*, 29(4):587–608, 2011.
- [22] J. Luo, Z. Luo, S. Chen, L. Tong, and M. Y. Wang. A new level set method for systematic design of hinge-free compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 198(2):318–331, 2008.
- [23] K. Maute and D. M. Frangopol. Reliability-based design of MEMS mechanisms by topology optimization. *Computers and Structures*, 81:813–824, 2003.
- [24] C. B. W. Pedersen, T. Buhl, and O. Sigmund. Topology synthesis of large-displacement compliant mechanisms. *International Journal for Numerical Methods in Engineering*, 50:2683–2705, 2001.

-
- [25] E. Reynders, G. De Roeck, P. Gundes Bakir, and C. Sauvage. Damage Identification on the Tilff Bridge by Vibration Monitoring Using Optical Fiber Strain Sensors. *Journal of Engineering Mechanics*, 133(2):185–193, 2007.
- [26] M. Schevenels. Topology optimization : basic concepts and robust design Introduction. In *Bauhaus Summer School 2014: Forecast Engineering*, number August 18-29, Weimar, 2014.
- [27] M. Schevenels, B. S. Lazarov, and O. Sigmund. Robust topology optimization accounting for spatially varying manufacturing errors. *Computer Methods in Applied Mechanics and Engineering*, 200:3613–3627, 2011.
- [28] W. L. Schulz, J. P. Conte, and E. Udd. Long-gage fiber optic Bragg grating strain sensors to monitor civil structures. *Proceedings of SPIE*, pages 56–65, 2001.
- [29] W. L. Schulz, J. P. Conte, E. Udd, and J. M. Seim. Static and dynamic testing of bridges and highways using long-gage fiber Bragg grating based strain sensors. *Industrial Sensing Systems*, 4202:79–86, 2000.
- [30] O. Sigmund. On the Design of Compliant Mechanisms Using Topology Optimization. *Mechanics Based Design of Structures and Machines*, 25(4):493–524, 1997.
- [31] O. Sigmund. A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 21:120–127, 2001.
- [32] O. Sigmund. Design of multiphysics actuators using topology optimization - Part II: Two material structures. *Computer Methods in Applied Mechanics and Engineering*, 190:6577–6604, 2001.
- [33] O. Sigmund. Manufacturing tolerant topology optimization. *Acta Mechanica Sinica/Lixue Xuebao*, 25(2):227–239, 2009.
- [34] Suzhen Li and Zhishen Wu. Sensitivity Enhancement of Long-gage FBG Sensors for Macro-strain Measurements. *Structural Health Monitoring*, 8(6):415–423, 2009.
- [35] K. Svanberg. MMA and GCMMA, two methods for nonlinear optimization. 2007.
- [36] R. C. Tennyson, A. A. Mufti, S. Rizkalla, G. Tadros, and B. Benmokrane. Structural health monitoring of innovative bridges in Canada with fiber optic sensors. *Smart Materials and Structures*, 10:560–573, 2001.
- [37] A. Teughels and G. De Roeck. Structural damage identification of the highway bridge Z24 by FE model updating. *Journal of Sound and Vibration*, 278:589–610, 2004.

- [38] J. F. Unger, A. Teughels, and G. De Roeck. Damage detection of a prestressed concrete beam using modal strains. *Journal of Structural Engineering*, 131(9):1456–1463, 2005.
- [39] F. Wang, B. S. Lazarov, and O. Sigmund. On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 43:767–784, 2011.

Fiche masterproef

Student: Tom Jonckers

Titel: Development of High-accuracy Strain Sensors using Topology Optimization

Nederlandse titel: Ontwikkeling van hoogperformante reksensoren met topologische optimalisatie

UDC: 69

Korte inhoud:

In this MSc thesis, high-accuracy strain sensors are developed. The aim of the sensor package is to perform very accurate, one dimensional strain measurements. Once the high-accuracy sensor package is fully developed, it can be used in applications such as force identification or structural health monitoring under ambient loading conditions.

In a first part of this thesis, the design of a transducer is made. This design is obtained using the topology optimization technique. Four different topology optimization models are constructed during this thesis. First, a model with very little constraints is made. Next, the complexity is increased, adding constraints for length scale control and imposing the optical fibre in the sensor package to be loaded in pure tension. Combining all models will result in the most complex topology optimization model. Ultimately, two final designs of the transducers are selected. One design will perform optimal in compression, the other in tension.

After the designs are finalized, a numerical validation of the transducers is performed. First, the results of the topology optimization are verified. Next, other properties of the transducers, which are not included in the topology optimization, are investigated. These properties include the occurring stresses, the eigenfrequencies and buckling modes of the transducers.

In a last part, an experimental validation is performed. A comparison is made between the numerically predicted and the measured performance. Lastly, further steps in the development of the strain upscaling sensor package are proposed, including some possible improvements and additional experiments.

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: bouwkunde, optie Civiele techniek

Promotoren: Prof. dr. ir. E. Reynders
Prof. dr. ir. G. De Roeck
Prof. dr. ir. M. Schevenels

Assessoren: Prof. dr. ir. T. Geernaert
Ir. C. Van hoorickx

Begeleider: Ir. C. Van hoorickx